2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Traffic Flow Prediction with Parallel Data*

Yuanyuan Chen, Yisheng Lv, Xiao Wang and Fei-Yue Wang

Abstract— **Traffic prediction is an elemental function of Intelligent Transportation Systems, and accurate and timely prediction is of great significance to both traffic management agencies and individual drivers. With the development of deep learning and big data, deep neural networks (DNN) achieve superior performances in traffic prediction. Developing DNN prediction models needs large scale and diverse data, however, it is costly to collect large volume of accurate traffic data. In this paper, we propose to use small volume of real traffic data and large volume of synthetic traffic data to developing traffic prediction models. The evolving of parallel system paradigm for traffic prediction and the algorithm to incrementally train traffic data generation models and traffic prediction models are presented. We use an improved generative adversarial networks to generate traffic data, and a stacked long short-term memory model for traffic prediction. Experimental results on a real traffic dataset demonstrate that our method can significantly improve the performance of traffic flow prediction.**

I. Introduction

As the management of transportation transiting from industrial technology to information technology, Intelligent Transportation System (ITS) has made transportation safer and more efficient [1]–[3]. However ITS has encountered technology bottlenecks with the development of societies and transportation systems. Traffic prediction with high accuracy is one the primary obstacles. Fortunately, with big data and deep learning, the performances of traffic prediction are continuously improved. However, it is time-consuming and expensive to collect large-scale traffic data. And more often, traffic data collected from physical sensors in the real world are missing or corrupted due to detector failures. Therefore we apply the parallel data paradigm that has been proposed to use synthetic and real data for data mining and data-driven processes [4]–[6]. The idea of using parallel data, i.e. synthetic data and real data, to develop a model has become appealing since the synthetic data can augment data automatically complementary to the original data, which provide a way to get big data cheaply and help train robust and powerful models [7], [8].

Traffic prediction, as a fundamental part in ITS, aims to estimate target values in the future with observed traffic data. Many methods have been proposed to solve traffic data imputation problem, which can be generally categorized to data-driven methods and simulation-based methods [9]. The simulation-based methods apply dynamic traffic assignment theories to build artificial systems and generate prediction. In this paper, we focus on reviewing the work of short-term traffic flow prediction.

The data-driven methods include parametric regression, nonparametric regression and neural network approaches [10], [11]. Autoregressive Integrated Moving Average (ARIMA) and its variants Kohonen ARIMA, subset ARIMA, ARIMA with explanatory variables and Seasonal ARIMA are typical parametric regression approaches [12], [13]. These methods are easily to apply, but they can't be utilized to predict traffic flows that vary quickly. Kalman filtering method is another parametric regression. It is suitable to model linear system and there exists a decay in prediction [14]. Nonparametric regression methods, including support vector regression (SVR) and k-Nearest neighbors (KNN), have advantages to handle the stochastic and nonlinear characteristics of traffic flow. SVR utilize kernel function to map the historical traffic flow into feature space, and then apply linear transformation to feature extracted to obtain the prediction [15]. The KNN approach aims to find closest data points and take them as the prediction [16]. The nonparametric method is easy to apply and extend for traffic flow prediction in different areas. With the development of deep learning theories and techniques, neural network approaches achieve better performances than traditional parametric and nonparametric regression methods. Stacked autoencoder model, LSTM model and CNN model are proposed to learn the features of traffic flow series, and these deep neural network models achieve superior performances [17]–[19].

Deep neural networks show great potentials in traffic prediction. While to develop traffic prediction models with high accuracy, traffic data must be large and diverse, and it is very expensive and even impossible to collect big and accurate data. Therefore, we investigate the alternative easy-access artificial data to be used in traffic prediction. In this paper, we use GANs to generate traffic data and further apply the synthetic data with real data for traffic prediction.

The rest of this paper is organized as follows. Section II introduces the basics of GANs and the improved GANs to generate traffic data. Section III presents our approach to predict traffic data with parallel data paradigm. Section IV provides the experimental results to verify the effectiveness of our approach. Section V concludes this paper.

*This work was supported in part by National Natural Science Foundation of China (61533019, 61702519), and in part by Beijing Municipal Science & Technology Commission (Z181100008918007).

Yuanyuan Chen, Yisheng Lv, Xiao Wang and Fei-Yue Wang are with State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. They are also affiliated with Qingdao Academy of Intelligent Industries Qingdao, Shandong, 266109, China. (e-mail: ychen5133@ia.ac.cn, yisheng.lv@ia.ac.cn, x.wang@ia.ac.cn, feiyue@ieee.org)

Yisheng Lv is the corresponding author of this paper.

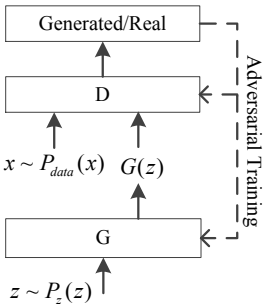


Fig. 1. Architecture of Generative Adversarial Networks

II. Traffic Data Generation

A. The Basics of GANs

GANs, trained in an adversarial manner, are a class of generative models to generate data resembling the distribution of real data, which have been applied in generating image, videos and text [20], [21]. The architecture of standard GAN is illustrated in Fig. 1. GAN comprises two competing components, named as the generator G and the discriminator D, respectively. Typical model for G and D are deep neural networks. The generator G learns to map the given samples from a standard random distribution to the samples whose distribution is resembling the distribution of real data. The discriminator D takes in samples both drawn from the real data distribution P_{data} and generated by the generator G. The generator G tries to fool the discriminator D that the generated samples are the same as real data samples, while the discriminator D tries to distinguish the generated samples from the real data samples. With well adversarial training, ideally, the generator generate samples that can not be recognised by the discriminator D.

The generator G and the discriminator D are simultaneously trained as a minimax two-player game. In practice, G and D are trained in an alternating manner. Formally, let $P_{data}(x)$ as the input standard distribution and $P_z(z)$ as the training data distribution, the minimax objective of GANs is defined as

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

B. Improved GANs to Generate Traffic Data

The GANs framework enables the generator to generate samples that are likely drawn from a certain distribution [22], thus a generator is well trained once the distribution of synthetic data is similar to or same with the distribution of real data regardless of the dependance on the data points within a sample. However traffic flow series is time-serial dependency, it is not a proper way to directly apply standard GANs in traffic data generation, especially based on a small-scale dataset. Intuitively, due to the time-serial dependence in traffic flow data, the latent code fed into the generator should also embed such dependence. Inspired by this idea, it is convenient to use the original real data as the latent code. As shown in Fig. 2, the real data are fed into generator G. The discriminator D takes real data and synthetic data, and

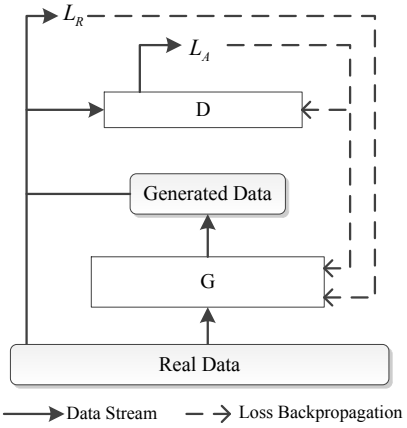


Fig. 2. Architecture of the Improved GANs to Generate Traffic Data

assigns a label to indicate its input coming from real data or synthetic data. The probability of assigning the correct label is defined as L_A, which is adversarial loss same with the loss function of standard GANs and used to optimize the generator G and the discriminator D. Besides, as the purpose of traffic data generation is to augment the real traffic data so that it is necessary to generate samples slightly differing from the corresponding real samples. We proposed to apply a representation loss L_R, as illustrated in the bottom of Fig. 2, to decrease the reconstruction error between real data samples and synthetic data samples. And the representation loss, only utilized to optimize the generator G, is defined as

$$L_R = V(G(x), x) \quad (2)$$

where V is a function to map the difference between the input sample x and the generated sample $G(x)$. In our experiments, we use the ℓ_1 norm, and the representation loss becomes

$$L_R = \|G(x) - x\|_1 \quad (3)$$

III. Parallel Traffic Prediction

A. The Evolving of Parallel Paradigm for Traffic Prediction

Parallel transportation management systems (PtMS) is a new mechanism for conducting operations of transportation systems [1], [23], [24]. In parallel transportation system, there exist one or more artificial transportation systems (ATS), which are developed according to different purposes. In the paper, we propose to improve traffic prediction by developing ATS or generative models that are designed to generate artificial traffic data [25], [26].

In parallel traffic prediction, the real transportation system (RTS), the ATS and the traffic prediction (TP) models are evolving over time as shown in Fig. 3. At the initial phase, the data in RTS is low-volume, so the ATS is trained with limited data and generate artificial data to augment the real data. Then the TP is trained based on the real data and the synthetic data. As time goes by, there are more data archived and the ATS the TP are incrementally refined. The process of training traffic prediction model based on parallel data is summarized in Algorithm 1.

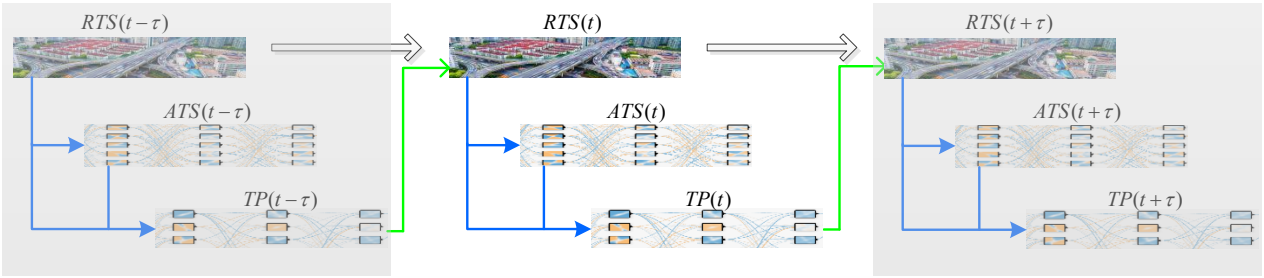


Fig. 3. Traffic Prediction based on Parallel Transportation Systems

Algorithm 1 Incremental Training of Traffic Prediction Model based on Parallel Transportation Systems

Input: real traffic dataset $X_{\tau,n} = \{x_n\}$, GANs (generator G$(t-\tau)$ and discriminator D$(t-\tau)$), traffic prediction model $TP(t-\tau)$, Historical Steps H, Predictive Steps P
Output: traffic prediction model $TP(t)$

//refine GANs
training discriminator with objective function L_A
training generator with objective function $L_A + \lambda L_R$
//refine traffic prediction model
generating serial data $Samples^{RT}, Target^{RT}$ = TIMEGENERATOR$(X_{\tau,n}, H, P)$
for $i = 1$ to T **do**
  if $i = 1$ **then**
    $X^G = G(X_{\tau,n})$
  else
    $X^G = G(X^G)$
  end if
  generating serial data $Samples^G$, $Target^G$ = TIMEGENERATOR(X^G, H, P)
  combing $Samples^{RT}$, $Target^{RT}$ and $Samples^G$, $Target^G$ into $Samples^{Train}$, $Target^{Train}$
  batch-training TP with $Samples^{Train}$, $Target^{Train}$
end for
//preprocessing
function TIMEGENERATOR(X, H, P)
  Reshape $X_{\tau,n}$ into $X_{1,\tau n}$
  for $j = 1$ to $\tau \times n - H - P$ **do**
    $Targets[j] \leftarrow X_{1,\tau \times n}[j + H + P - 1]$
    $Samples[j] \leftarrow X_{1,\tau \times n}[j : j + H - 1]$
  end for
  return $Samples$, $Targets$,
end function

B. Traffic Prediction Model based on LSTM Recurrent Network

Traffic prediction can be transformed into the task of time series prediction, which can be tackled by LSTM recurrent network. The LSTM block consists of cell, input gate, forget gate and output gate [27], [28]. The gate architecture is for removing information from or adding information to cell state and the cell state is only changed by linear interactions. This design helps to tackle the vanishing gradient problem in standard recurrent neural network as it enables the cell to store and read long range contextual information [29]. To obtain cell state C_t and hidden state h_t, we should successively compute the output of the forget gate:

$$f_t = \phi(W_f[h_{t-1}, v(w_t)] + b_f) \tag{4}$$

the output of the input gate:

$$i_t = \phi(W_i[h_{t-1}, v(w_t)] + b_i) \tag{5}$$

and output of the output gate

$$o_t = \phi(W_o[h_{t-1}, v(w_t)] + b_o) \tag{6}$$

where W_f , W_i and W_o are weight matrices of the forget gate, the input gate and the output gate respectively, and b_f , b_i and b_o are their bias vectors. ϕ is the gate activation function and is usually the sigmoid function. Then compute the cell state:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \mu(W_C[h_{t-1}, v(w_t)] + b_C) \tag{7}$$

and the hidden output:

$$h_t = o_t \otimes \sigma(C_t) \tag{8}$$

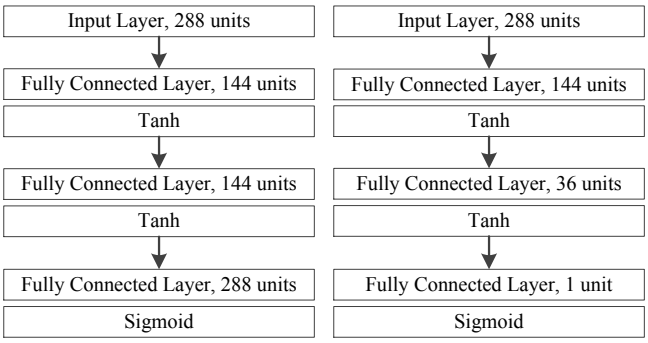
where μ and σ are activation functions, and they are usually tanh, and $\otimes$ represents point-wise multiplication. The complete sequence of cell states and hidden outputs can be computed by applying (4), (5), (6), (7) and (8) recursively from $t = 1$ to $t = T$.

As the features of traffic data series are complicated, we apply stacked LSTM layers to extract deep and abstract features. Historical series $x_{t-H+1}, x_{t-H+2}, ..., x_t$ are fed into the first LSTM layer and the returned sequences are fed into next LSTM layer. The returned sequences of top LSTM layer are flattened into a 1D tensors that are fed into a FCN. And the output of top FCL is the prediction.

IV. EXPERIMENTS

A. Dataset and Experiments Settings

1) Dataset Description: In this paper, we evaluate the proposed method on traffic flow datasets obtained from Caltrans Performance Measurements Systems (PeMS). Caltrans PeMS has placed over 39,000 individual detectors spanning the freeway system across all major metropolitan areas of the State of California and their data are widely used for researcher to develop and evaluate traffic models. The



(a) Generator Architecture (b) Discriminator Architecture

Fig. 4. An Implementation of GANs for Generating Traffic Flow

proposed method is applied to the 5-min traffic flow data of District 5 in the whole year of 2013 except two days due to missing points. There are 153 vehicle detector stations (VDSs) in this district, and we use the data of 147 VDSs among them as there exist null numbers in the data of the other 6 ones.

TABLE I
PERFORMANCES OF INCREMENTALLY TRAINING PREDICTION MODELS

Task[a]		*MAE*	*RMSE*	*MRE*
300-330[b]	real data	15.1790	21.4139	0.3507
	parallel data	12.1010	17.1039	0.2592
270-330[c]	real data	11.5343	16.3350	0.2644
	parallel data	**10.9429**	**15.6154**	**0.2495**

[a] data between 331-th and 363-th days used as test dataset
[b] data between 300-th and 330-th days used as training dataset
[c] data between 270-th and 330-th days used as training dataset

2) Model Configurations: Tensorflow, an open source software library for numerical computation, is used to build the proposed models. As shown in Fig. 4 (a), the generator network comprises one input layer and three fully connected layers (FCL). The activation function for the former two FCL is *tanh*, and the activation function of output FCL is *sigmoid*. The Adam algorithm is used to optimize gradient descent of generator and the learning rate is set to 0.0001. The discriminator network is built by stacking one input layer and three FCL, as illustrated in Fig. 4 (b). The hidden two FCL apply *tanh* activation function, and the output FCL utilizes *sigmoid* as activation function. Adam optimizer is also used to train discriminator, and the learning rate is set to 0.0002.

The implementation of traffic flow prediction model based on LSTM is shown in Fig. 5. The input layer takes in 3D tensor with shape $(batch\_size, history\_steps, 1)$, and the $history\_steps$ in this paper is selected from $\{6, 12, 24, 36, 48, 60\}$. The following five hidden layers are

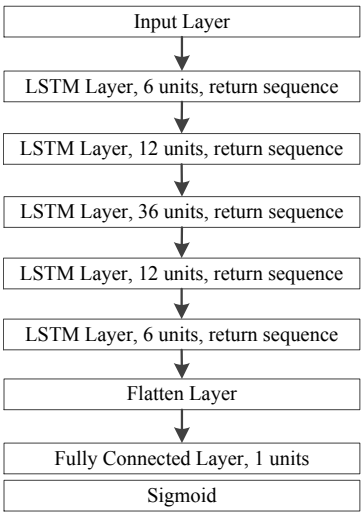


Fig. 5. An Implementation of Traffic Flow Prediction Model based on LSTM recurrent network

built based on LSTM recurrent network. The output activation function of LSTM layer is *tanh* and the recurrent activation function is *hard_sigmoid*. The next layer flattens the returned sequence into a 1D tensor. And the output layer is a FCL with *sigmoid* activation. RMSprop optimizer is used to train prediction model and the learning rate is set to 0.0025.

B. Evaluation Metrics

To evaluate the performances of the proposed traffic flow prediction based on parallel data paradigm, we employ three statistical metrics, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Relative Error (MRE). These indexes are commonly used in evaluating the prediction models and defined as in [10].

C. Performances of Traffic Flow Prediction

To evaluate the improvement achieved by parallel data paradigm, we compare the performances of traffic flow prediction with parallel data and with real data only. We firstly conduct experiments that incrementally train models. The initial GANs model and traffic prediction model are trained with the data between 300-th and 330-th days, and the data between 331-th and 363-th dyas are used as test dataset. Due to the limitation of computational resources, we randomly choose 50 VDSs among all the 147 VDSs. In the experiments, we use 30 minute historical traffic flow data to make up the samples. The experimental results are listed in Table I. In the first task, data between 300-th and 330-th is used to train GANs and traffic prediction model. The parallel data approach improves the *MAE* value from 15.1790 to 12.1010, the *RMSE* value from 21.4139 to 17.1039, and the *MRE* value from 0.3507 to 0.2592. For the second task, data between 270-th and 299-th are used to incrementally refine the models in the first task. The performances of this task are improved at the grounds of the first task. To conclusion,

TABLE II
PERFORMANCES ON USING DIFFERENT HISTORICAL STEPS

Task	*MAE*		*RMSE*		*MRE*	
	parallel data	real data	parallel data	real data	parallel data	real data
30 mins	11.6811	13.1586	16.7114	18.8609	0.2469	0.2976
60 mins	11.1001	11.8192	15.8398	16.743	0.2445	0.2513
120 mins	11.3911	12.3856	16.3966	18.0604	0.2376	0.2589
180 mins	**10.7314**	11.7505	**15.3652**	17.0285	0.2336	0.2376
240 mins	10.9049	11.6952	15.7877	17.0257	0.2213	0.2406
360 mins	10.963	11.6632	15.894	16.9184	**0.2208**	0.2288

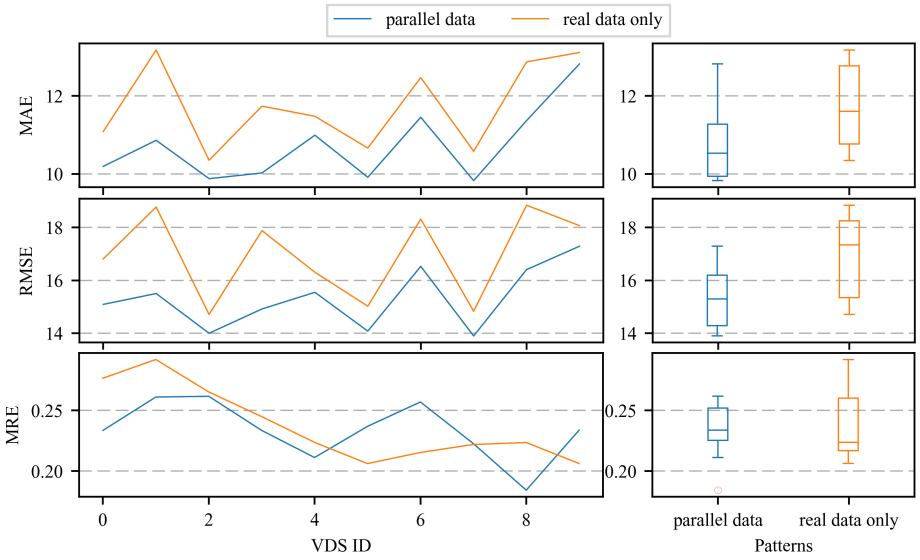


Fig. 6. Performances of Traffic Flow Prediction on Dimension of Vehicle Detector Stations (VDSs)

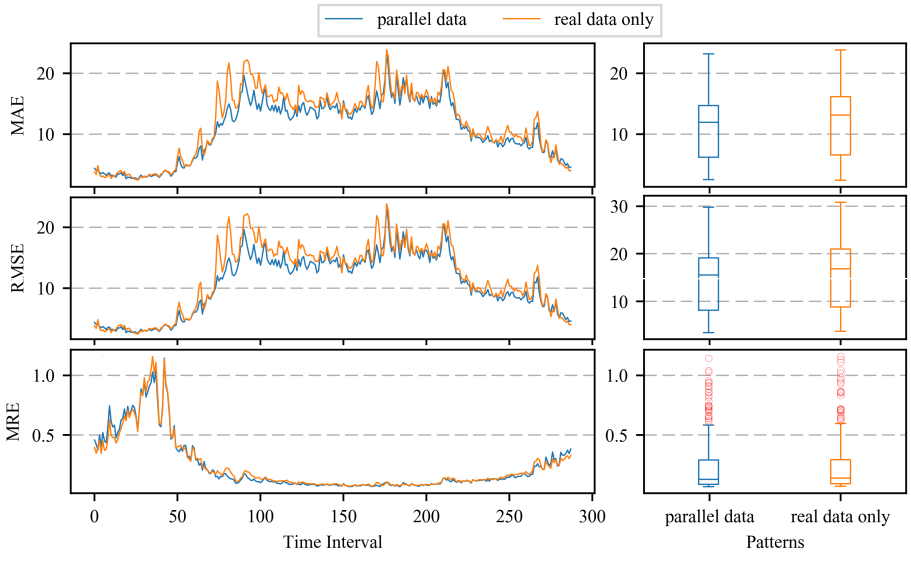


Fig. 7. Performances of Traffic Flow Prediction on Dimension of Time Intervals

parallel data paradigm work well to improve the performance of traffic flow prediction.

To further investigate the effects of historical steps used on prediction performances, we conduct experiments with different historical steps on randomly chosen 10 VDSs. The overall performances is give in Table II. In the experiments, we use data of the first 311 days as training set and data of the last 52 days as test set. For all the tasks, the predictions based on parallel data achieve superior performances. Next we give more details on performances at VDSs and time interval dimension. For different VDSs, the performances are shown in Fig. 6. The parallel data approaches achieve lower *MAE* and *RMSE* for all the VDSs in test, and get higher *MRE* for fifth sixth and ninth VDSs due to the worse prediction at lower traffic flow points, e.g. the traffic flow between 22 o'clock and 24 o'clock. For different time intervals, the performances are shown in Fig. 7. The *MAE* and *RMSE* between 7 o'clock and 19 o'clock is relatively high than that of other periods, while *MRE* is contrary. And the parallel approaches achieve better performances at most intervals.

V. Conclusion

In this paper, we proposed applying parallel data paradigm to improve traffic prediction, especially with small volume data. To reach this goal, real transportation systems, artificial transportation systems for a certain purpose, and traffic prediction models operate and interact in a parallel manner. For traffic prediction, the purpose of artificial transportation system is to generate traffic flow, and we used an improved GANs model. We evaluated the performances of the proposed approach on traffic flow data from Caltrans PeMS. Experimental results showed that the proposed method leads to an improvement in traffic flow data prediction.

In the future, we plan to explore more methods of synthetic traffic data generation like variational autoencoders (VAE), and apply our approach to more tasks in this field, such as travel time prediction.

VI. Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[2] W. Chen, F. Guo, and F.-Y. Wang, "A survey of traffic data visualization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2970–2984, 2015.

[3] F.-Y. Wang, "Driving into the future with ITS," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 94–95, 2006.

[4] L. Li, Y. Lin, N. Zheng, and F. Y. Wang, "Parallel learning: a perspective and a framework," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 389–395, 2017.

[5] X. Liu, X. Wang, W. Zhang, J. Wang, and F. Y. Wang, "Parallel data: from big data to data intelligence," *Pattern Recognition and Artificial Intelligence*, vol. 30, no. 8, pp. 673–681, 2017.

[6] F.-Y. Wang, "A big-data perspective on AI: Newton, Merton, and analytics intelligence," *IEEE Intelligent Systems*, vol. 27, no. 5, pp. 2–4, 2012.

[7] F.-Y. Wang, J. J. Zhang, X. Zheng, X. Wang, Y. Yuan, X. Dai, J. Zhang, and L. Yang, "Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 2, pp. 113–120, 2016.

[8] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2242–2251.

[9] Y. Duan, Y. Lv, Y. L. Liu, and F. Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transportation Research Part C*, vol. 72, pp. 168–181, 2016.

[10] Y. Chen, Y. Lv, Z. Li, and F. Y. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 132–137.

[11] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.

[12] N. L. Nihan and K. O. Holmesland, "Use of the box and Jenkins time series technique in traffic forecasting," *Transportation*, vol. 9, no. 2, pp. 125–143, 1980.

[13] M. S. Ahmed and A. R. Cook, *Analysis of Freeway Traffic Time-series Data by Using Box-Jenkins Techniques*, 1979.

[14] X. J. Chen, J. H. Ma, W. Guan, and W. Y. Tu, "Traffic volume prediction using improved Kalman filter," in *Advanced Manufacturing and Information Engineering, Intelligent Instrumentation and Industry Development*, ser. Applied Mechanics and Materials, vol. 602. Trans Tech Publications, 2014, pp. 3881–3885.

[15] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[16] J. Arroyo and C. Mat, "Forecasting histogram time series with k-nearest neighbours methods," *International Journal of Forecasting*, vol. 25, no. 1, pp. 192–207, 2009.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[18] Y. Duan, Y. Lv, and F.-Y. Wang, "Travel time prediction with lstm neural network," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1053–1058.

[19] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, 2017.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[21] G. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *CoRR*, vol. abs/1701.06264, 2017.

[22] Y. Lv, Y. Chen, L. Li, and F. Wang, "Generative adversarial networks for parallel transportation systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 3, pp. 4–10, 2018.

[23] F.-Y. Wang and S. Tang, "Artificial societies for integrated and sustainable development of metropolitan systems," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 82–87, 2004.

[24] F.-Y. Wang, "Toward a paradigm shift in social computing: the ACP approach," *IEEE Intelligent Systems*, vol. 22, no. 5, pp. 65–67, 2007.

[25] K. Wang, C. Gou, N. Zheng, J. Rehg, and F.-Y. Wang, "Parallel vision for perception and understanding of complex scenes: Methods, framework, and perspectives," in *Artificial Intelligence Review*, vol. 48, 2017, pp. 299–329.

[26] C. Gou, Y. Wu, K. Wang, F.-Y. Wang, and Q. Ji, "Learning-by-synthesis for accurate eye detection," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 3362–3367.

[27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[28] Y. Chen, Y. Lv, X. Wang, L. Li, and F.-Y. Wang, "Detecting traffic information from social media texts with deep learning approaches," *IEEE Transactions on Intelligent Transportation Systems*.

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