Transportation Research Part C 117 (2020) 102635

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

GE-GAN: A novel deep learning framework for road traffic state estimation

Dongwei Xu[a,b,c,*], Chenchen Wei[a,b], Peng Peng[a,b], Qi Xuan[a,b], Haifeng Guo[b,c]

[a] Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China
[b] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
[c] Research Institute, Enjoyor Co., Ltd., Hangzhou 310023, China

ARTICLE INFO

Keywords:
Traffic state estimation
Graph embedding
Generative adversarial network
Deepwalk

ABSTRACT

Traffic state estimation is a crucial elemental function in Intelligent Transportation Systems (ITS). However, the collected traffic state data are often incomplete in the real world. In this paper, a novel deep learning framework is proposed to use information from adjacent links to estimate road traffic states. First, the representation of the road network is realized based on graph embedding (GE). Second, with this representation information, the generative adversarial network (GAN) is applied to generate the road traffic state information in real-time. Finally, two typical road networks in Caltrans District 7 and Seattle area are adopted as cases study. Experimental results indicate that the estimated road traffic state data of the detectors have higher accuracy than the data estimated by other models.

1. Introduction

With the increasing demand for trips and transportation, traffic problems such as traffic congestion and traffic safety are becoming increasingly severe, which seriously affects economic development and people's lives (Xu et al., 2015b). The accurate determination of road traffic states is the most important and basic part of solving these traffic problems (Xu et al., 2015a). The road traffic state is represented by measurements from many traffic detectors. In the existing literature (Liang et al., 2018), the traffic state usually refers to traffic flow, traffic density, traffic speed, and travel time. Accurate traffic state estimation can allow traffic system managers to effectively handle and control transportation networks and avoid traffic jams. Thus, travelers can better plan their departure times and travel routes to make their trips more efficient (Xu et al., 2012). However, it is difficult to determine all traffic states in a road network because of the sparse distribution of detectors.

Recently, many traffic state estimation methods have been proposed. Existing methods for traffic state estimation can be divided into two categories: model-driven and data-driven methods (Wang et al., 2018). The former mainly use macroscopic traffic models to describe the traffic state of freeway corridors, while the latter apply statistical analysis based on known measurements obtained by traffic state detectors. Conventional traffic state estimation methods include the Kalman filter (KF) (Nanthawichit et al., 2003), KF-like techniques (KFTs), the extended Kalman filter (EKF) (Wang et al., 2007), the unscented Kalman (UKF) filter, (Pueboobpaphan et al., 2007), particle filters (PF) (Mihaylova and Boel, 2004), and nonlinear Kalman filtering (NKF) (Antoniou et al., 2007). The KFTs are an extension of KF that estimate the most likely state variables concerning an available observation, a system model, and noise. Road traffic running states have a certain amount of regularity and reproducibility when the road conditions and traffic flow conditions are similar (Xu et al., 2014). Data-driven methods usually extract spatiotemporal features from historical data from loop

* Corresponding author.
E-mail address: dongweixu@zjut.edu.cn (D. Xu).

https://doi.org/10.1016/j.trc.2020.102635
Received 6 September 2019; Received in revised form 29 February 2020; Accepted 26 March 2020
Available online 27 June 2020
0968-090X/

detectors and apply statistical or machine learning approaches to infer real-time traffic states. With the drastic increase in the amount of traffic state data, several researchers have developed methodologies that increasingly depend on historical data. The auto-regressive integrated moving average (ARIMA) (Lee and Fambro, 1999; Xu et al., 2017) employs a time-series dataset from historical data to estimate traffic states. K-nearest neighbor (KNN) (Tak et al., 2016; Xu et al., 2018) and probabilistic principal component analysis (PPCA) (Qu et al., 2009) have been utilized to incorporate additional spatiotemporal information and increase the accuracy of traffic state estimations. Besides, with the development of machine learning and the rapid improvement of computing power, there are many novel traffic state estimation methods. The tensor decomposition based model (Wu et al., 2018; Bahadori et al., 2014; Chen et al., 2019) is a representative one that used neighborhood information as graph regularizer to perform traffic state estimation. However, the tensor decomposition based models always organize the traffic state data as a third-order or higher order tensor (e.g., road segment × day × time interval), and the calculation of these high-dimensional tensors requires high memory and computing power. Moreover, with the development of graph neural network (Scarselli et al., 2008; Liang and Bose, 1996), many state-of-the-art graph-based traffic state estimation models have been proposed (Yu et al., 2017; Lee et al., 2019; Zheng et al., 2019).

However, most estimation algorithms are realized based on spatiotemporal correlations, which usually artificially selected the spatial information (Li et al., 2015; Li et al., 2019). Hence, a novel deep learning framework based on GE-GAN is proposed to estimate traffic states in this paper. The deep learning framework based on GE-GAN is to generate road traffic states with data of efficient adjacent links. Hence the traffic state estimation can ensure the integrity of road traffic state data. The road traffic state estimation mainly includes the following two issues: (1) how to select the effective spatial information; (2) how to generate road traffic state data in real-time. To deal with the two issues, the graph embedding (GE) (Yan et al., 2006) is applied to select the most relevant links for the target link from the road network, and the generative adversarial network (GAN) (Goodfellow et al., 2014) is used to generate data of target road traffic state detector based on the spatiotemporal information of the selected links. In this paper, DeepWalk (Perozzi et al., 2014) is employed to obtain the spatial representation information in the road network, which can work efficiently even the road traffic state detectors are sparse.

Our contributions are listed as follows:

(1) DeepWalk is used for the graph embedding of the road network; it transfers the road network to low-dimensional space and acquires representation information of the target road traffic state sensor.
(2) With the representation information and road traffic state information of adjacent links, the generative adversarial network (GAN) is applied to learn the road traffic state data distribution and generate road traffic state data of the road traffic detector. Moreover, the Wasserstein GAN (WGAN) (Arjovsky et al., 2017) is employed to train the GAN model to solve the training difficulties of original GAN.
(3) Results show that the novel deep learning framework based on GE-GAN has increased accuracy, which has better performances compared to some state-of-art road traffic state estimation methods.

The remainder of the paper is organized as follows. Section 2 reviews the related work about GAN and graph embedding. In Section 3, the methodology framework of road traffic state estimation is detailed described. Section 4 presents the results of numerical experiments and discussions about the cases studies. At last, this paper is concluded in Section 5.

2. Related work

In the last few years, deep learning methods have been widely used in different fields. Compared with traditional neural networks, they are able to effectively extract features from mass data. Generative adversarial networks (GANs) provide a powerful modeling framework for learning complex high-dimensional distributions. GANs represent a probability distribution through a generator G that learns to directly produce samples from the desired distribution. The generator is trained adversarially by optimizing a minimax objective together with a discriminator D. Formally, let P_z (Z) be the input standard distribution and P_{data} (X) be the training data distribution; then the minimax objective of GANs is defined as in Eq. (1). Applications of GANs have included image dataset synthesis (Shrivastava et al., 2017), image-to-image style translation (Isola et al., 2017), image superresolution (Ledig et al., 2017), and dialogue generation (Li et al., 2017).

$$\min_G \max_D E_{x\sim P_{data}(x)}[\log(D(x))] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

Recently, GANs are also widely used in the field of transportation. There are many novel methods for road traffic state estimation based on GAN. For example, K. Zhang proposed a novel framework that utilize the Info-GAN for estimation of trip travel time distribution with trajectory data (Zhang et al., 2019). J. J. Q. Yu used the graph convolution network as the internal structure of the GAN to generate the traffic speed in real-time (Yu and Gu, 2019). Yilun Lin proposed pattern-sensitive networks to prediction the traffic flow based on GAN (Lin et al., 2018). However, the problem of GAN training difficulties need to be solved, because the most difficult part of GAN is how to train the generator and discriminator to achieve a Nash equilibrium and deal with the problem of model collapse. The WGAN was proposed to solve the GAN training difficulties. WGAN proposes a new cost function using Wasserstein distance which has a smoother gradient than the Jensen-Shannon divergence.

Graph analysis has been attracting increasing attention due to the ubiquity of networks in the real world. Graphs naturally exist in a wide variety of real-world scenarios, e.g., social graphs, diffusion graphs in social media networks, traffic graphs, user interest graphs in electronic commerce, and knowledge graphs. Graph embedding (GE) assigns nodes in a network to low-dimensional

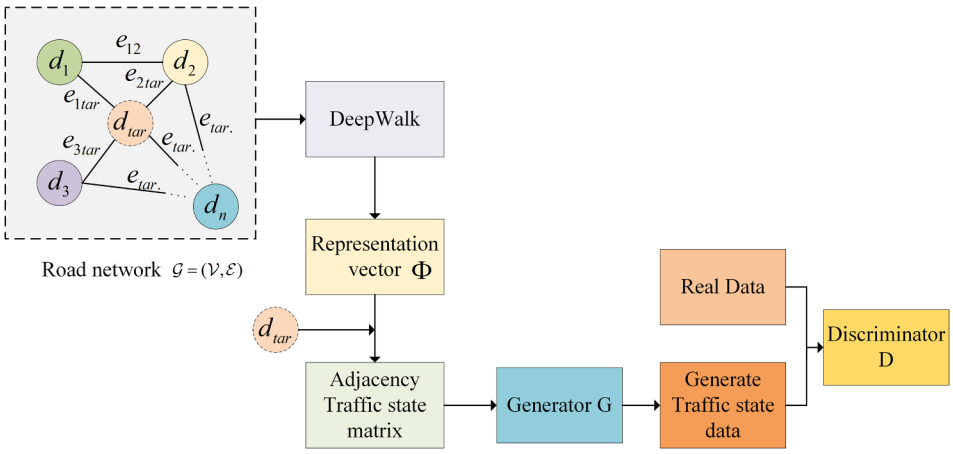

Fig. 1. The structure of the GE-GAN.

representations and effectively preserves the network structure. Analyzing these representations provides insights into how to make good use of the information hidden in graphs. Graph embedding is a promising method of network representation that is capable of supporting subsequent network processing and analysis tasks such as node classification (Bhagat et al., 2011), node clustering (Ding et al., 2001), network visualization(Maaten and Hinton, 2008) and link prediction (Liben-Nowell and Kleinberg, 2007).

3. Methodology

Let $\mathcal{G}=(\mathcal{V},\mathcal{E})$ be a road network that is a collection of detectors ($\mathcal{V}=(d_1, d_2,\cdots,d_n)$) and edges $\mathcal{E}=\{e_{ij}\}_{i,j=1}^{n}$, where $n=|\mathcal{V}|$ denotes the number of the detectors in the road network. Two detectors are regarded as adjacent if the two corresponding links are connected. If the i^{th} detector d_i and the j^{th} detector d_j are not adjacent in the road network, $e_{ij}=0$; otherwise, $e_{ij}=1$. Let d_{tar} denote the target road traffic state detector. Then, the framework of the GE-GAN algorithm for road traffic state estimation, is shown in Fig. 1, is composed of two parts: DeepWalk and GAN. The goals of DeepWalk are to obtain the representation vectors Φ of the road network and to obtain the corresponding traffic state matrix. Then, the GAN utilizes the traffic state matrix as the input for adversarial training to generate the traffic state data of the target detector.

3.1. Graph embedding of the road network based on DeepWalk

DeepWalk is a typical method of graph embedding, which can train parallelly when the graph is large. Besides, it can adapt to the local network changes because the changes of edges and nodes will only affect a part of random walk paths. It consists of two main components: random walk generation and the update procedure. Random walks have been used as similarity measurements for a variety of problems in content recommendations and community detection. Deepwalk can use adjacent nodes in network to realize the effective representation of the target node. For the road network, the node should be road traffic state detector or road segment, so the Deepwalk of road network can be considered as effective spatial representation of the original road network, which have been proven to be useful for road traffic state prediction (Xu et al., 2019). The architecture of DeepWalk is shown in Fig. 2. Let $\mathcal{W}_{d_i}$ denote a random walk rooted at road traffic state detector d_i. It is a stochastic process with random variables $\{\mathcal{W}_{d_i}^1, \mathcal{W}_{d_i}^2,\cdots,\mathcal{W}_{d_i}^k\}$, and $\mathcal{W}_{d_i}=\{\mathcal{W}_{d_i}^1, \mathcal{W}_{d_i}^2,\cdots,\mathcal{W}_{d_i}^k\}$. $\mathcal{W}_{d_i}^j$ represents the j^{th} detector in the walk rooted at d_i detector and $\mathcal{W}_{d_i}^{j+1}$ is the detector chosen at random from the j^{th} detector. For each detector, the length of the random walk is specified as k and random walk iterates η times.

After finishing the random walk for each detector, the Skip-Gram algorithm (Mikolov et al., 2013) is adopted to update these representations. The mapping function Φ: $d\in\mathcal{V}\mapsto \mathrm{R}^{|\mathcal{V}|\times\mathcal{G}}$ gives the latent representation associated with each detector d in the road network. The goal is to find the most relevant detectors for each road traffic state detector d_i. To do this, the model must maximize the probability that any detector appears in the context; this is expressed as an optimization problem in Eq. (2). Through the update procedure shown in Eq. (2), the most effective representation of the original matrix can be achieved (Perozzi et al., 2014), which can be seen as the procedure of searching effective spatial representation in road network.

$$\underset{\Phi}{\text{minimize}} \ -\log \Pr(\{d_{i-w},\cdots,d_{i-1}, d_{i+1},\cdots,d_{i+w}\}|\Phi(d_i)) \tag{2}$$

where w is the size of the selected window.

Then, it approximates the conditional probability in Eq. (2) using an independence assumption, as in Eq. (3):

$$\Pr(\{d_{i-w},\cdots,d_{i-1}, d_{i+1},\cdots,d_{i+w}\}|\Phi(d_i)) = \prod_{j=i-w,j\neq i}^{i+w} \Pr(d_j|\Phi(d_i)) \tag{3}$$

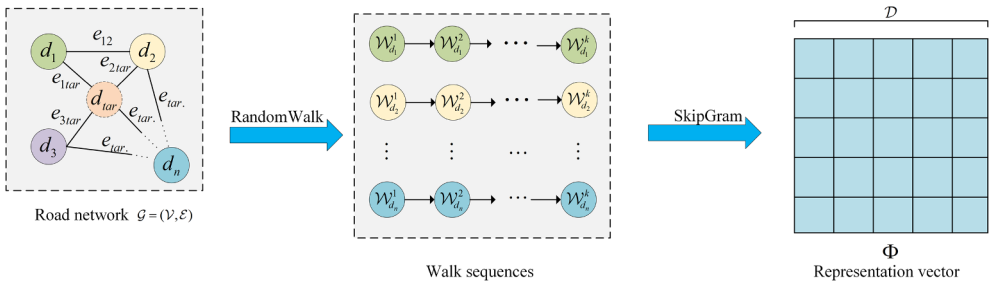

Fig. 2. Architecture of DeepWalk on the road network.

This means that each detector $d_i \in \mathscr{W}_{d_i}$ will map to its current representation vector $\Phi(d_i) \in R^{\mathscr{D}}$, where $\mathscr{D}$ represents the dimension of the representation vector. Then, hierarchical softmax will be used to approximate the probability distribution in order to reduce the computing resources needed. Hierarchical softmax assigns the vertices to the leaves of a binary tree, turning the prediction problem into the problem of maximizing the probability of a specific path in the hierarchy. Given that $u_p \in \{d_{i-w}, \cdots, d_{i-1}, d_{i+1}, \cdots, d_{i+w}\}$, that the path from the tree root $\Phi(d_i)$ to the leaf u_p of the binary tree is identified by the sequence of tree nodes $b_1, b_2, \cdots, b_{\lceil \log \rceil}$ and that $b_{\lceil \log \rceil} = u_p$, it follows that $\Pr(u_p|\Phi(d_i))$ can be calculated as in Eq. (4):

$$\Pr(u_p|\Phi(d_i)) = \prod_{r=1}^{\lceil \log \rceil} \Pr(b_r|\Phi(d_i)) \tag{4}$$

Moreover, $\Pr(b_r|\Phi(d_i))$ can be modeled by a binary classifier that is assigned to the parent of node b_r, as shown in Eq. (5):

$$\Pr(b_r|\Phi(d_i)) = \frac{1}{1 + e^{-\Phi(d_i)\cdot\phi(b_r)}} \tag{5}$$

where $\phi(b_r)$ is the representation assigned to the parent of tree node b_r. From Eq. (2), the representation traffic state matrix of the most relevant links can be obtained.

3.2. The road traffic state generation based on the generative adversarial network

The architecture of the generative adversarial network consists of two parts, a generator G and a discriminator D. The generative network generates candidates and the discriminative network evaluates them; the generator G learns to map from a latent space to a data distribution of interest, and the discriminator D distinguishes the candidates produced by the generator from the true data distribution.

The objective of the generator model is to generate target detector data that utilize the representation information gained by GE. The discriminator provides supervision to ensure that the generated data distribution is similar with real data distributions.

To utilize the spatial information from the most relevant detectors, the traffic state data of relevant detectors within the specified sliding window are denoted as $X^t = \{x_1^{(t-l+1):t}, x_2^{(t-l+1):t}, \cdots, x_m^{(t-l+1):t}\}$, where l is the time length of the sliding window. The sliding window contains the temporal and spatial information from the represented neighbor detectors from time $t - l + 1$ to t. The sliding window of the i^{th} detector is denoted $x_i^{(t-l+1):t} = \{x_i^{t-l+1}, x_i^{t-l+2}, \cdots, x_i^t\}$, $i \in (1, 2, \cdots, m)$. The generator can generate the road traffic state data of a detector: $\tilde{X}_{tar}^t = \{\tilde{x}_{tar}^{t-l+1}, \tilde{x}_{tar}^{t-l+2}, \cdots, \tilde{x}_{tar}^t\}$. The architecture of the GAN in our model is shown in Fig. 3.

Our implementation adapts a variant of GAN called Wasserstein GAN (WGAN) (Arjovsky et al., 2017); this variant optimizes earth mover's distance instead of Jensen-Shannon divergence. It solves the problems of GAN training difficulties and collapse mode, which can ensure the diversity of the generated samples. The generator takes one input: the traffic state matrix of the represented detectors within the sliding window.

The objective function in WGAN includes generator loss and discriminator loss. The goal of the generator is to produce the data that follow the distribution of real data. It can be formed as shown in Eq. (6):

$$L_G = -\mathrm{D}(\mathrm{G}(X^t)) + \alpha \cdot L_{cons} \tag{6}$$

where the consistency loss L_{cons} is used to decrease the reconstruction error of the detector data record, and α is the coefficient of the consistency loss.

The discriminator receives either true or generated data and then outputs a scalar. The goal of the discriminator is to maximize the expected difference between $\mathrm{D}(X_{tar}^t)$ and $\mathrm{D}(\tilde{X}_{tar}^t)$, which can be denoted as in Eq. (7):

$$L_D = \mathrm{D}(\mathrm{G}(\mathrm{X}^t)) - \mathrm{D}(X_{tar}^t) \tag{7}$$

where X_{tar}^t is the real historical data within the sliding widows of the detector, and $\tilde{x}_{tar}^t$denotes the generated traffic state data at time t. The whole procedure for this learning method is illustrated in Algorithm 1.

Algorithm 1 (*Generate the traffic state data of target detector utilize WGAN*).

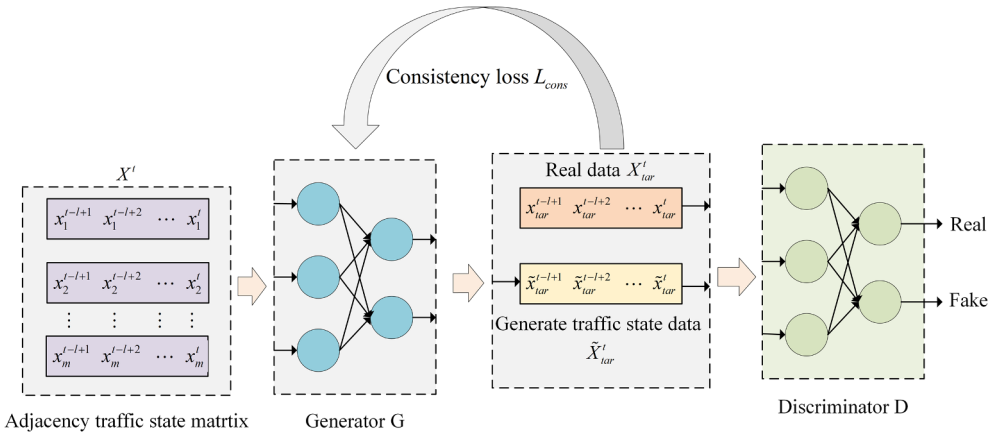

Fig. 3. Architecture of the GAN.

Require: the learning rate $l_r = 0.00005$, number of iterations of the critic per generator iteration $n_d = 5$, the clipping parameter $c = 0.01$, sliding window size l, consistency loss coefficient α, the number of historical data T.

Initialize θ_G, θ_D;
while θ_G has not converged **do**
  for $t = l$ to T **do**
    for $i = 1$ to n_d **do**
      sample neighbor detector data $X^t = \{x_1^{(t-l+1):t}, x_2^{(t-l+1):t}, ..., x_m^{(t-l+1):t}\}$
      evaluate stochastic gradient of θ_D:
      $\delta_{\theta_D} \leftarrow \nabla_{\theta_D}[D(X^t_{tar}) - D(G(X^t))]$
      $\theta_D \leftarrow \theta_D + lr \cdot RMSProp(\theta_D, \delta_{\theta_D})$
      $\theta_D \leftarrow clip(\theta_D, -c, c)$
    end for
    fix θ_D
    sample neighbor detector data $X^t = \{x_1^{(t-l+1):t}, x_2^{(t-l+1):t}, ..., x_m^{(t-l+1):t}\}$
    evaluate stochastic gradient of θ_G :
    $\theta_G \leftarrow -\nabla_{\theta_f}[D(G(X^t)) + \alpha \cdot L_{cons}]$
    $\theta_G \leftarrow \theta_G + lr \cdot RMSProp(\theta_G, \delta_{\theta_G})$
  end for
end while

4. Experiments results

4.1. Data description

To better evaluate the performances of the proposed model, two types of data sets, which include traffic volume and speed data, have been employed in cases study. The traffic volume data from Caltrans PeMS are used in the experiments. Caltrans PeMS has more than 15,000 detectors, which collect traffic data every 30 s (Duan et al., 2016). The volume data collected from May 1, 2014 to June 30, 2014 are used in this study. The collected data are aggregated to 5-min intervals for each detector. To build the road network, 24

Table 1
Selected VDSs and their number in PeMS dataset.

Number	VDS	Number	VDS	Number	VDS
1	763,828	9	716,421	17	760,196
2	773,656	10	716,424	18	716,440
3	760,074	11	716,476	19	716,442
4	760,080	12	760,167	20	760,226
5	716,414	13	716,427	21	760,236
6	760,101	14	716,431	22	716,449
7	760,112	15	716,433	23	718,155
8	716,419	16	760,187	24	716,453

vehicle detector stations (VDSs) in district seven are selected, each of which has 17,568 data points. The selected VDSs are shown in Table 1.

The other dataset, which contains traffic speed data, is collected in Seattle area. This dataset includes the traffic speed data of 323 detectors in one year (2015.1.1–2015.12.31), which cover four connected freeways: I-5, I-405, I-90, and SR-520 (Cui et al., 2018). The data collecting interval is 5 min. All the traffic state data are divided into workdays mode and weekends mode.

4.2. Model settings and evaluation criteria

The model setting consists of two parts: the parameters of DeepWalk and the parameters of the GAN. The parameters of DeepWalk on the two data are set as follows: the selected window size $w = 5$; the iterate times of random walk $\eta = 10$; the embedding size $\mathscr{D} = 64$; the number of selected adjacent detectors $m = 4$ (PeMS dataset) and $m = 10$ (Seattle dataset); the length of the random walk $k = 40$ (PeMS dataset) and $k = 100$ (Seattle dataset).

The two data sets have the same parameters of GAN which are set as follows: the generator and the discriminator each have three hidden layers; the numbers of the hidden layer units are [512,256,128]; and the activation function for each hidden layer is the ReLU function. In addition, the output layer's activation function in the generator is a sigmoid function. There is no activation function in the discriminator's output layer. Gaussian noise is applied to the model input, and the size is set to 50 when the influence of noise on model performance is evaluated. 20% of the data is used as a test set to evaluate the performances of the proposed model. The consistency loss coefficient $\alpha = 100$; and the learning rate $lr = 0.00005$.

To evaluate the performance of the proposed method, three criteria have been adopted to measure the error of the generated data: mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE), which are shown as follows:

$$MAE = \frac{1}{K}\sum_{k=1}^{K} |x_{tar}^{k} - \tilde{x}_{tar}^{k}| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K} (x_{tar}^{k} - \tilde{x}_{tar}^{k})^2} \tag{9}$$

$$MAPE = \frac{1}{K}\sum_{k=1}^{K} \left| \frac{x_{tar}^{k} - \tilde{x}_{tar}^{k}}{x_{tar}^{k}} \right| \tag{10}$$

where K represents the total number of traffic flow samples, $\tilde{x}_{tar}^{k}$represents the generated road traffic state, and x_{tar}^{k}represents the real road traffic state.

4.3. Baseline methods

To further validate the performance of the proposed method, we compare it to several baselines: K-nearest neighbor (KNN) (Cai et al., 2016; Xu et al., 2016), backpropagation network (BP) (Smith and Demetsky, 1994), long shortterm memory (LSTM) (Hochreiter and Schmidhuber, 1997), Deeptrend2.0 (Dai et al., 2019) and Bayesian Gaussian CANDECOMP/PARAFAC (BGCP) tensor decomposition (Chen et al., 2019). For the KNN model, the number of neighbors k_{ne} is set to ten, which gives best performance among all values of k_{ne} from 2 to 25. In the BP neural model, the hidden layer structure is [512,256,128]. In the LSTM model, the hidden layer structure is [512,256]. In the Deeptrend 2.0 model, the filter size of Conv1 and fully connected layers is 3×3, the number of hidden layers is 4 and the number of kernels in each hidden layer is 128. In the BGCP model, third-order tensor representation (detector $\times$ day $\times$ time interval) is utilized in this study. The baselines models contain of machine learning methods and advanced traffic state estimation methods, e.g, Deeptrend 2.0 model. The proposed model and all baselines except KNN, due to the results of KNN are invariant when the k_{ne} is determined, have been tested 10 times repeatly. All the models only utilize data from neighbor detectors to generate the traffic state of the target detector.

4.4. Experiments on PeMS dataset

4.4.1. Volume estimation of single detector

The goal of the DeepWalk is to find the most relevant detectors of the target detector in spatial correlation through embedding the network into low-dimensional space. Generally, the adjacent detectors with strong spatial correlation of the target detector can be selected by DeepWalk. As described in Section 3 in Eq. (2), the Deepwalk can maximize the probability that any detector appears in the context to find the most relevant detectors with strong spatial correlations of the target detector. Besides, to further illustrate the usefulness of graph embedding, the performances of the proposed model with graph embedding (GE-GAN) and without graph embedding model (GAN) are compared in the experiments. In the experiments of GAN, the most relevant detectors are selected manually from the adjacent detectors, while the detectors with strong spatial correlations are selected by Deepwalk in the experiments of GE-GAN. The comparison results are shown in Table 2. As shown in Table 2, the error of GE-GAN is much smaller than GAN, which can verify the effectiveness of DeepWalk.

To further evaluate the influence of noise on model performance, the comparative experiments are conducted. And VDS 760112 is

Table 2
The average error comparison before and after applying graph embedding in PeMS dataset.

Adding GE	Workdays			Weekends		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
Before	32.04	23.25	6.70	23.44	18.00	5.56
After	20.07	14.12	3.92	15.97	11.92	3.57

selected as the target road traffic state detector. In one case, the input of the generator is only the traffic state matrix of adjacent represented links within the sliding window, while in the other case, the inputs of the generator are both noise and the traffic state matrix of adjacent represented links within the sliding window. The rest of the generator structure is the same. The error measurements of the results before and after adding noise to the generator input are shown in Table 3. The sliding time window is an hour because enough spatial information can be obtained in an hour to determine the effect of sliding window size on experimental results.

From Table 3, it is obvious that the performance of the model before the addition of noise is always better than the performance after the addition of noise for all sliding window sizes. Compared with the results obtained before noise addition, the average outcome after noise addition has an increase of 17.11% in MAE, of 14.2% in RMSE and of 22.01% in MAPE. The cause of poor performance after noise addition is that the noise has a certain impact on the spatial information. Among different sliding window sizes, the best results are seen when the sliding window size is 12. The error will increase with increasing sliding size. The reason is that if the sliding window is too large, it will lead to excessive redundant information and will make the error gradually increase. The results when other detectors are shown in Table 4, where the sliding window size is 12 and there is no noise in the input of generator.

As in the above comparative experiment on noise and window size, the sliding window size is also set to 12 when the performance of the proposed GE-GAN model is compared with that of other models. The error of different models in the last week is shown in Fig. 4. It is obvious that GE-GAN has the smallest error of the various models. The LSTM model has the worst performance of all the models. The reason is considered to be that the LSTM model is better at processing continuous time series. However, in this case, the spatial data of adjacent detectors are selected to generate the traffic state data of the target detector. There is no strong relationship between the input and the output of the time series. In addition, the reason that the KNN has the similar performance with the BP model is that the traffic data have strong regularity after dividing the road traffic state data into workdays mode and weekends mode. Moreover, The Deeptrend2.0 model is conducted to generate the data of road traffic state detector through an image built with detectors, which make the pixels of strongly correlated detectors be close. And the results show that the performance of proposed model is basically best. Especially, the results of BGCP model are not shown due to its large error. The BGCP model is utilized to impute the missing data by Bayesian Gaussian tensor decomposition. The road traffic state estimation in real time can be considered as imputation of missing data, which means that the information of the detector is completely unknown in the application of BGCP.

4.4.2. Volume estimation of multi-detectors at the same time

In this part, some experiments are conducted to generate the traffic state data of multi detectors at the same time. Firstly, the number of the detectors are denoted as M and the set of most relevant adjacent detectors for each targer detector is denoted as m_i $(i = 1, 2 \cdots M)$. The last collection of the relevant adjacency detectors denoted as $\Theta = \{m_1 \cap m_2 \cap \cdots \cap m_M\}$. If there is no overlap from the set of most relevant detectors for target detectors as Θ is a empty set, it is essential to increase the sliding window size in the process of DeepWalk. Finally, the traffic state of target detectors can be generated with the traffic state matrix of the relevant adjacency detectors. From the experiments for multi detectors, the sliding window size is also 12 and there is no noise in the input of generator. The results of the generated road traffic states of different detectors are shown in Table 5.

In Table 5, the error which include RMSE, MAE, MAPE are obtained from the average error of multi-detectors. The relationship of

Table 3
Error comparison before and after adding noise within different slidin window size.

Days	Error	Adding noise	Sliding window size				
			6	12	18	24	30
Workdays	MAE	Before	**13.11**	**12.71**	**13.27**	**14.84**	**14.77**
		After	16.32	16.21	16.95	17.08	17.75
	RMSE	Before	**18.40**	**18.00**	**18.60**	**19.71**	**19.80**
		After	21.52	21.53	22.55	22.74	23.15
	MAPE (%)	Before	**4.11**	**3.91**	**4.18**	**4.57**	**4.56**
		After	5.22	5.23	5.58	5.55	5.75
Weekends	MAE	Before	**13.02**	**12.27**	**13.60**	**12.93**	**14.55**
		After	14.95	15.57	16.06	15.80	16.26
	RMSE	Before	**17.28**	**16.13**	**17.76**	**16.78**	**18.80**
		After	18.79	19.78	20.25	20.19	20.70
	MAPE (%)	Before	**3.83**	**3.82**	**4.07**	**4.06**	**4.45**
		After	4.91	5.08	5.37	5.24	5.36

Table 4
The performance of different detectors.

VDS	Workdays			Weekends		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
716,449	10.46	7.56	5.16	9.39	7.50	5.32
716,476	13.20	9.05	4.54	11.66	8.32	4.47
760,226	12.82	9.16	**3.12**	11.50	8.44	**3.28**
716,442	13.58	9.72	3.72	13.42	9.98	4.54
760,167	17.02	10.50	5.21	14.30	9.97	4.96
760,196	16.47	10.57	4.26	15.39	9.81	4.99
760,236	14.99	11.12	3.49	15.18	11.41	3.98
760,187	17.02	11.92	4.25	14.51	10.66	4.40
760,101	22.96	13.88	4.20	15.38	11.70	3.65

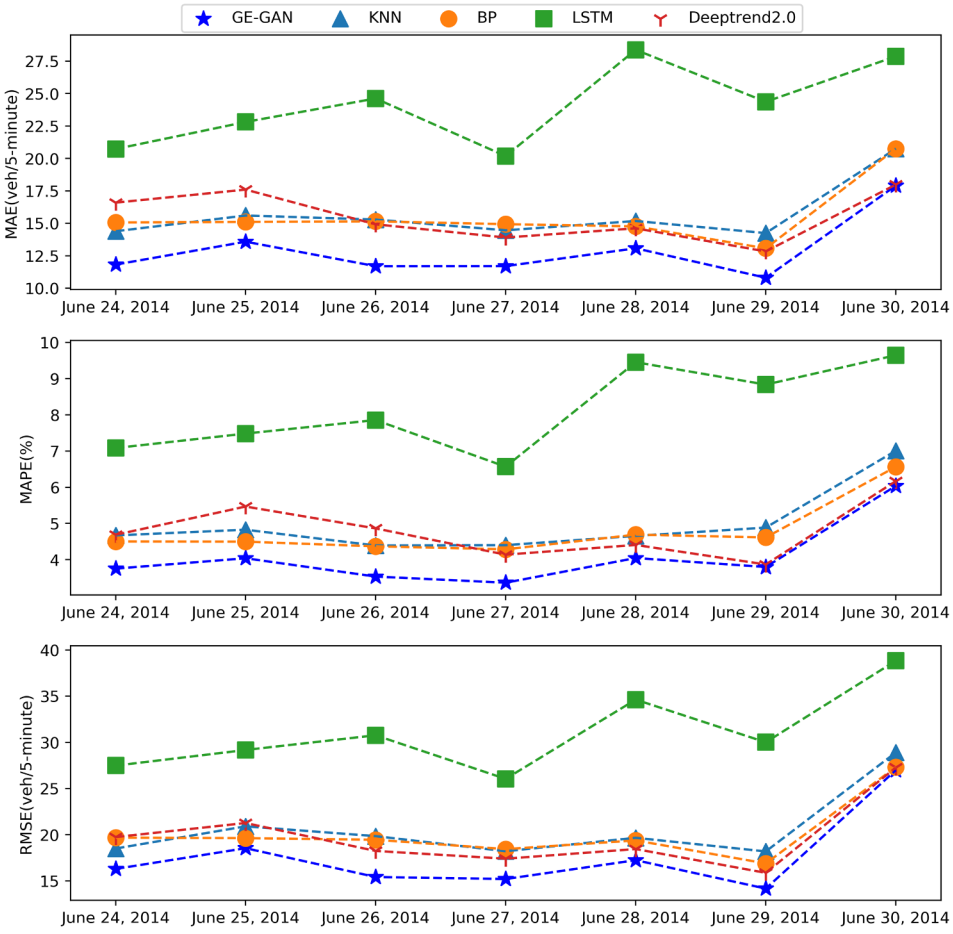

Fig. 4. The error from June 24, 2014 to June 30, 2014 in different models on single detector.

multi-detectors can be divided into 2 parts: adjacent and not adjacent. The results indicate that the proposed model have better performance when multi-detectors are not adjacent. The reason is considered to be that it will lose some spatial informance to generate road traffic state with adjacent detectors.

Compared to the experiments which generate the traffic state of single detector within, the experiments for multi-detectors have the higher MAPE in all situation. The reason for consideration is that more spatial information can be obtained in the experiments for single detector, while the spatial information has certain overlap in the experiments of multi detectors.

The performance of the proposed GE-GAN model is compared with other models when generating road traffic state of multi detectors. The results of different models in the last week are shown in Fig. 5. In addition, the error is the average error which calculated from all selected detectors. In Fig. 5, it is obvious that GE-GAN has the best performance. The LSTM has the worst performance because of its limitations of processing spatial information. KNN and BP model have almost similar performance.

Table 5
The average error of generated data for different detectors.

Relation of VDSs	Objective numbers of VDSs	Workdays			Weekends		
		RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
Adjacent	6,7	23.91	15.03	4.33	18.32	13.96	4.18
	11,12	16.43	11.49	5.63	15.71	11.35	5.80
	16,17	19.58	13.92	5.16	17.84	12.11	5.74
	20,21	14.47	10.73	3.55	18.75	12.27	4.92
	21,22	17.92	13.19	6.04	29.89	18.6	10.22
Average error		**18.46**	**12.87**	**4.94**	**20.10**	**13.66**	**6.17**
Not adjacent	7,11	16.80	12.12	4.48	15.59	11.51	4.69
	19,22	12.18	8.83	4.53	11.59	8.75	4.81
	16,20	14.93	10.35	3.67	14.08	10.20	4.39
	6,9	21.14	15.45	4.61	17.88	13.27	4.25
	20,22	12.04	8.80	4.43	11.51	8.71	4.72
Average error		**15.42**	**11.11**	**4.34**	**14.13**	**10.49**	**4.57**

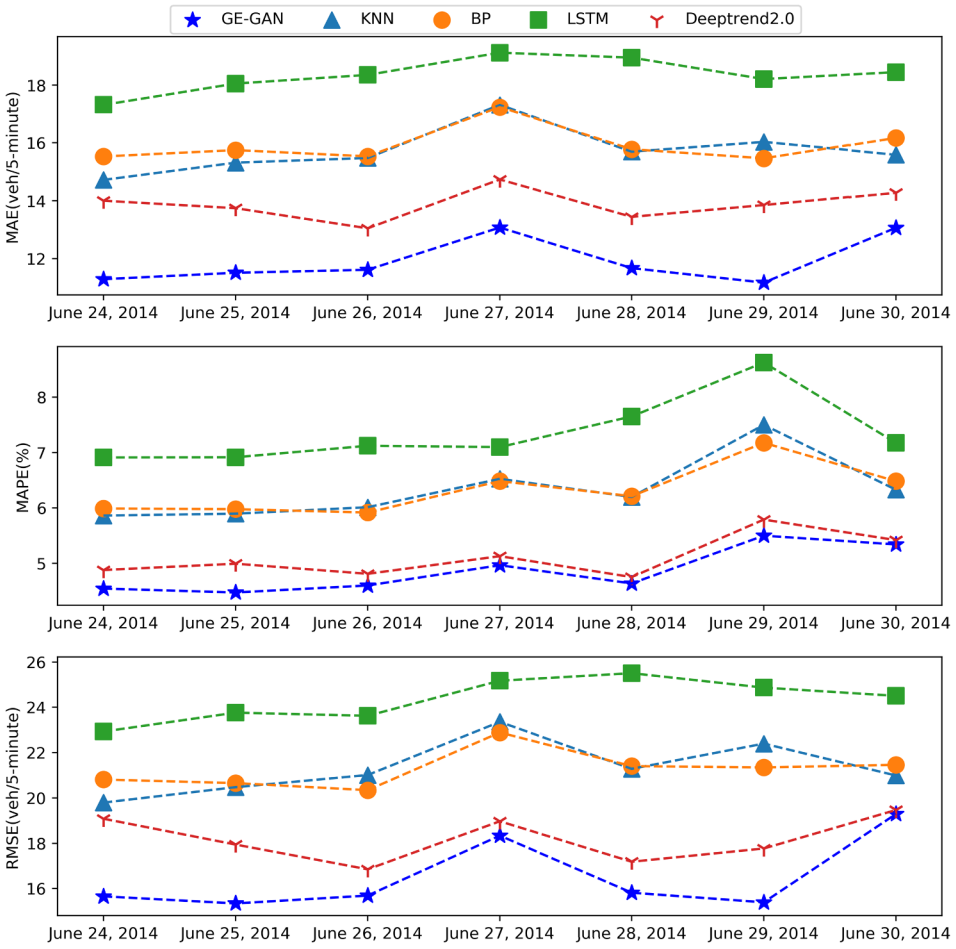

Fig. 5. The average error of all selected multi-detectors from June 24, 2014 to June 30, 2014 in different models.

Compared with the results of volume estimation of single detector, the performance of the Deeptrend2.0 for volume estimation of multi detectors is much better than the KNN and BP model. And the results of BGCP are also not shown, due to the same reason. Furthermore, there are no dramatic fluctuations in the performance of proposed model.

The curves of loss functions of generator and discriminator are shown in Fig. 6. The loss of the generator gradually decreases in the process of model training. The loss of the discriminator changes around 0 which indicates the discriminator can't distinguish the

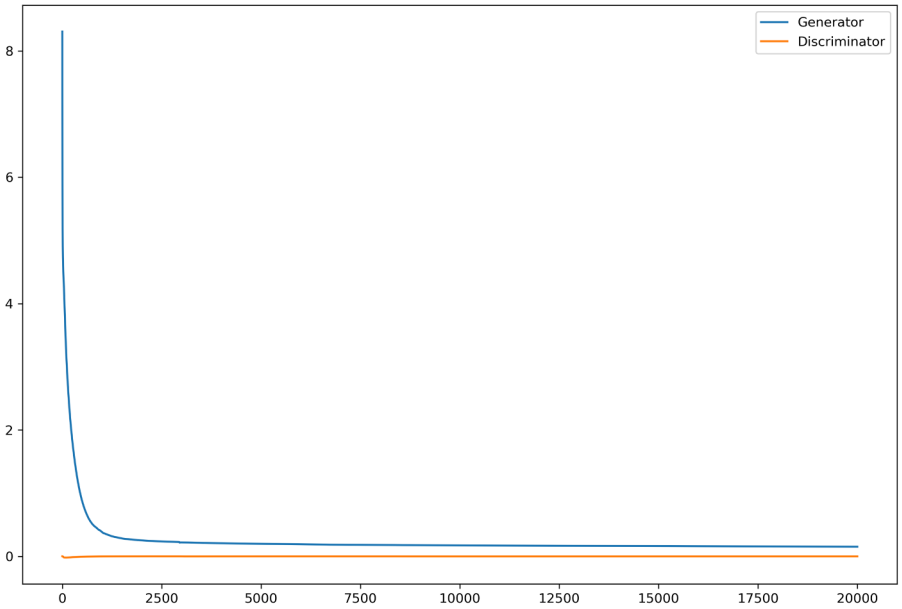

Fig. 6. The curves of loss functions of generator and discriminator.

generated data from the true data. It is obvious that WGAN can solve the training difficulties of original GAN and makes the model converge quickly.

4.5. Experiments on Seattle data

The experiments for Seattle dataset are divided into two parts: speed estimation of single detector and speed estimation of multi-detectors at same time under workdays mode and weekends mode.

4.5.1. Speed estimation of single detector

To illustrate the usefulness of graph embedding, the performances of the proposed model with graph embedding (GE-GAN) and without graph embedding model (GAN) are also compared in the experiments, and the results are shown in Table 6. The error of model with graph embedding are also smaller than the error of model without graph embedding. The comparative results of the two data sets can both verify the effectiveness of graph embedding.

From the experiments on PeMS dataset, the performance of the model before the addition of noise is always better than the performance after the addition of noise. Thus, there is no noise in the input of model on Seattle data. And partial results are shown in Table 7, in which different mileposts represent different detectors. From in Table 7, it is obvious that the MAPE is less than 10% for all detectors.

To further evaluate the performance of the proposed model, the average error of different detectors obtained by different models are shown in Fig. 7. Fig. 7(a) shows the average error of different models on workdays, while the average error of different models on weekends are shown in Fig. 7(b). As depicted in Fig. 7, the proposed model has the best performance and the error of the proposed model on the workdays are smaller than the error on the weekends. The KNN and BP model also have the similar performances as the experiments on the PeMS dataset. However, the LSTM model has a better performance than KNN and BP. The Deeptrend2.0 is the second best, due to more spatial information when building the image with more detectors in the Seattle dataset.

4.5.2. Speed estimation of multi-detectors at the same time

The speed estimation of multi detectors at the same time is also conducted. The goal of these experiments is to increase the efficiency of data generation. The selected detectors are also divided into two parts: adjacent and not adjacent. The average error of

Table 6
The average error comparison before and after adding graph embedding in Seattle dataset.

Adding GE	Workdays			Weekends		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
Before	4.83	3.27	7.18	5.81	3.45	9.27
After	3.35	2.06	4.65	4.01	2.23	6.35

Table 7
The performance of different detectors in Seattle dataset.

Milepost	Workdays			Weekends		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
d005es15036	2.60	1.67	3.45	3.21	1.82	4.63
d005es16396	4.90	3.56	8.34	5.11	3.52	7.68
i005es16466	2.18	2.05	4.80	2.95	2.12	4.08
i520es00624	3.39	2.67	4.40	3.15	2.47	4.06
i520es00028	2.80	2.09	8.43	2.86	2.15	7.21
i090es00720	3.27	2.26	4.82	3.21	2.11	4.34
d090es00278	4.85	2.75	7.22	4.48	2.84	6.44
d090es00720	2.46	1.87	4.43	2.66	1.95	4.43
d405es01079	5.93	2.48	9.11	5.14	2.24	7.27
d405es01465	4.56	2.96	8.96	4.48	2.97	7.61

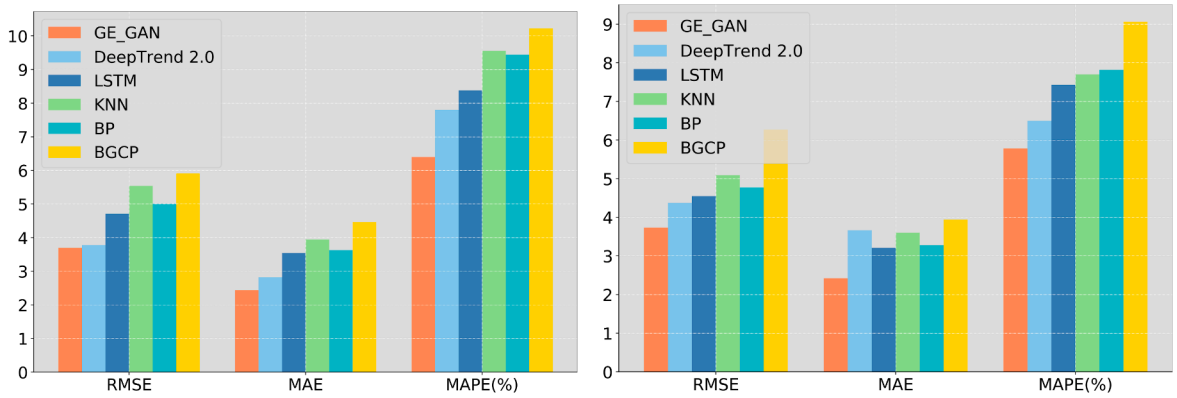

Fig. 7. The average error of single detector of different models.

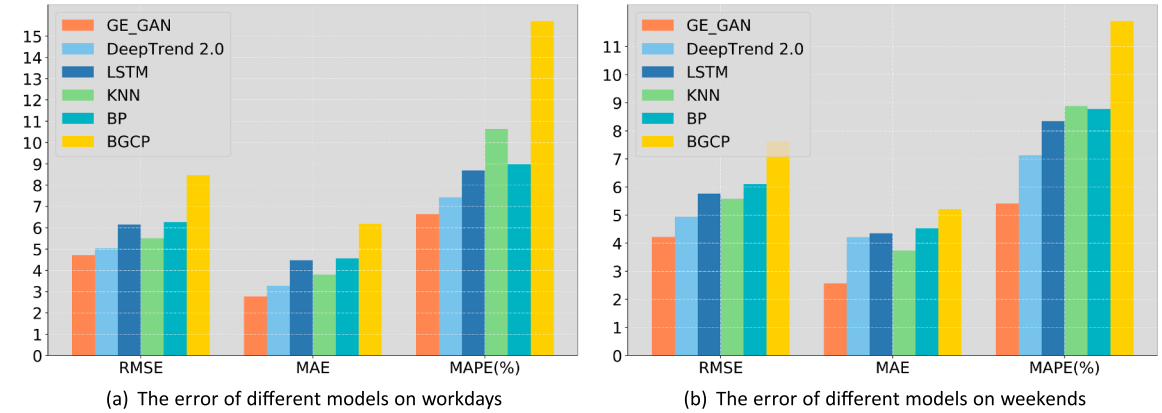

Fig. 8. The average error of multi-detectors of different models.

multi-detectors under different models are shown in Fig. 8. Fig. 8(a) shows the error of all models on workdays and the error of all models on weekends are shown in Fig. 8(b).

The proposed model also has the best performance in all models, and the error change little compared with the results of single detector. Hence, the GE-GAN can applied in speed generation of single detector and multi detectors at the same time.

5. Discussions and conclusions

In this paper, a novel deep learning framework for road traffic state estimation was proposed. We constructed the road network and chose the most relevant represented detectors of the target detectors through graph embedding (GE). The traffic state data of the

selected detectors were used as the input of GAN, which generated the traffic state data of the target detector. A comparison of the performance of the GE-GAN model with that of KNN, BP, Deeptrend2.0, BGCP and LSTM was provided. The results showed that the performance of GE-GAN was better than that of other models.

For future work, we will apply our approach to a more complex road network and we will also try to combine this model with other network models in the hope of improving performance. In addition, different data sets will be used in our model to further evaluate the model performance.

CRediT authorship contribution statement

Dongwei Xu: Conceptualization, Methodology, Writing - review & editing, Funding acquisition, Supervision. **Chenchen Wei:** Data curation, Writing - original draft, Software. **Peng Peng:** Visualization, Investigation. **Qi Xuan:** Funding acquisition, Supervision. **Haifeng Guo:** Resources, Validation.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (61903334, 61973273), in part by the Zhejiang Provincial Natural Science Foundation under Grant(LQ16E080011, LR19F030001, LY20E080023), and in part by the China Postdoctoral Science Foundation under Grant 2018M632501.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.trc.2020.102635.

References

Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2007. Nonlinear kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. IEEE Trans. Intell. Transp. Syst. 8, 661–670.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.

Bahadori, M.T., Yu, Q.R., Liu, Y., 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. Adv. Neural Inform. Process. Syst. 3491–3499.

Bhagat, S., Cormode, G., Muthukrishnan, S., 2011. Node classification in social networks. In: Social Network Data Analytics. Springer, pp. 115–148.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., Sun, J., 2016. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transport. Res. Part C: Emerg. Technol. 62, 21–34.

Chen, X., He, Z., Sun, L., 2019. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transport. Res. Part C: Emerg. Technol. 98, 73–84.

Cui, Z., Ke, R., Wang, Y., 2018. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. CoRR abs/1801.02143. URL: http://arxiv.org/abs/1801.02143, arXiv:1801.02143.

Dai, X., Fu, R., Zhao, E., Zhang, Z., Lin, Y., Wang, F.Y., Li, L., 2019. Deeptrend 2.0: a light-weighted multi-scale traffic prediction model using detrending. Transport. Res. Part C: Emerg. Technol. 103, 142–157.

Ding, C.H., He, X., Zha, H., Gu, M., Simon, H.D., 2001. A min-max cut algorithm for graph partitioning and data clustering. In: Proceedings 2001 IEEE International Conference on Data Mining. IEEE, pp. 107–114.

Duan, Y., Lv, Y., Liu, Y.L., Wang, F.Y., 2016. An efficient realization of deep learning for traffic data imputation. Transport. Res. Part C: Emerg. Technol. 72, 168–181.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Adv. Neural Inform. Process. Syst. 2672–2680.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690.

Lee, D., Jung, S., Cheon, Y., Kim, D., You, S., 2019. Demand forecasting from spatiotemporal data with graph networks and temporal-guided embedding. arXiv preprint arXiv:1905.10709.

Lee, S., Fambro, D.B., 1999. Applicationofsubsetautoregressiveintegratedmovingaveragemodelforshort-termfreewaytrafficvolumeforecasting. Transp. Res. Rec. 1678, 179–188.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D., 2017. Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547.

Li, L., Su, X., Wang, Y., Lin, Y., Li, Z., Li, Y., 2015. Robust causal dependence mining in big data network and its application to traffic flow predictions. Transport. Res. Part C: Emerg. Technol. 58, 292–307.

Li, Z., Jiang, S., Li, L., Li, Y., 2019. Building sparse models for traffic flow prediction: An empirical comparison between statistical heuristics and geometric heuristics for bayesian network approaches. Transportmetrica B: Transport Dyn. 7, 107–123.

Liang, P., Bose, N., 1996. Neural Network Fundamentals with Graphs, Algorithms and Applications. Mac Graw-Hill.

Liang, Y., Cui, Z., Tian, Y., Chen, H., Wang, Y., 2018. A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation. Transp. Res. Rec. 2672, 87–105.

Liben-Nowell, D., Kleinberg, J., 2007. The link-prediction problem for social networks. J. Am. Soc. Inform. Sci. Technol. 58, 1019–1031.

Lin, Y., Dai, X., Li, L., Wang, F.Y., 2018. Pattern sensitive prediction of traffic flow based on generative adversarial framework. IEEE Trans. Intell. Transp. Syst. 20, 2395–2400.

Maaten, L.V.D., Hinton, G., 2008. Visualizing data using t-sne. J. Mach. Learn. Res. 9, 2579–2605.

Mihaylova, L., Boel, R., 2004. A particle filter for freeway traffic estimation. In: 2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601), IEEE. pp. 2106–2111.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Adv. Neural Inform. Process. Syst. 3111–3119.

Nanthawichit, C., Nakatsuji, T., Suzuki, H., 2003. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. Transp. Res. Rec. 1855, 49–59.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining, ACM. pp. 701–710.
Pueboobpaphan, R., Nakatsuji, T., Suzuki, H., 2007. Unscented Kalman filter-based real-time traffic state estimation. Technical Report.
Qu, L., Li, L., Zhang, Y., Hu, J., 2009. Ppca-based missing data imputation for traffic flow volume: a systematical approach. IEEE Trans. Intell. Transp. Syst. 10, 512–522.
Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2008. The graph neural network model. IEEE Trans. Neural Netw. 20, 61–80.
Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R., 2017. Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2107–2116.
Smith, B.L., Demetsky, M.J., 1994. Short-term traffic flow prediction: neural network approach. Transp. Res. Rec.
Tak, S., Woo, S., Yeo, H., 2016. Data-driven imputation method for traffic data in sectional units of road links. IEEE Trans. Intell. Transp. Syst. 17, 1762–1771.
Wang, C., Ran, B., Yang, H., Zhang, J., Qu, X., 2018. A novel approach to estimate freeway traffic state: parallel computing and improved kalman filter. IEEE Intell. Transp. Syst. Mag. 10, 180–193.
Wang, Y., Papageorgiou, M., Messmer, A., 2007. Real-time freeway traffic state estimation based on extended kalman filter: a case study. Transport. Sci. 41, 167–181.
Wu, Y., Tan, H., Li, Y., Zhang, J., Chen, X., 2018. A fused cp factorization method for incomplete tensors. IEEE Trans. Neural Netw. Learn. Syst. 30, 751–764.
Xu, D., Dai, H., Wang, Y., Peng, P., Xuan, Q., Guo, H., 2019. Road traffic state prediction based on a graph embedding recurrent neural network under the scats. Chaos: Interdisciplinary J. Nonlinear Sci. 29. https://doi.org/10.1063/1.5117180.
Xu, D., Dong, H., Jia, L., Qin, Y., 2012. Virtual speed sensors based algorithm for expressway traffic state estimation. Sci. China Technol. Sci. 55, 1381–1390.
Xu, D., Wang, Y., Li, H., Qin, Y., Jia, L., 2015a. The measurement of road traffic states under high data loss rate. Measurement 69, 134–145.
Xu, D., Wang, Y., Peng, P., Beilun, S., Deng, Z., Guo, H., 2018. Real-time road traffic state prediction based on kernel-knn. Transportmetrica A: Transport Science 1–15.
Xu, D.W., Dong, H.H., Jia, L.M., Tian, Y., 2014. Road traffic states estimation algorithm based on matching of regional traffic attracters. J. Central South Univ. 21, 2100–2107.
Xu, D.W., Dong, H.H., Li, H.J., Jia, L.M., Feng, Y.J., 2015b. The estimation of road traffic states based on compressive sensing. Transportmetrica B: Transport Dyn. 3, 131–152.
Xu, D.W., Wang, Y.D., Jia, L.M., Li, H.J., Zhang, G.J., 2016. Real-time road traffic states measurement based on kernel-knn matching of regional traffic attractors. Measurement 94, 862–872.
Xu, D.W., Wang, Y.D., Jia, L.M., Qin, Y., Dong, H.H., 2017. Real-time road traffic state prediction based on arima and kalman filter. Front. Inform. Technol. Electron. Eng. 18, 287–302.
Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S., 2006. Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. 29, 40–51.
Yu, B., Yin, H., Zhu, Z., 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875.
Yu, J.J.Q., Gu, J., 2019. Real-time traffic speed estimation with graph convolutional generative autoencoder. IEEE Trans. Intell. Transp. Syst.
Zhang, K., Jia, N., Zheng, L., Liu, Z., 2019. A novel generative adversarial network for estimation of trip travel time distribution with trajectory data. Transport. Res. Part C: Emerg. Technol. 223–244.
Zheng, Z., Yang, Y., Liu, J., Dai, H.N., Zhang, Y., 2019. Deep and embedded learning approach for traffic flow prediction in urban informatics. IEEE Trans. Intell. Transp. Syst.