DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction

Xingyuan Dai, Rui Fu, Yilun Lin, Fei-Yue Wang, *Fellow, IEEE*, and Li Li, *Fellow, IEEE*

***Abstract*—In this paper, we consider the temporal pattern in traffic flow time series, and implement a deep learning model for traffic flow prediction. Detrending based methods decompose original flow series into trend and residual series, in which trend describes the fixed temporal pattern in traffic flow and residual series is used for prediction. Inspired by the detrending method, we propose DeepTrend, a deep hierarchical neural network used for traffic flow prediction which considers and extracts the time-variant trend. DeepTrend has two stacked layers: extraction layer and prediction layer. Extraction layer, a fully connected layer, is used to extract the time-variant trend in traffic flow by feeding the original flow series concatenated with corresponding simple average trend series. Prediction layer, an LSTM layer, is used to make flow prediction by feeding the obtained trend from the output of extraction layer and calculated residual series. To make the model more effective, DeepTrend needs first pre-trained layer-by-layer and then fine-tuned in the entire network. Experiments show that DeepTrend can noticeably boost the prediction performance compared with some traditional prediction models and LSTM with detrending based methods.**

I. Introduction

Traffic flow prediction is one of the major tasks of intelligent transportation systems (ITSs) that should be resolved [1], [2]. It is strongly needed for individuals, companies, governments and so on to make decisions in time according to different conditions of traffic flow. However, accurate and real-time traffic prediction remains challenging and unsolved for many decades due to its stochastic and nonlinear feature. Traditional methods mainly use linear models like autoregressive integrated moving average (ARIMA) [3]–[7] and multi-variable linear regression (MVLR) [8], [9], and some machine learning models like support vector regression (SVR) [10] to predict incoming traffic flow but cannot consider the entire features in traffic flow and perform not very well.

In recent years, with the development of deep learning [11]–[15], some deep learning methods for traffic flow prediction are put forward like stacked autoencoders (SAEs) [16], long short-term memory network (LSTM) [17], deep belief network (DBN) [18], etc., and have good performance. On one hand, these models generally have complex network structures which can fit nonlinear parts in traffic flow series. On the other hand, some deep learning models like LSTM and gated recurrent unit (GRU) [19] are designed especially for the time series which are adept in dealing with traffic flow.

In this paper, we explore whether deep networks like LSTM can learn the temporal patterns existed in flow time series, which is of great importance for traffic prediction. However, the experiments show that LSTM has similar prediction performance with some traditional machine learning models. To make LSTM more effective in flow prediction, we introduce detrending based methods, which are frequently used in traffic flow prediction nowadays [8], [20]–[22]. It is based on the hypothesis that there exists a certain temporal pattern trend in traffic flow time series and can be separated from the remaining fluctuations. So researchers often assume there exists invariant periodic trend in traffic flow time series. Some methods were used to retrieve intra-day or seasonal trend via simple-average, principal component analysis (PCA) or wavelet methods. Inspired by the idea of detrending, we propose a well-designed deep network architecture named DeepTrend. DeepTrend has two kinds of hidden layers: extraction layer and prediction layer. The extraction layer is used to learn to extract the time-variant trend, and the prediction layer is used to predict the incoming flow by feeding the extracted trend and calculated residual series. Experiments show that DeepTrend outperforms other baselines based on original flow data or detrending methods.

The rest of this paper is organized as follows. Section II reviews the studies on short-term traffic flow prediction. Section III proposes the DeepTrend architecture for traffic flow prediction and the detrending based method. Section IV discusses the experimental design and performance of the proposed architecture, and comparison with several selected models. Finally, Section V concludes the paper.

II. Literature Review

In general, traffic flow prediction approaches can be divided into two major categories: parametric approach and nonparametric approach.

The main parametric approach includes ARIMA [3] model, MVLR [8], [9]. The model architecture of these approaches is predetermined based on the certain theoretical assumptions and the model parameters should be calculated by empirical data. ARIMA model is based on the assumption that the traffic condition is in a stationary process. It was first used for

*This work was partially supported by National Natural Science Foundation of China under Grants 61533019, 71232006, and 91520301.

Xingyuan Dai, Yilun Lin and Fei-Yue Wang are with The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and with University of Chinese Academy of Sciences, Beijing, 100049, China. They are also affiliated with Qingdao Academy of Intelligent Industries, Qingdao, Shandong, 266109, China.

Rui Fu is with Department of Automation, Tsinghua University, Beijing, 100084, China.

Li Li is with Department of Automation, Tsinghua University, Beijing, 100084, China. (e-mail: li-li@tsinghua.edu.cn). Corresponding author of this paper.

short-term traffic flow prediction in the 1970s [3], and then ARIMA (0, 1, 1) [4] was found more statistically significant for flow prediction. Moreover, some improved ARIMA models like subset ARIMA [5], space-time ARIMA [6] and seasonal ARIMA (SARIMA) [7] were also proposed to forecast traffic flow. The parametric approach has simple and explicit architecture and takes a little time to obtain the results.

However, due to the stochastic and nonlinear feature in traffic flow, the parametric approach with linearity cannot present a high performance for traffic flow prediction. Therefore, researchers have paid much attention to the nonparametric approach such as k-NN [23], SVR [10], online support vector regression (OL-SVR) [24], random forests regression (RF) [25], gradient boosting regression [26]. A variety of artificial neural network (ANN) models were proposed to predict traffic flow and perform well [27]–[29]. Recently, with the development of deep learning, many deep learning models were applied to traffic flow prediction. SAE [16], DBN [18], LSTM [17] and GRU [19] model were proposed in traffic flow forecasting and got superior performance. However, these recent studies do not further explore to extract the intra-patterns of flow series in models, which needs to be concerned for better traffic flow prediction.

III. Methodology

A. Recurrent Neural Network (RNN)

The RNN [30] is a generation of the feedforward neural networks which is adept in dealing with sequences. The structure of RNN is shown in Fig. 1. Given a general input sequence $(x_1, x_2, \cdots, x_k)$ where $x_i \in \mathbb{R}^d$, a hidden state is obtained at each time step, resulting in a hidden sequence $(h_1, h_2, \cdots, h_k)$. The hidden state at time step t is calculated by the function

$$h_t = f\left(x_t, h_{t-1}\right) \tag{1}$$

in which x_t is the current input and h_{t-1} is the previous hidden state. Then the optional output at each time step is calculated by $y_t = g(h_t)$. The output of RNN can be a sequence as $(y_1, y_2, \cdots, y_k)$ or a single value y_k which is dependent on the objective of the problems.

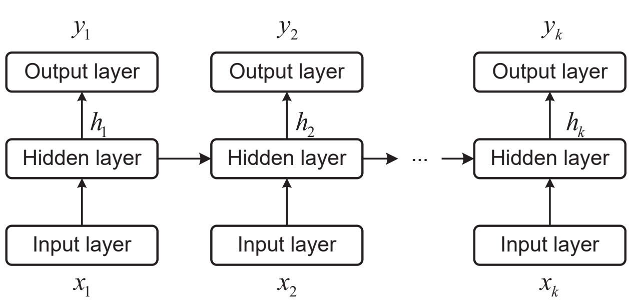

Fig. 1: The structure of RNN.

The simple RNN calculates the output at each time step, making the network very deep. It is hard for simple RNN to train and capture the dependence of the input sequence. Thus, the design of hidden layer structure is essential.

B. Long Short-Term Memory network (LSTM)

LSTM [31], [32] is a special kind of RNN, designed to learn long-term dependencies. It has a complex structure named LSTM unit in its hidden layer which contains three gates namely input gate, forget gate and output gate to protect and control the unit state. The LSTM unit is shown in Fig 2, in which IN represents the input data and the previous unit's output.

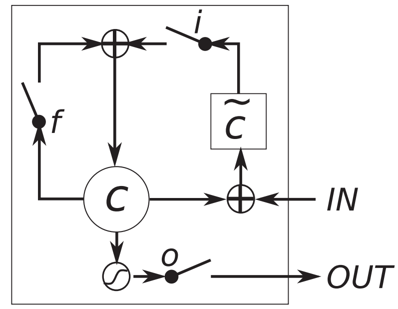

Fig. 2: Long Short-Term Memory [32]

Denote that the input is x_t and the hidden units output is h_t at time step t and their previous output is h_{t-1}. For the j-th LSTM unit, the input gate i_t^j, forget gate f_t^j and output gate o_t^j can be calculated using the following equations:

$$i_t^j = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right)^j \tag{2}$$

$$f_t^j = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right)^j \tag{3}$$

$$o_t^j = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right)^j \tag{4}$$

where σ is a logistic sigmoid function, W terms are weight matrices, and b terms are bias vectors.

Unlike traditional recurrent unit, each j-th LSTM unit maintains a memory c_t^j at time t. The memory cell c_t^j is updated by

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j \tag{5}$$

where new memory content is

$$\tilde{c}_t^j = \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)^j \tag{6}$$

The LSTM unit output is computed by

$$h_t^j = o_t^j \tanh\left(c_t^j\right) \tag{7}$$

C. LSTM Network for Traffic Flow Prediction

We apply one-layer LSTM to traffic flow prediction. The main architecture is shown in Fig 3. At time t, the input of the network is the observed historical traffic data which we use the previous N steps data as $\mathbf{x} = (x_{t-N+1}, x_{t-N+2}, \cdots, x_t)$ and the output $\hat{x}_{t+1}$ is the predicted traffic flow in next time step. We can get the hidden unit output h_t using the above equations, and the output of the network can be calculated as

$$\hat{x}_{t+1} = W_{ho}h_t + b \tag{8}$$

where W_{ho} is the weight matrix between the hidden layer and output layer and b is bias term. Then, we use Back Propagation Through Time (BPTT) [33] algorithm to train the model.

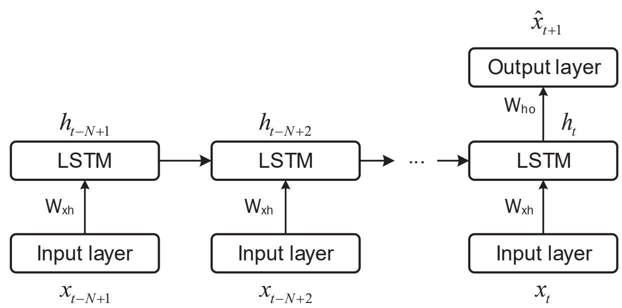

Fig. 3: The structure of one-layer LSTM network for traffic prediction.

D. Detrending Based Prediction

Detrending [8], [20]–[22] based methods have been widely utilized in predicting traffic flow series. The goal of detrending is to remove the periodic trend that may influence traffic prediction and using the residual time series to make predictions. This scheme enables the networks to only pay attention to the local limited change in series without considering the change caused by periodic trend .

In this paper, we make prediction without distinguishing between weekday and weekend. Therefore, the daily periodic trend that the previous works [8], [20]–[22] use can not be considered because there are huge different patterns of traffic flow in weekday and weekend, and we use weekly periodic trend.

The easiest way to calculate the trend is to use the average of periodic traffic flow time series collected in the same station, which is called simple average trend.

Let y^i_{j-k} denote the k-th sample point data at station i in j-th week. The traffic time series in N continuous weeks can be written as a series of vectors

$$\begin{aligned} Y^i_1 &= \left[y^i_{1-1}, y^i_{1-2}, \cdots, y^i_{1-n}\right], \\ Y^i_2 &= \left[y^i_{2-1}, y^i_{2-2}, \cdots, y^i_{2-n}\right], \\ &\cdots \\ Y^i_N &= \left[y^i_{N-1}, y^i_{N-2}, \cdots, y^i_{N-n}\right] \end{aligned} \tag{9}$$

where n is the number of sample data points per week. In the paper, we consider that the sample time interval is 5 minutes, and get 288 sample points in a day, so we have $n = 2016$.

The simple average trend over past D weeks can be calculated as

$$Y^i_{Average} = \left[\frac{1}{D}\sum_{j=N-D+1}^{N} y^i_{j-1}, \cdots, \frac{1}{D}\sum_{j=N-D+1}^{N} y^i_{j-n}\right] \tag{10}$$

where $D = N$ indicates the average for all sample weeks.

Then, we can obtain the residual time series $R^i_j = \left[R^i_{j-1}, R^i_{j-2}, \cdots, R^i_{j-n}\right]$ by subtracting the simple average trend from the original time series as

$$R^i_j = Y^i_j - Y^i_{Average} \tag{11}$$

The residual time series instead of original ones are finally fed into the prediction models in detrending based methods and the predicted residual value can be obtained. The predicted flow are then calculated by adding the predicted residual value and the trend value together.

E. DeepTrend

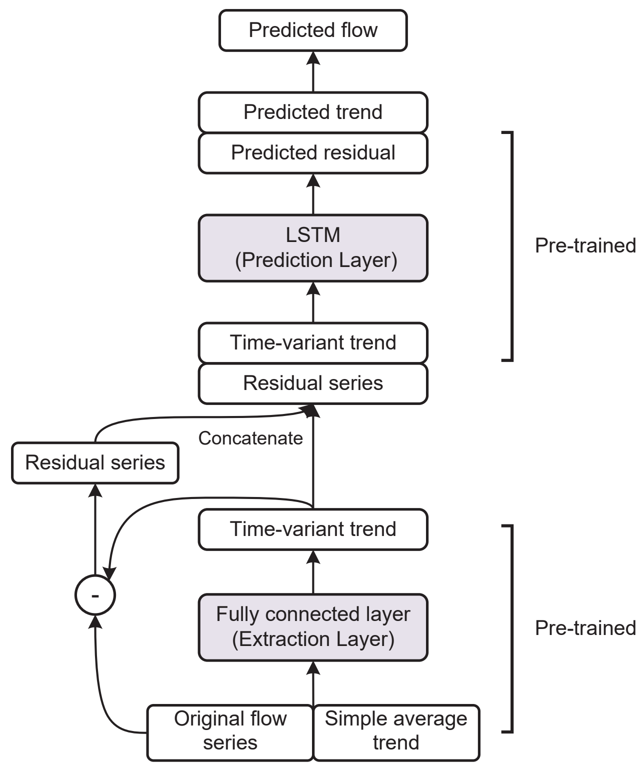

Fig. 4: The structure of DeepTrend.

Simple average detrending is useful to eliminate the fixed temporal pattern existed in traffic flow and make better prediction using residual series. However, it is based on the hypothesis that there exists invariant periodic trend in flow series. It is common that the traffic flows may have big difference even they are at the same time in different weeks. In this case, their temporal patterns may be variant, but their simple average trends are still same. This will make that the residual series hard to forecast.

To solve the problem and make it better for the predictor to learn the temporal pattern existed in flow time series, we propose DeepTrend, which is used to better capture the time-variant trend and lift the prediction performance.

As shown in Fig. 4, DeepTrend contains two kinds of hidden layers: extraction layer and prediction layer. Extraction layer, a fully connected layer, is designed to extract the time-variant trend by feeding the original flow series and corresponding simple average trend series. Prediction layer, an LSTM layer, is used to predict the incoming traffic flow which is computed by adding the predicted trend and residual value together. The prediction layer is fed by the obtained time-variant trend series from extraction layer and the residual series calculated by subtracting the obtained trend from original flow. In a sense, DeepTrend can be regarded as a special detrending method which decomposes the flow time series into trend and residual series.

In order to make the network learn the flow patterns better, avoid the deep network falling into a local minimum during training, and speed up the convergence, we implement the method that first pre-training the network layer-by-layer and then fine-tuning the entire network. That is to say, we first use the original flow series concatenated with simple average trend as input and simple average trend also as output to pre-train the extraction layer to enable the layer to extract the simple average trend from input. Then, we take the output from extraction layer and corresponding calculated residual series as input and the incoming trend and residual value as output to pre-train the prediction layer. Finally, we train the total network using a small learning rate by feeding the original flow series and simple average trend series as input and the predicted flow value in next time step as output, after which the output trend from extraction layer will be time-variant and contains information from original flow.

The pre-training process enables the network to make prediction using the simple average detrending method, and the fine-tuning process enables the network to be self-adaptive and make full use of time-variant trend for prediction. This training scheme will make the model further improve the performance for traffic flow forecasting.

IV. Experiments

A. Dataset

We evaluate the performance of DeepTrend on PeMS dataset [34]. In the dataset, the traffic data are collected every 5 minutes in freeway systems across California, and we only focus on the traffic data in district 4. In this paper, the data collected in the first 16 weeks of 2016 are used for experiments. The first 12 week' data are selected as the training set, and the remaining 4 week' data are selected as the test set.

Considering the limited computational resource, in the experiment, we select 50 stations in district 4 to forecast traffic flow. For the missing data in theses stations, we impute them using simple average trend. Before feeding into the model, the flow data in each station are first normalized to be zero mean and unit variance.

B. Performance Index

In the experiment, we use mean square error (MSE) to evaluate the performance of the proposed model. The performance index is defined as

$$\mathrm{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \tag{12}$$

where y_t and $\hat{y}_t$ are the actual traffic flow and predicted traffic flow.

TABLE I: Performance comparison of different models.

	Model	MSE
Original data based	ARIMA	1129.89
	MVLR	1138.60
	SVR	1062.94
	RF	1110.54
	LSTM	1072.23
Detrending based	ARIMA	1028.64
	MVLR	1036.78
	SVR	1031.51
	RF	1085.31
	LSTM	1024.43
	DeepTrend	**984.47**

C. Predictor Architecture Settings

Considering the temporal correlation, we use the previous N steps data as $\mathbf{x} = (x_{t-N+1}, x_{t-N+2}, \cdots, x_t)$ to predict traffic flow in next time step denoted as $\hat{x}_{t+1}$. In our experiment, N is set to 12. That is to say, we use the history values within the last 1 hour to predict traffic flow in next 5 minutes.

There are several parameters in our prediction architecture that need defining and tuning. For DeepTrend, the extraction layer contains 128 neurons units and the prediction layer contains 128 LSTM units. The activation functions of the two hidden layers are both ReLU. The optimization algorithm is using Adam [35]. For ARIMA model, the number of lag order p is 12, the degree of differencing d is 0, and the order of moving average q is 1. For SVR, the penalty parameter C is 1.0, and RBF kernel is used. For random forests (RF), the number of trees and maximum depth for each tree are both 10. For LSTM, one-layer LSTM network is adopted and it has 128 LSTM units. The activation function for the hidden layer is ReLU, and for the output layer is a linear function. The optimization algorithm is also Adam.

In the experiment, we use Keras [36] framework to build DeepTrend and LSTM models, and use scikit-learn [37] library to build MVLR, SVR, and RF models.

D. Experimental Results

We compare the performance of the proposed DeepTrend with the traditional models like ARIMA, MVLR, SVR, RF, and deep network LSTM. The comparative models are tested based on original flow and detrending methods. We calculate the average MSE in 50 test stations for a 5-min traffic flow prediction of each model. The results are shown in Table I.

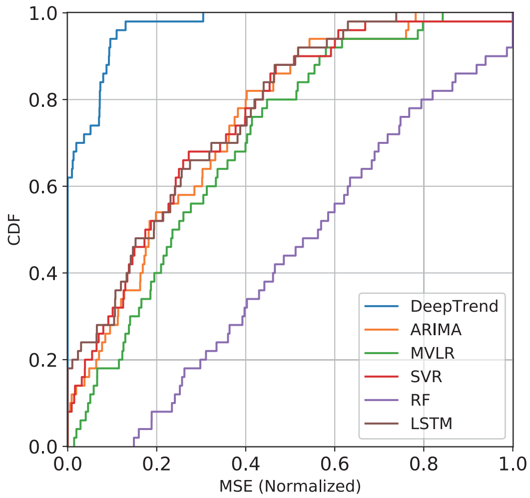

Fig. 5: Empirical CDF of MSE for 50 test stations.

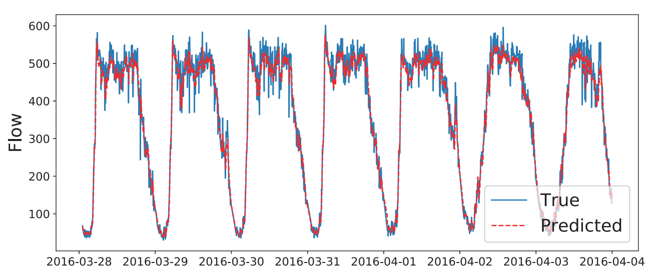

Fig. 6: Traffic flow prediction in station (ID 40006) from March 28, 2016 to April 3, 2016.

From the table, we can see that (1) simply using deep network LSTM does not outperform other traditional models if just fed by original flow; (2) detrending based models significantly outperform the original data based models, and (3) the proposed DeepTrend performs better than other detrending based models.

If the original flow time series data are used in prediction, SVR performs best in terms of MSE. Although LSTM as a deep network is adept in dealing with time series and learning the data representation, in the experiment, simply using an LSTM network still has not learned most intra-pattern of original flow series and is not dominant compared with the traditional model SVR. This may be caused by lack of training data, which limits the advantage of deep learning model, and we will make further experiment using larger dataset.

If detrending based methods are used, all models have gained significant enhancement in prediction performance. This demonstrates that detrending method which eliminates the periodic trend pattern in flow and only considers residual series is quite effective for traffic prediction. For detrending based models, LSTM performs better than traditional models, which shows that LSTM is adept in predicting residual flow series.

As shown in Table I, the proposed DeepTrend makes MSE drop to 984.47, which noticeably outperforms other models. A visual display of performance comparison is given in Fig. 5. It presents the cumulative distribution function (CDF) of MSE for DeepTrend and the detrending based models, which describes the statistical results on 50 test stations. In the figure, MSE has been first normalized between 0 and 1 for the test results of all models in each test station. We can find that DeepTrend outperforms other models in most of test stations, which demonstrates that the proposed model is effective and promising.

We also plot in Fig. 6 the profile of predicted flow values of the proposed DeepTrend with respect to the true flow values of station (ID 40006) in one week. From the figure, we can see that the predicted values can well track the true values, which demonstrates that the proposed DeepTrend is effective for traffic flow in practice.

V. Conclusion

In this paper, we explore whether the deep network LSTM can learn the temporal pattern of traffic flow in prediction. Experiments reveal that simply using LSTM is not superior to some traditional machine learning models like SVR if detrending is not used, showing that it does not learn the patterns in traffic flow. To better capture the temporal pattern, we propose DeepTrend, a deep hierarchical neural network which integrates the process of pattern extraction and flow prediction. Compared with traditional LSTM, DeepTrend needs pre-training layer-by-layer and then fine-tuning in the entire network. The extraction layer is used to learn the temporal pattern and extract the time-variant trend in flow series, and the prediction layer is to make a prediction for incoming flow which is fed by output series from extraction layer and calculated residual series. The experiments show that DeepTrend outperforms LSTM and other baselines based on detrending methods.

We only take account of temporal pattern in this paper. For future work, it would be considered that making the deep network learn the spatial correlations between different stations and integrating the temporal-spatial dependence in one network for traffic flow prediction.

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, Sep. 2010.

[2] X. Zheng, W. Chen, P. Wang, D. Shen, S. Chen, X. Wang, Q. Zhang, and L. Yang, "Big data for social transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 620–630, Mar. 2016.

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

[4] M. Levin and Y.-D. Tsao, “On forecasting freeway occupancies and volumes,” *Transportation Research Record*, no. 773, pp. 47–49, 1980.

[5] S. Lee and D. Fambro, “Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1678, no. 99, pp. 179–188, Jan. 1999.

[6] Y. Kamarianakis and P. Prastacos, “Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches,” *Transportation Research Record*, vol. 1857, no. 1, pp. 74–84, Jan. 2003.

[7] 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, Nov. 2003.

[8] L. Li, X. Su, and Y. Zhang, “Trend modeling for traffic time series analysis : An integrated study,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 1–10, Dec. 2015.

[9] R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg, “Different methods of traffic forecast based on real data,” *European Journal of Operational Research*, vol. 155, no. 3, pp. 558–568, Jun. 2004.

[10] X. Jin, Y. Zhang, and D. Yao, “Simultaneously prediction of network traffic flow based on PCA-SVR,” in *Advances in Neural Networks ISNN 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4492 LNCS, no. PART 2, pp. 1022–1031.

[11] F.-Y. Wang, X. Wang, L. Li, and L. Li, “Steps toward parallel intelligence,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 4, pp. 345–348, Oct. 2016.

[12] F.-Y. Wang, 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, Apr. 2016.

[13] L. Li, Y.-L. Lin, D.-P. Cao, N.-N. Zheng, and F.-Y. Wang, “Parallel learning — a new framework for machine learning,” *Acta Automatica Sinica*, vol. 43, no. 1, p. 1, 2017.

[14] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[15] 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: Emerging Technologies*, vol. 72, pp. 168 – 181, 2016.

[16] 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. 1–9, 2014.

[17] Y. Tian and L. Pan, “Predicting short-term traffic flow by long short-term memory recurrent neural network,” *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 153–158, Dec. 2015.

[18] W. Huang, G. Song, H. Hong, and K. Xie, “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, Oct. 2014.

[19] R. Fu, Z. Zhang, and L. Li, “Using lstm and gru neural network methods for traffic flow prediction,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nov. 2016, pp. 324–328.

[20] C. Chen, Y. Wang, L. Li, J. Hu, and Z. Zhang, “The retrieval of intra-day trend and its influence on traffic prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 103–118, Jun. 2012.

[21] L. Li, X. Su, Y. Wang, Y. Lin, Z. Li, and Y. Li, “Robust causal dependence mining in big data network and its application to traffic flow predictions,” *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 292–307, Sep. 2015.

[22] Z. Li, Y. Li, and L. Li, “A comparison of detrending models and multi-regime models for traffic flow prediction,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 34–44, 2014.

[23] G. a. Davis and N. L. Nihan, “Nonparametric regression and short-term freeway traffic forecasting,” *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[24] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, “Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6164–6173, 2009.

[25] G. Leshem and Y. Ritov, “Traffic flow prediction using Adaboost algorithm with random forests as a weak learner,” *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, vol. 1, no. 1, pp. 2–7, 2007.

[26] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[27] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, Jun. 2005.

[28] 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 Levenberg-Marquardt algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, Jun. 2012.

[29] M. Zhong, S. Sharma, and P. Lingras, “Short-term traffic prediction on different types of roads with genetically designed regression and time delay neural network models,” *Journal of Computing in Civil Engineering*, vol. 19, no. 1, pp. 94–103, Jan. 2005.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[33] P. J. Werbos, “Backpropagation through time: What it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[34] Caltrans, performance measurement system (pems). [Online]. Available: http://pems.dot.ca.gov

[35] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[36] F. Chollet. Keras. [Online]. Available: https://github.com/fchollet/keras

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.