2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)
Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

Long Short-Term Memory Model for Traffic Congestion Prediction with Online Open Data

Yuan-yuan Chen*†, Yisheng Lv*, Zhenjiang Li*, and Fei-Yue Wang*‡
*State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
Email: zhenjiang.li@ia.ac.cn
†University of Chinese Academy of Sciences
Beijing, 100049, China
‡Qingdao Academy of Intelligent Industries
Qingdao, Shandong, 266109, China

***Abstract*— Traffic congestion in metropolitan areas has become more and more serious. Over the past decades, many academic and industrial efforts have been made to alleviate this problem, among which providing accurate, timely and predictive traffic conditions is a promising approach. Nowadays, online open data have rich traffic related information. Typical such resources include official websites of traffic management and operations, web-based map services (like Google map), weather forecasting websites, and local events (sport games, music concerts, etc.) websites. In this paper, online open data are discussed to provide traffic related information. Traffic conditions collected from web based map services are used to demonstrate the feasibility. The stacked long short-term memory model, a kind of deep architecture, is used to learn and predict the patterns of traffic conditions. Experimental results show that the proposed model for traffic condition prediction has superior performance over multilayer perceptron model, decision tree model and support vector machine model.**

I. Introduction

With the steadily increasing number of vehicles, road traffic congestion has become an increasingly important issue [1]–[3]. To alleviate traffic congestion, providing accurate, timely and predictive traffic condition information to the participants of transportation systems, especially to the drivers, will be a practical and promising approach [4], [5]. However traffic data from traditional sensors, e.g., inductive loops and microwave radar detectors, are not accessible to researchers in most cases. With the development of the World Wide Web, online data are much easier to access by individuals. This leads us to utilize open data to help travelers to make decisions autonomously according to the real time traffic condition.

In the context of transportation systems, open data are traffic related and freely available to traffic participants in technique. Recently, some academic and industrial efforts have been made into mining traffic related open data, which can be classified into two classes: traffic flow prediction and traffic incident detection.

Traffic flow prediction is a key functional component in intelligent transportation systems (ITSs) [6]–[8]. With the development of social transportation [9], traffic flow prediction methods are enriched. He et al. improved long-term traffic volume prediction by incorporating tweets semantics [10]. They found traffic volume and the number of tweets posted are statistically correlated. And they proposed an optimization framework to minimize the prediction error meanwhile obtain a matrix to transform tweet words to traffic indicators. Ni et al. incorporated tweets rate and semantic features into four different regression models to predict short-term traffic volume under the sport games events [11]. They found that tweet features can improve prediction performance for all four models. Grosenick extracted traffic incidents from tweets manually and combined them with traffic speed data to make predictions [12]. Although incorporating tweet semantics does not improve overall prediction performance, this work points out the future work in this domain should be focused on reducing uncertainty produced by noisy data and by modeling approaches.

In the field of traffic incident detection, D'Andrea et al. labeled tweets as traffic and non-traffic classes manually and trained a support vector machine (SVM) model with a balanced dataset [13]. Experiment results show the proposed method achieves better classification accuracy. Liu et al. presented an application to detect traffic events, where a wavelet analysis model was applied to tweets [14].

Mining social media to obtain traffic related information is an emerging field and gains increasing interests [15]. In addition to social media, web based map (e.g., Google Map), weather websites and online tickets sales are potential resources to extract traffic related information. In this paper, we utilized the historical traffic conditions obtained from AMAP, a web map service provider in China, to predict traffic conditions in the future. The long short-term memory (LSTM) model was trained to learn the patterns in traffic condition sequences.

The rest of the paper is organized as follows: Section II reviews potential resources containing traffic information and presents methods to access them. Section III presents the LSTM approach for traffic condition prediction. Section IV gives experimental results. Section V concludes our work and points out possible directions for the further research.

II. Traffic Related Online Open Source Data

In this section, we review four categories of the existing online open source services that offer traffic related data. Two kinds of methods to collect open source data are also presented.

A. Categories of Traffic Related Online Open Data

There are mainly four categories of online open data services that offer traffic related data:

- Web Map, represented by Google Maps and official websites of traffic management and operations.
- Social Media, represented by Twitter, Micro blog and Instagram.
- Local Events, represented by Damai and Ticketmaster.
- Weather Websites, represented by weatherUSA.

Web map is a kind of service that provides online map through websites or mobile applications. Despite of highlighting the relationships between spatial elements as the traditional geographical map do, it offers a number of other services, such as positioning and real time traffic conditions. We take Google Maps as an example to show what web map services provide. Google Maps offers satellite imagery, street maps, real-time traffic conditions, and route planning for traveling by foot, car, bicycle, or public transportation. The former two services are beyond the scope of this paper. Real-time traffic conditions, provided by Google Traffic, are shown by different colors. To be specific, green, orange, red and dark red represents fast, slow, congestion and more serious congestion, respectively. Google Maps uses traffic data from cellular telephone in a crowd-sourcing way and from road detectors, such as loop detectors, if applicable. As for route planning, travelers can obtain a specific route with estimated time of travel for a couple of orientation and destination. With above information, travelers can make better decisions before a trip or adjust their plans en-route. In addition to real time traffic information, heatmap, depicting the intensity of individuals, is also potential to be used to extract traffic information, which could be used to estimate traffic demands further.

Social media, consisting of self-generated contents is more and more popular. People are more pleased to share what has happened in their life than ever. And individuals may post texts about the traffic jams or traffic incidents, e.g. "what a traffic jam, I am going to be late again." To mining text messages, natural language processing methods and techniques, e.g. named entity recognition [16] and sentiment analysis [17], are needed. In addition to text message, photos or images also contains traffic related information. As far as we know, only text messages in social media are used to extract traffic information. In the future, image processing and analysis should be applied to photos posted in social media.

Local events and weather are important factors to influence traffic [18], [19]. Tickets sales are the best sources to get local event information. They would publish local event (like sports games and concerts) schedules in advance. The information contains when and where the events would be held, which will help us estimate the traffic demand. And weather, especially extreme weather (like storms) should be incorporated to mining traffic data. Besides extreme weather, haze in developing countries, especially in China, should be taken into consideration. The visibility in heave haze is impaired, which would affect the traffic speed and traffic demand.

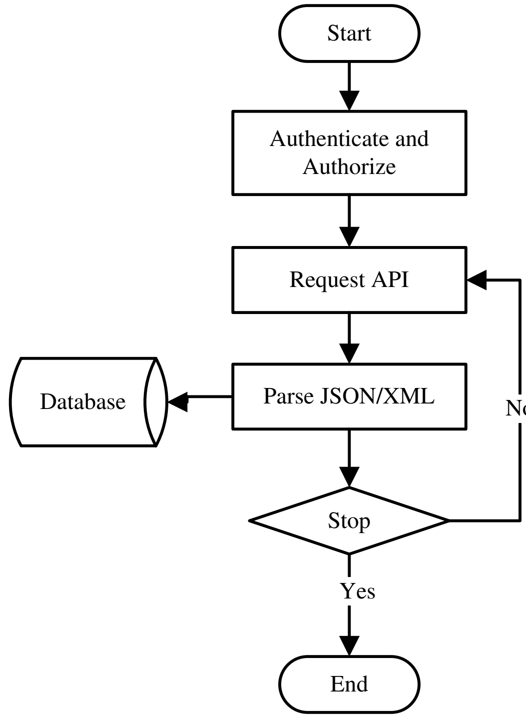

Fig. 1. Diagram of requesting APIs

B. Methods to Access Online Open Data

The open source data, fed into traffic data mining, are mostly not designed for scientific research, which means additional efforts are needed to collect these data. There are two types of methods to access the open source data: requesting Application Programming Interfaces (APIs) and crawling the websites. The former approach, usually not free, is provided for developers officially, and has requesting rate limits. Crawling the websites is more comprehensive than requesting through the API directly. However it is free and has no rate limits with technical supports.

1) Requesting APIs: As for API, it is easy to access the data by sending a HTTP GET request or a HTTP POST request. The diagram of this procedure is shown in Fig. 1. For security reasons, the first step is to get the authentication and authorization. Then it is permitted to access authorized data by request APIs over HTTP. Most services support JSON, XML, RSS and Atom response formats. The next step is to parse the response resources and save the data into database. This process, starting from requesting APIs, should be iteratively executed until reaching the stop criteria.

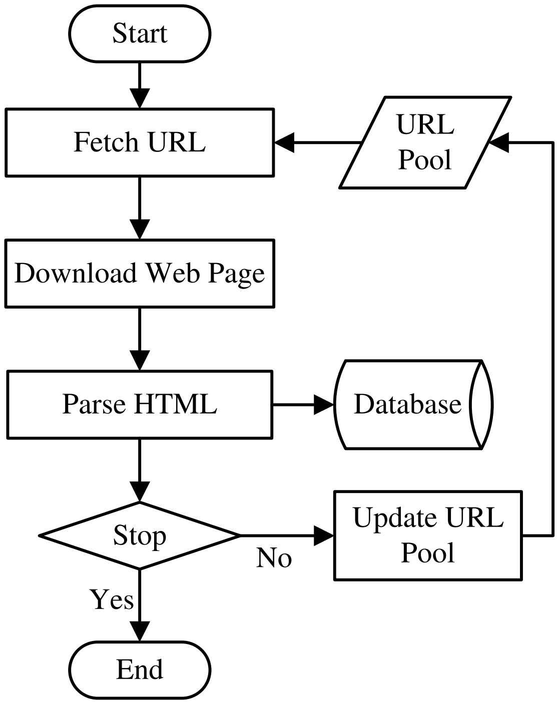

Fig. 2. Diagram of crawling web pages

2) Crawling Web Pages: Compared to the API approach above, crawling the web pages is technically free but more complicated. The diagram of this procedure is illustrated in Fig. 2. It begins with fetching a uniform resource locator (URL) and downloads its web page. It is worth to mention that web servers usually design anti-crawling strategies, which will prevent crawler to access the websites. To tackle this problem, using a proxy and accessing within a random time interval should be adopted. After successfully downloading the web page, it needs to obtain target urls before saving data into database.

III. Methodology

In this section, the LSTM model, as a particular type of Recurrent Neural Network (RNN), is introduced.

A. RNN

The RNN is a neural network that has a feedback loop compared to feedfoward neural networks [20]. Fig. 3 gives an illustration of a standard RNN model. Given a length T input sequence $(x_1, x_2, \ldots, x_T)$, the RNN maps the input x_t and previous hidden state h_{t-1} to current hidden state h_t as in (1).

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

where W_{xh}, W_{hh} are weight matrices from the input layer to the hidden layer and from the hidden layer to the hidden layer, respectively, b_h is a bias vector to the hidden layer, and ϕ is the activation function of the hidden layer. For the output units,we have

$$z_t = \sigma(W_{hz}h_t + b_z) \tag{2}$$

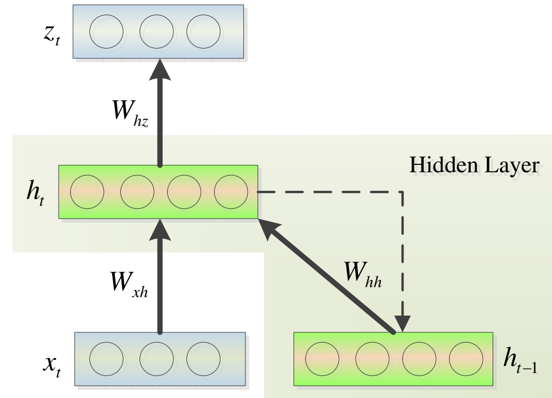

Fig. 3. Architecture of standard RNN

where W_{hz} is a weight matrix from the hidden layer to the output layer, b_z is a bias vector to the output layer and σ is the activation function of output layer. The complete sequence of hidden states and outputs can be computed by recursively applying (1) and (2) from $t = 1$ to $t = T$.

The loss of the RNN is the sum of losses over all time steps

$$L(z, y) = \sum_{t=1}^{T} L(z_t; y_t) \tag{3}$$

where y_y is the target class in the output sequence at time t. By minimizing the loss function, we optimize the RNN model by updating its parameters. To compute the derivatives of loss function with respect to model parameters, we employ back-propagation through time (BPTT) algorithm [21], [22].

Although RNN has potential to learn the contextual dependency and it is easy to compute its gradients, the range of input sequence is limited in practice due to the vanishing gradient problem [23], [24]. Various efforts were made to tackle this problem since 1990s. Among them, LSTM model, as a variant of standard RNN, was proposed by Sepp Hochreiter and Juergen Schmidhuber [25].

B. LSTM

The LSTM architecture is very similar to the standard RNN as described above, and the only difference between them is in the hidden layer [26]–[28]. Fig. 4 gives an

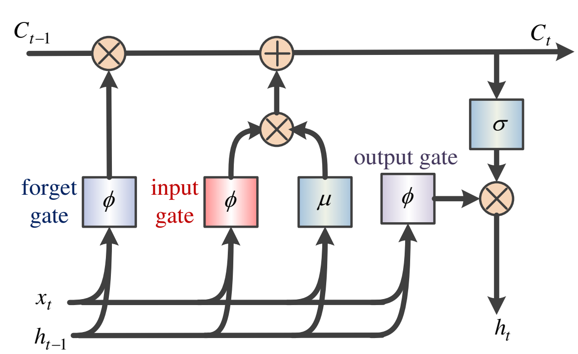

Fig. 4. Hidden layer of LSTM model

illustration of the hidden layers of LSTM model. In LSTM, there is a memory block with cell, input gate, forget gate and output gate. The core behind LSTM is the cell state, which is only changed by linear interactions. This basic change helps to tackle the vanishing gradient problem as it enables the cell to store and read long range contextual information. The structures called gates are for removing information from or adding information to cell state. Given input x_t at time t, cell state C_{t-1} at time $t-1$ and hidden output h_{t-1} at time $t-1$, to compute cell state C_t and hidden output h_t, we firstly should compute forget gate:

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

input gate:

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

and output gate:

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

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 always sigmod. Then we obtain cell state:

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

and 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 feedback progress to update model parameters is the same with RNN described above, which is omitted for clarity.

C. Training Algorithm

It is a common approach to train a classifier on the training dataset and then classify instances in test dataset offline. However, as the sequences fed into LSTM model are temporally dependent, this approach usually has bad performance. In this paper, we firstly trained the model on the training dataset by a batch learning approach. And then we classified a instance in the test dataset, after which we trained the model with this instance in an online approach. The details of this procedure are summarized in Algorithm 1.

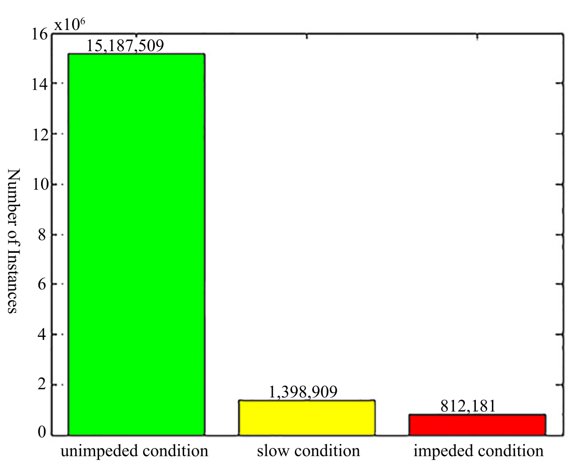

Fig. 5. Distribution of traffic conditions in dataset

Algorithm 1 Training and Prediction

Step 1: Batch learning on training dataset

for $b = 1$ to n **do**
  Randomly choose a minibatch of m samples $\{x^{(1)}, ..., x^{(m)}\}$
  for $mb = 1$ to m **do**
    Perform forward propagation on x^{mb} to compute loss function
  end for
  Accumulate the loss function values
  Perform backward propagation through time to update model parametes
end for

Step 2): Prediction and online learning on test dataset

for $i = 1$ to p **do**
  Make prediction on x^i in testing dataset
  Perform forward propagation on x^i to compute loss function
  Perform backward propagation through time to update model parametes
end for

IV. Experiments

A. Data Description

The proposed LSTM model was applied to the data collected from AMAP [29] between December 28, 2014 and February 3, 2015. The traffic condition data were collected every five minutes covering 1649 segments of arterial roads in Beijing, China. There are three categories of traffic conditions, i.e., unimpeded condition, slow condition and impeded condition. In the dataset, there are 15,187,509 unimpeded conditions, 1,398,909 slow conditions and 812,181 impeded conditions, which is shown in Fig. 5. To illustrate the patterns of traffic conditions, we sampled 20 segments randomly. And the traffic condition time series during 06/01/2015 and 07/01/2015 are shown in Fig. 6. It is obvious that traffic congestions occur during rush hours for most segments. In this paper, the former 9000 sequences of each segment were selected as the training set and the last 1551 sequences were selected as the testing set.

B. Index of Performance

To evaluate the performance of the proposed deep architecture, we adopt three performance indexes. They are precision:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

recall:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

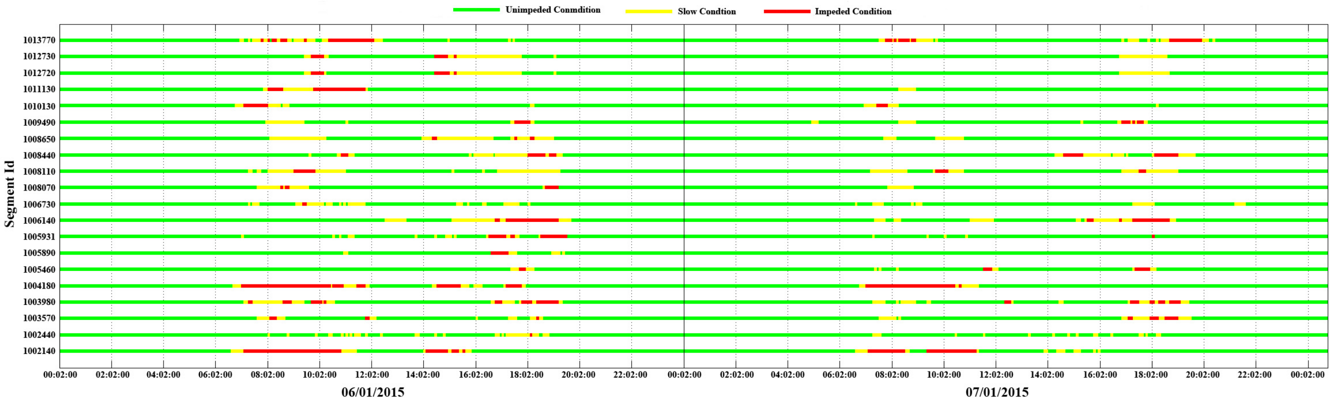

Fig. 6. Sample of traffic conditions during 06/01/2015 and 07/01/2015

and F-measure:

$$F_\beta = (1+\beta^2)\frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{11}$$

where TP, FP and FN is true positive, false positive and false negative respectively, and β is a non-negative real coefficient. And F-measure is called F_1 measure when β equals to 1.

C. Determine the Structure of a LSTM model

To use the temporal context, traffic conditions of last 30 minutes were selected to predict 30-min traffic condition. Specificly, traffic condition sequence:

$$(x_{t-5}, x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t)$$

is utilized to predict the traffic condition x_{t+5} at time $t+5$. It is worth noting that performances were not improved by utilizing the day of week and time of day features according to our experiments. Thus they are not taken into consideration in this paper. We apply one hot method to encode the traffic conditions and 3 bit binary codes are needed to represent traffic conditions. Thus the dimension of a input sequence is $(3, 6)$, whereas the dimension of output is 3, where 3 represents the bit of codes and 6 represent the length of a sequence.

We choose the number of output units in a LSTM layer from $\{3, 6, 9, 12, 36\}$ and decide whether to stack 3 LSTM layers on top of each other or not. The learning rate is chosen from 0.1 to 1.2 with a step 0.1. After performing grid search, stacked LSTM with output units size of each layer is 6, 6, and 6, respectively, is the best architecture as for the indexes in this paper.

D. Results

We compared the performance of the proposed stacked LSTM model with the multilayer perceptron (MLP) model, the decision tree model, and the SVM model. Among these three competing methods, the MLP model has good performance for the traffic flow prediction [6]-[7], and decision tree and SVM are relatively effective and advanced models for prediction [30], [31]. It is worth to mention that all baseline methods utilized the same feature with the LSTM model.

The prediction results of unimpeded condition, slow condition and impeded condition are given in Table I. In Table I, we can see the precision, recall and F_1 measure of MLP, SVM and decision tree are close. Taking F_1 measure as an example, the MLP method achieves 0.9568, 0.5119 and 0.6200 for unimpeded condition, slow condition and impeded condition respectively. And we can also find that stacked LSTM has the highest score at precision, recall and F_1 measure at the same time than the other three methods. It is worth to mention that the data are dominated by the instances of unimpeded conditions, which means traffic is unimpeded in most of the time except some special cases, i.e. the peak hours and durations of traffic accidents. The improvement of F_1 measure for slow condition and impeded condition by stacked LSTM is up to 40% and 23%. Stacked LSTM is clearly superior over other three methods on this unbalanced dataset.

V. Conclusions and Future Works

A. Conclusions

Online open data are potential sources to be used to provide traffic related information. In this paper, we review four categories of open data platforms, i.e., web map, social media, local events and weather website. As a case study, we collect traffic condition data from web map provider to learn the hidden patterns of traffic conditions. We propose a deep learning approach with a stacked LSTM model for traffic condition prediction. Online training was applied during the test phrase. We compared the performance of the proposed stacked LSTM model with MLP model, decision tree model and SVM model. Experiments show the proposed method is superior to the competing methods.

B. Future Works

As the dataset is dominated by the instances of unimpeded condition, the prediction performance on slow condition and impeded condition are not as well as on unimpeded condition. For future research, other machine learning techniques, such as ensemble method, should be adopted to improve the prediction performances on unbalanced dataset. Meanwhile,

TABLE I
PERFORMANCE COMPARISON OF MLP, DECISION TREE, SVM AND STACKED LSTM

Task	Unimpeded Condition			Slow Condition			Impeded Condition		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
MLP	0.9418	0.9721	0.9568	0.5930	0.4503	0.5119	0.6517	0.5913	0.6200
Decision Tree	0.9751	0.9403	0.9574	0.4447	0.6110	0.5147	0.5929	0.6703	0.6292
SVM	0.9755	0.9397	0.9573	0.4454	0.6084	0.5143	0.5853	0.6801	0.6291
Stacked LSTM	0.9754	0.9864	0.9808	0.7418	0.6972	0.7188	0.8263	0.7270	0.7734

social media, local events and weather information are expected to be incorporated into the prediction model in our future work.

ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation of China under Grant 61533019, 71232006, 61233001, 61304201, Guangdong Provincial Department of Science and Technology under Grant 2016B090918004, 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," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, Sept 2010.

[2] N. Zhang, F. Y. Wang, F. Zhu, D. Zhao, and S. Tang, "Dynacas: Computational experiments and decision support for its," *IEEE Intelligent Systems*, vol. 23, no. 6, pp. 19–23, Nov 2008.

[3] 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, Dec 2011.

[4] H. Dia, "An agent-based approach to modelling driver route choice behaviour under the influence of real-time information," *Transportation Research Part C-emerging Technologies*, vol. 10, no. 5, pp. 331–349, 2002.

[5] J. Zhou, C. L. P. Chen, L. Chen, and W. Zhao, "A user-customizable urban traffic information collection method based on wireless sensor networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1119–1128, Sept 2013.

[6] 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, April 2015.

[7] L. Li, X. Su, Y. Zhang, Y. Lin, and Z. Li, "Trend modeling for traffic time series analysis: An integrated study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3430–3439, Dec 2015.

[8] 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, 2012.

[9] 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, March 2016.

[10] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence, "Improving traffic prediction with tweet semantics." in *IJCAI*, 2013.

[11] M. Ni, Q. He, and J. Gao, "Using social media to predict traffic flow under special event conditions," in *The 93rd Annual Meeting of Transportation Research Board*, 2014.

[12] S. Grosenick, "Real-time traffic prediction improvement through semantic mining of social networks," Ph.D. dissertation, University of Washington, 2012.

[13] E. D'Andrea, P. Ducange, B. Lazzerini, and F. Marcelloni, "Real-time detection of traffic from twitter stream analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2269–2283, Aug 2015.

[14] M. Liu, K. Fu, C.-T. Lu, G. Chen, and H. Wang, "A search and summary application for traffic events detection based on twitter data," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 549–552.

[15] N. Wanichayapong, W. Pruthipunyaskul, W. Pattara-Atikom, and P. Chaovalit, "Social-based traffic information extraction and classification," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, Aug 2011, pp. 107–112.

[16] S. Jabeen, S. Shah, and A. Latif, "Named entity recognition and normalization in tweets towards text summarization," in *Digital Information Management (ICDIM), 2013 Eighth International Conference on*, Sept 2013, pp. 223–227.

[17] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, 2010, pp. 1320–1326.

[18] T. Maze, M. Agarwai, and G. Burchett, "Whether weather matters to traffic demand, traffic safety, and traffic operations and flow," *Transportation research record: Journal of the transportation research board*, no. 1948, pp. 170–176, 2006.

[19] T. F. Golob and W. W. Recker, "Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions," *Journal of transportation engineering*, vol. 129, no. 4, pp. 342–353, 2003.

[20] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, Mar 1994.

[21] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8624–8628.

[22] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6645–6649.

[23] 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, Mar 1994.

[24] J. F. Kolen and S. C. Kremer, *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. Wiley-IEEE Press, 2001, pp. 237–243. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5264952

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

[26] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[27] J. Schmidhuber, F. Gers, and D. Eck, "Learning nonregular languages: A comparison of simple recurrent networks and lstm," *Neural Computation*, vol. 14, no. 9, pp. 2039–2041, 2002.

[28] A. Graves, N. Jaitly, and A. r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 273–278.

[29] "Amap," http://ditu.amap.com/, accessed: 2016-09-01.

[30] M. Gan, C. Philip Chen, L. Chen, and C.-Y. Zhang, "Exploiting the interpretability and forecasting ability of the rbf-ar model for nonlinear time series," *International Journal of Systems Science*, vol. 47, no. 8, pp. 1868–1876, 2016.

[31] M. Gan, C. L. P. Chen, H. X. Li, and L. Chen, "Gradient radial basis function based varying-coefficient autoregressive model for nonlinear and nonstationary time series," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 809–812, July 2015.