Performance Evaluation of the Deep Learning Approach for Traffic Flow Prediction at Different Times

Yanjie Duan*‡, Yisheng Lv*† and Fei-Yue Wang*‡
*The State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
†Beijing Engineering Research Center of Intelligent Systems and Technology
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
‡Qingdao Academy of Intelligent Industries
Qingdao, Shandong, 266109, China

***Abstract*—Traffic flow prediction is very important in the deployment of intelligent transportation system. Based on our previous research on deep learning approach for traffic data prediction, we further evaluates the performance of the SAE model for traffic flow prediction at daytime and nighttime. Through 250 experimental tasks training a SAE model and evaluating its performance at daytime and nighttime with 3 different criteria, we obtain the best combination of hyper parameters for each criterion at different times on weekday and non-weekday, respectively. Experimental results show that the MAE and RMSE at daytime are larger than that at nighttime, while the MRE at daytime are smaller than that at nighttime. For different criteria, the hyper parameters of the SAE model should vary accordingly. The results in this paper indicate that in real applications, traffic flow prediction using the deep learning approach can be a combination of multiple SAE models with different parameters suitable for different periods, which is of significance in future research.**

I. Introduction

Traffic flow information is very important in the deployment of intelligent transportation system (ITS) [1]. Accurate and timely prediction of traffic flow can provide support for road conditions analysis, dynamic route guidance and traffic signal control [2]. Thus traffic flow prediction is a critical subject in ITS research and applications.

There have been many researchers studying traffic flow prediction using various methods. These methods are mainly divided into two categories: data-driven methods and model-driven methods. Data-driven methods rely on traffic flow data collected from traffic sensors, e.g. cameras, inductive loops etc. Typically, historical traffic flow data are utilized to build a prediction model. Then real-time traffic flow data are fed into the model and the predicted traffic flow in future time is obtained from the output of the model. Models or algorithms in these methods include the autoregressive integrated moving-average (ARIMA) model [3] and its variations [4], [5], Kalman filtering model [6], k-nearest neighbor (kNN) algorithm [7] , multivariate regression model [8], neural network model [9], [10] and deep learning model [11] and so on. Model-driven methods are based on the principle of dynamic traffic assignment (DTA). A proper traffic model and the travel demand data of the road network are essential to conduct the simulation process of DTA. Through the simulation, the traffic flow of one specific site can be detected from the virtual road network. There are many transportation softwares such as DynaSmart [12], DynaMIT [13], Vissim [14], Paramics [15], TransWorld [1], which can build virtual road networks for traffic simulations. However, they all need the travel demand data named origin-destination (OD) matrix or the demographic data which must be determined by engineers in advance and is hard to acquire.

With the growth of traffic data quantity and the improvement of computing capability, data-driven methods [16] have attracted the attention of many researchers and achieved inspiring results. Among them, the deep learning based methods lead the trend of big data processing. Exploring deep learning models in the application the traffic flow prediction is of great significance. We have done the work [17] in which global road flow are predicted with the deep learning model named stacked autoencoder (SAE). Huang etc. [18] used the deep learning model named deep belief network to predict traffic flow. The existing research demonstrated that deep learning is promising in traffic flow prediction, and the performance evaluations focused on traffic flow prediction at all times in a day. However, we are more concerned about the performance of the deep learning model at different times in a day. Therefore in this paper, we further evaluated the performance of the deep learning model for traffic flow prediction at different times. Different from our previous research, we discuss the prediction of traffic flow collected from original vehicle detector stations (VDSs) rather than the processed global road flow.

The rest of this paper is organized as the following: Section II describes the SAE model for traffic flow prediction. Section III presents the experiments conducted on the model using real traffic flow with different model set, evaluates the performances at different times. Section IV concludes this paper.

978-1-5090-2927-3/16/$31.00 ©2016 IEEE

II. METHODOLOGY

Our deep learning approach for traffic flow prediction uses the SAE model consisting of a regression layer on the top and multiple basic network blocks, which are obtained from multiple autoencoders (AEs) [19]. An AE extracts features from its input and SAE can obtain an abstract representation of the input through gradual feature extraction. Then the regression layer fits the expected output and the feature representations. Here, the traffic flow prediction problem is defined to predict the traffic flow in the SAE model output $\mathbf{y} = \widetilde{\mathbf{x}}_t$ using history traffic flow as the SAE model input $\mathbf{x} = [\mathbf{x}_{t-K}, \cdots, \mathbf{x}_{t-1}]$. $\mathbf{x}_t$ is a vector consisting of traffic flow data collected from all the VDSs in a district at time t, thus its dimension equals N, which is the number of the VDSs. The following paragraphs will introduce the structure and training algorithm of an AE, then illustrate the construction of the SAE model for traffic flow prediction.

A. AE

An AE shown in Fig. 1 includes two parts: encoder and decoder. Encoder f_θ maps an input vector $\mathbf{x}$ into its hidden representation $\mathbf{h}$, thus $\mathbf{h} = f_\theta(\mathbf{x})$. f_θ is typically a nonlinear transformation function in the following form:

$$f_\theta(\mathbf{x}) = s(\mathbf{x}\mathbf{W}^T + \mathbf{b}), \tag{1}$$

where θ represents all its parameters containing $\mathbf{W}$ and $\mathbf{b}$. $\mathbf{W}$ is the weight matrix between the hidden layer and the input layer. $\mathbf{b}$ is the bias vector of the hidden layer. Decoder $g_{\theta'}$ maps the hidden representation $\mathbf{h}$ back to a reconstructed vector $\mathbf{y}$ of the input vector $\mathbf{x}$, thus $\mathbf{y} = g_{\theta'}(\mathbf{h})$. $g_{\theta'}$ is as well typically a nonlinear transformation in the following form:

$$g_{\theta'}(\mathbf{h}) = s(\mathbf{h}\mathbf{W}^{'T} + \mathbf{b}^{'}), \tag{2}$$

where $\theta^{'}$ represents all its parameters containing $\mathbf{W}^{'}$ and $\mathbf{b}^{'}$. $\mathbf{W}^{'}$ is the weight matrix between the output layer and the hidden layer. $\mathbf{b}^{'}$ is the bias vector of the output layer. In equation (1) and (2), s is called activity function.

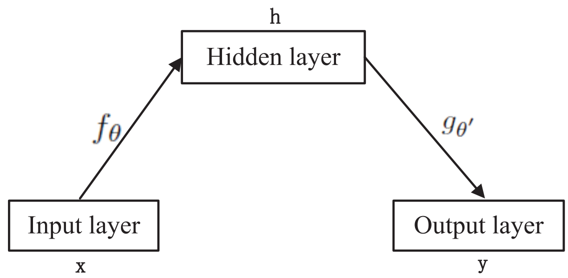


Fig. 1. The structure of an AE

Training an autoencoder is the process of minimizing the reconstruction error by solving the following optimization problem:

$$\theta, \theta^{'} = \arg\min_{\theta,\theta'} L(X, Y), \tag{3}$$

where X is a set of input vector $\mathbf{x}$ and Y is the corresponding set of reconstructed vector $\mathbf{y}$. This process is summarized in Algorithm 1. The hidden representation of a trained AE is seen as the feature vector of the input vector. The feature vector will be treated as the input of the higher AE and used to extract higher level representation in a SAE.

Algorithm 1 Training an autoencoder

Given data set $X = \{\mathbf{x}\}$, $\mathbf{x} \in \mathbb{R}^{KN}$;
Set the number of hidden units n_h, the iterations T;
Initialize the weights and biases randomly: $\mathbf{W}(n_h \times KN)$, $\mathbf{W}^{'}(KN \times n_h)$, $\mathbf{b}(1 \times n_h)$, $\mathbf{b}^{'}(1 \times KN)$;
for i = 1 to T
  Perform forward propagation to compute Y
  Compute output error: $X - Y$
  Perform backward propagation to compute $\Delta\theta$, $\Delta\theta^{'}$
  Update θ, $\theta^{'}$: $\theta = \theta + \Delta\theta$, $\theta^{'} = \theta^{'} + \Delta\theta^{'}$
end for

B. The SAE Model

The SAE model shown in Fig. 2 is structured with multiple AEs and a regression layer on the top. Each hidden layer is obtained from an AE. The SAE model is constructed from the bottom layer up to the top layer. Each hidden layer is treated as the input of the AE top on it. For the purpose of traffic flow prediction, the top layer is a regression layer for fitting the top hidden layer and the expected output. Specifically, the model feed forward calculates the predicted output corresponding to a input vector $\mathbf{x}$ through the following equations:

$$\mathbf{h}_l = \begin{cases} s(\mathbf{x}\mathbf{W}_1^T + \mathbf{b}_1) & l = 1 \\ s(\mathbf{h}_{l-1}\mathbf{W}_l^T + \mathbf{b}_l) & L \geqslant l > 1 \end{cases}, \tag{4}$$

$$\mathbf{y} = s(\mathbf{h}_{L+1}\mathbf{W}_{L+1}^T + \mathbf{b}_{L+1}), \tag{5}$$

where $\mathbf{h}_l$ is the hidden representation of the lth hidden layer, $\mathbf{W}_l$ is the weight matrix between the lth hidden layer and the layer below it, $\mathbf{b}_l$ is the bias vector of the lth hidden layer, $\mathbf{W}_{L+1}$ is the weight matrix between the output layer and the Lth hidden layer, $\mathbf{b}_{L+1}$ is the bias vector of the output layer.

Training the SAE model is the process of minimizing the prediction error by solving the following optimization problem:

$$\Theta = \arg\min_{\Theta} L(X, Y), \tag{6}$$

where Θ represents all its parameters containing $\mathbf{W}_l$ and $\mathbf{b}_l(l = 1, 2, \cdots, L+1)$. Different from training an AE, this process includes two training steps: pretraining and fine-tuning. Pretraining trains the parameters of each AE while fine-tuning adjusts all the parameters in the SAE model. The process is summarized in Algorithm 2. After training, the SAE model can predict the future traffic flow $\mathbf{y} = \widetilde{\mathbf{x}}_t$ from history traffic flow input $\mathbf{x} = [\mathbf{x}_{t-K}, \cdots, \mathbf{x}_{t-1}]$.

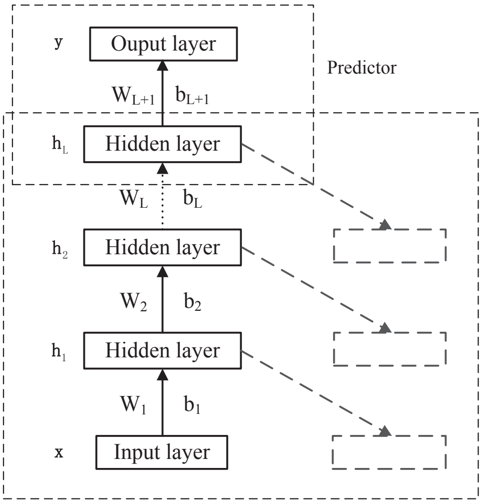


Fig. 2. The structure of the SAE model

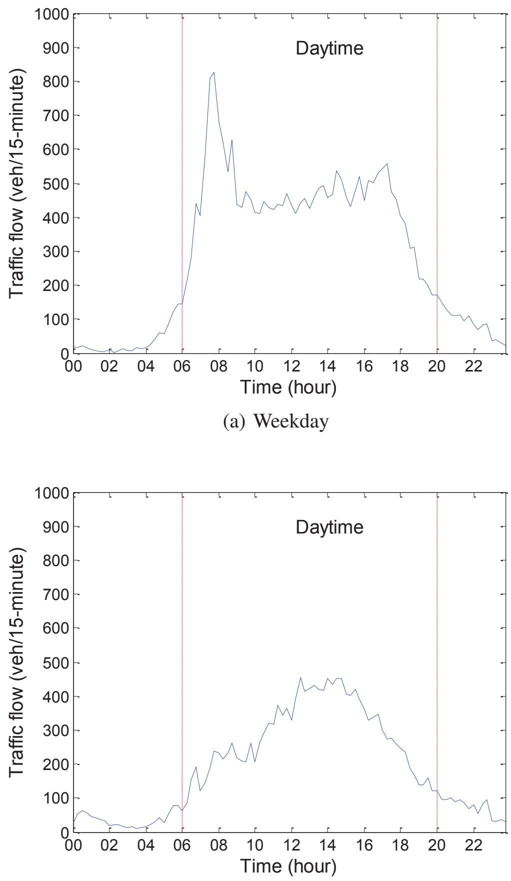
Algorithm 2 Training the SAE model

Given data set $X = \{\mathbf{x}\}$, $Y = \{\mathbf{y}\}$, $\mathbf{x} \in \mathbb{R}^{KN}$, $\mathbf{y} \in \mathbb{R}^{N}$;
Set the number of hidden layers L, the number of hidden units $n_l, l = 1, 2, \cdots, L$, the pretraining iterations T, the fine-tuning iterations $T^{'}$;
Step 1: Pretraining
  X is fed to the input layer;
  Train the first AE using Algorithm 1;
  for l = 2 to L
    H_{l-1} is fed to the lth AE
    Train the lth AE using Algorithm 1
  end for
Step 2: Fine-tuning
  Initialize $\mathbf{W}_{L+1}(N \times n_L)$, $\mathbf{b}(1 \times N)$ randomly;
  for i = 1 to $T^{'}$
    Perform forward propagation to compute Y
    Compute output error: $X - Y$
    Perform backward propagation to compute $\Delta\Theta$
    Update Θ: $\Theta = \Theta + \Delta\Theta$
  end for

III. Experiments and Performance Evaluation

A. Data Description

The traffic flow data for this study are from the Caltrans (California Department of Transportation) PeMS (Performance Measurement System) [20]. We exploit the 5-minute incremental traffic flow data collected from the VDSs in District 5: Central Coast during the first three month in Year 2013. There are 153 VDSs and 90 days in the chosen dataset. However, we only use data collected from 147 VDSs on 89 days. Because 6 VDSs involve null numbers due to some unknown reason and the data collected on March 10th in Year 2013 consist of less than one day's time stamps, and we remove them. Then the 5-minute incremental traffic flow data are aggregated into 15-minute incremental traffic flow data, which will be used in the following experiments. Fig. 3(a) shows the 15-minute incremental traffic flow collected on a weekday while Fig. 3(b) shows that on a non-weekday from VDS 500010092 in the experimental district. Obviously, the traffic flow at daytime (6a.m.-8p.m.) is higher than that at nighttime (8p.m.-6a.m.). We conduct experiments to see the performance of the deep learning approach for traffic flow prediction at daytime and nighttime on weekday and non-weekday, respectively. The ratio of training set and test set is 4:1.



(a) Weekday

(b) Non-weekday

Fig. 3. Typical traffic flow collected from a VDS

B. Evaluation Criteria

In order to evaluate the performances of the deep learning model for traffic flow prediction at different times, we adopt three criteria to measure the error of the predicted data. They are mean absolute error (MAE),

$$MAE = \frac{\sum_{t=1}^{M}\sum_{i=1}^{N}|x_t^i - y_t^i|}{MN},$$

root mean square error (RMSE),

$$RMSE = \sqrt{\frac{\sum_{t=1}^{M}\sum_{i=1}^{N}(x_t^i - y_t^i)^2}{MN}},$$

and mean relative error (MRE),

$$MRE = \frac{\sum_{t=1}^{M} \sum_{i=1}^{N} \frac{|x_t^i - y_t^i|}{x_t}}{MN},$$

where x_t^i is the observed traffic flow collected from the ith VDS at time t while y_t^i is the predicted one, M is the total number of test samples and N is the dimension of the output. according to our dataset, $N = 147$.

C. Experiments Set

From the data description part, we find different data patterns between weekday traffic flow and non-weekday traffic flow. Therefore the SAE model is applied to the prediction of weekday traffic flow and non-weekday traffic flow separately. Besides, the hyper parameters of the SAE model: the activation function, the number of hidden layers of the network, the number of hidden units in each hidden layer, the times of iteration and the time delay steps must be determined separately from the training process of parameters Θ in equation 6. With the experience in our previous work [17], we set the range of the hyper parameters and choose them with the listed values in Table I. Then we have 125 combinations of the hyper parameters i.e. 125 choices for the SAE model. Considering weekday and non-weekday, we conduct 250 experimental tasks training a SAE model and evaluating its performance at daytime and nighttime. To train the SAE model, we set the cost function

$$L(X, Y) = \frac{\sum_{t=1}^{M} \sum_{i=1}^{N} (x_t^i - y_t^i)^2}{2MN}.$$

The training process is conducted following Algorithm 2.

TABLE I
HYPER PARAMETERS SET IN THE SAE MODEL

Hyper parameters	Range	Experiment values
Activation function	sigmoid	sigmoid
Number of hidden layers	1∼5	1,2,3,4,5
Number of hidden units	100∼500	100,200,300,400,500
Times of pretraining iteration	50∼100	100
Times of fine-tuning iteration	20∼100	100
Time delay steps	1∼5	1,2,3,4,5

D. Results and Discussions

Through the experiments above, the performance of the SAE model for traffic flow prediction on the test dataset is obtained. Table II and Table III present the best combination of the hyper parameters in the our experiment range for each criterion on weekdays and non-weekdays, respectively. Obviously, the best combination of hyper parameters is different for each criterion at different times. The MAE and RMSE at daytime are larger than that at nighttime, while the MRE at daytime is smaller than that at nighttime. In terms of the time delay steps, most best performances need more than only one time delay step, which shows the importance of long term memory in the model. For the number of hidden layers, weekday traffic flow prediction tends to need more hidden layers than non-weekday to achieve better performance in most cases. Considering the different patterns shown in Fig 3, we infer that the more complex the data pattern is, the deeper the SAE model needs to be. All the number of hidden units are larger than the dimension of the output. In fact, the quantity of the SAE model parameters with some experiment set in Table I is more than the number of samples in our dataset. However, the experimental results show that all the SAE model can converge in the limited times of iterations and avoid over fitting. That indicates the advantages of deep learning model in training a model with limited data. The traffic flow at daytime are much larger than that at nighttime, thus we concern more about the prediction accuracy of the the predicted traffic flow at daytime. Here we present the predicted traffic flow using the SAE model with the best hyper parameters for daytime prediction in terms of MAE. Fig. 4(a) shows the predicted flow on weekday with the hyper parameters $K = 4, L = 2, n_h = 400$ and Fig. 4(b) shows that on non-weekday with the hyper parameters $K = 5, L = 1, n_h = 500$. The predicted traffic flow data follow the trend of the observed ones.

TABLE II
WEEKDAY EXPERIMENTAL RESULTS:
BEST HYPER PARAMETERS FOR EACH CRITERION

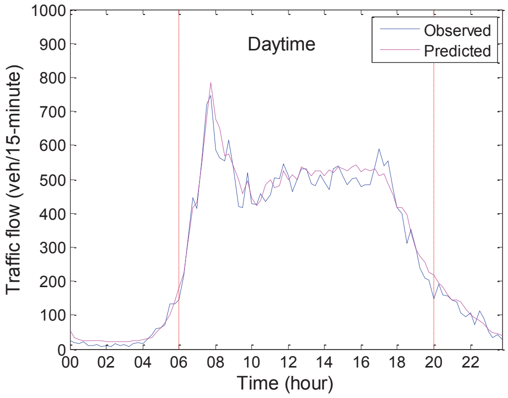
Time type	Criterion	Error value	K	L	n_h
All	MAE	38.28	4	2	400
Daytime	MAE	49.26	4	2	400
Nighttime	MAE	21.73	5	3	500
All	RMSE	59.07	1	1	500
Daytime	RMSE	71.52	1	1	500
Nighttime	RMSE	33.53	4	3	300
All	MRE	25.78%	3	2	400
Daytime	MRE	16.59%	1	1	400
Nighttime	MRE	36.05%	5	5	400

TABLE III
NON-WEEKDAY EXPERIMENTAL RESULTS:
BEST HYPER PARAMETERS FOR EACH CRITERION

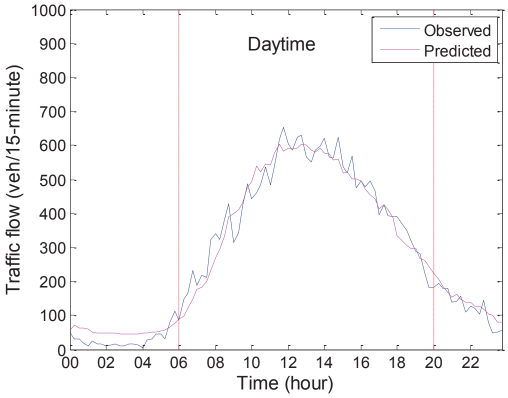
Time type	Criterion	Error value	K	L	n_h
All	MAE	35.56	5	1	500
Daytime	MAE	43.11	5	1	500
Nighttime	MAE	24.38	3	1	500
All	RMSE	52.48	2	1	400
Daytime	RMSE	61.30	2	1	400
Nighttime	RMSE	35.70	3	1	500
All	MRE	23.69%	1	4	200
Daytime	MRE	13.14%	5	1	500
Nighttime	MRE	37.39%	1	4	200

IV. CONCLUSION

This paper evaluates the performance of the SAE model for traffic flow prediction at daytime and nighttime. Through 250 experimental tasks training a SAE model and evaluating its



(a) Weekday



(b) Non-weekday

Fig. 4. Predicted traffic flow for a VDS

performance at daytime and nighttime for 3 different criteria, we obtain the best combination of hyper parameters for each criterion at different times on weekday and non-weekday, respectively. Experimental results show that the MAE and RMSE at daytime are larger than that at nighttime, while the MRE at daytime is smaller than that at nighttime. For different criteria, the hyper parameters of the SAE model should vary accordingly. The results in this paper indicate that in real applications, traffic flow prediction using the deep learning approach can be a combination of multiple SAE models with different parameters suitable for different periods, which is of significance in future research.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61233001, 61203166, 71232006 and 61533019, and the Early Career Development Award of SKLMCCS.

REFERENCES

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

[2] Z. Nan, W. Fei-Yue, Z. Fenghua, Z. Dongbin, and T. Shuming, “Dynacas: computational experiments and decision support for its,” *IEEE Intelligent Systems*, vol. 23, no. 6, pp. 19–23, 2008.

[3] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. 1979.

[4] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[5] T. Mai, B. Ghosh, and S. Wilson, “Multivariate short-term traffic flow forecasting using bayesian vector autoregressive moving average model,” in *Transportation Research Board 91st Annual Meeting*.

[6] F. Yang, Z. Yin, H. Liu, and B. Ran, “Online recursive algorithm for short-term traffic prediction,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1879, pp. 1–8, 2004.

[7] H. Chang, Y. Lee, B.-J. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: systemoriented approach based on past experiences,” *Intelligent Transport Systems, IET*, vol. 6, no. 3, pp. 292–305, 2012.

[8] P. Dell’Acqua, F. Bellotti, R. Berta, and A. De Gloria, “Time-aware multivariate nearest neighbor regression methods for traffic flow prediction,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 3393–3402, 2015.

[9] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, “Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenbergmarquardt algorithm,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 644–654, 2012.

[10] Y.-J. Kim and J.-s. Hong, “Urban traffic flow prediction system using a multifactor pattern recognition model,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 5, pp. 2744–2755, 2015.

[11] N. Polson and V. Sokolov, “Deep learning predictors for traffic flows,” *arXiv preprint arXiv:1604.04527*, 2016.

[12] H. Mahmassani, T. Hu, and R. Jayakrishnan, “Dynamic traffic assignment and simulation for advanced network informatics (dynasmart),” in *Proceedings of the 2nd international CAPRI seminar on Urban Traffic Networks, Capri, Italy*.

[13] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, “Dynamit: a simulation-based system for traffic prediction,” in *DACCORS short term forecasting workshop, The Netherlands*, Citeseer.

[14] M. Fellendorf, “Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority,” in *64th Institute of Transportation Engineers Annual Meeting*, pp. 1–9, Springer.

[15] G. D. Cameron and G. I. Duncan, “Paramicsparallel microscopic simulation of road traffic,” *The Journal of Supercomputing*, vol. 10, no. 1, pp. 25–53, 1996.

[16] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *Ieee Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

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

[18] H. Wenhao, S. Guojie, H. Haikun, and X. Kunqing, “Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[19] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[20] “Performance measurement system, pems, university of california, berkeley. available at http://pems.dot.ca.gov/, accessed june 2014..”