Attention-based Recurrent Neural Network for Traffic Flow Prediction

Qi Chen, Wei Wang, Xin Huang, Hai-ning Liang

Department of Computer Science and Software Engineering, Xi'an Jiangtong-Liverpool University, China
{qi.chen, wei.wang03, xin.huang, haining.liang}@xjtlu.edu.cn

Abstract

Traffic flow prediction is an important while complex problem in transportation modeling and management. Many uncertain, non-linear and stochastic factors could have large influence on the prediction performance. With the recent development in deep learning, researchers have applied deep neural networks for the traffic flow prediction problem and achieved promising results. However, existing studies still have some issues unaddressed, e.g., the models only predict the traffic flow at next time step while travelers may need a sequence of predictions to make better, long-term decisions; temporal factors are (e.g., day of the week, national holiday) usually not well considered during prediction. To address these limitations, this paper proposed an attention-based recurrent neural network architecture for multi-step traffic flow prediction. Experimental results demonstrate that the proposed method has superior performance compared to the existing models. We also show how the method can be used to develop traffic anomaly detection systems.

Keywords: Traffic flow prediction, Recurrent neural network, Attention mechanism, Deep Learning, long short-term memory

1 Introduction

With the development of Internet of Things (IoT), embedded devices, e.g., sensors, actuators, mobiles phones and RFIDs, can be built into every fabric of urban environments and connected with each other. Data generated by these devices can be preprocessed, integrated, and made available in standard formats through open services [1]. Many machine learning techniques, e.g. classification, regression and clustering methods, have been applied to process and analyse IoT data to extract useful knowledge. Real-world applications have been developed and deployed to help citizens better understand their surroundings and informs city authorities to provide better and more efficient public services, for example intelligent transportation [2], healthcare [3], environment monitoring [4], and public safety [5].

Traffic flow information is crucial for individual travelers, business sectors, and government agencies to make better travel decisions, alleviate traffic congestion, and improve traffic operation efficiency [6]. With the rapid development and deployment of intelligence transportation systems (ITSs), traffic flow prediction has gained increasing attention in recent years. With the widespread traffic sensors, available traffic data (e.g., loop sensors, GPS, cameras, social media, etc.) for analysis is exploding; with advanced networking technologies, big traffic data can be efficiently and securely collected, processed, cached, shared and delivered [7-11]. Through sophisticated analysis of historical and real-time traffic data, ITSs enable users to make safer, more coordinated, and smarter use of the transport networks.

Recently, deep learning has drawn a lot of academic and industrial attention in various areas, including intelligent transportation. It is a relatively "young" learning paradigm in the machine learning family, and has in fact its origin from Artificial Neural Networks (ANNs). It allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction and is able to discover intricate structures from natural data in its raw forms without the needs of sophisticated feature engineering and tuning [12]. Compared to traditional machine learning methods, deep learning can model extremely sophisticated functions through multiple layers of non-linear transformation trainable from the beginning to the end. Methods based on deep models have significantly improved the state-of-the-art in an array of problems, such as natural language processing (more specifically, neural machine translation) [13], computer vision [14], and speech recognition [15].

Deep learning-based methods have been applied to capture traffic patterns and provide traffic predictions even without prior knowledge in transportation areas, and have shown promising results [16-21]. However, these studies still leave some issues unaddressed. For example, they mostly only focus on predicting traffic flow at next time step (e.g. after 15 mins), while

*Corresponding Author: Wei Wang; E-mail: wei.wang03@xjtlu.edu.cn
DOI: 10.3966/160792642020052103020

travelers may need a sequence of traffic flow predictions (e.g., traffic flow sequence for the next few hours) in order to make better, longer-term travel decisions. Meanwhile, temporal information (e.g., time of the day, day of the week, national holiday) of traffic flow is usually not considered, e.g., weekdays, weekends and national holidays should be considered separately. To address the above limitations, we propose an Attention-Based Recurrent Neural Network architecture with Temporal Component (ABRNN_TC) for traffic flow prediction. The proposed model is compared to other baseline methods and shows superior performance. To further illustrate the use of the method, we present a case study in real-time traffic event detection.

The rest of the paper is organised as follows. Section 2 reviews the existing studies on traffic flow prediction. Section 3 presents the details of the proposed attention-based recurrent neural network architecture for traffic flow prediction. Section 4 shows the experimental and evaluation results. The conclusion and future work are presented in Section 5.

2 Related Work

Over the past few decades, a number of traffic flow prediction models have been proposed, e.g., ARIMA [22], k-NN [23], SVR [24] and ANN [25]. The Autoregressive Integrated Moving Average (ARIMA) model focuses on finding the patterns of the temporal variation of traffic flow and has been applied to short-term traffic flow prediction [22]. Due to the stochastic and nonlinear nature of traffic flow, nonparametric approaches, e.g., k-NN [23], SVR [24] and ANN [25], have attracted much more attention from researchers. Chang *et al.* presented a dynamic multi-interval traffic volume prediction model based on the k-NN nonparametric regression [23]. Jeong et al. presented an online learning weighted support vector regression (SVR) for short-term traffic flow prediction [24]. There are a few major limitations about these traditional methods. First, for time-series based methods like ARIMA, a linear architecture is often preferred. Meanwhile, the future traffic flow is only predicted based on historical traffic flows on a particular road regardless of others, although a transportation system is a highly correlated network. Second, complex hand-engineered features are usually needed, which requires prior knowledge of transportation domain and is extremely time-consuming. Finally, most ANN based approaches use shallow architectures, e.g., one single hidden layer in [25], which may not be able to learn effective representation from the traffic flow data and results in poor performance.

Simply speaking, when a neural network contains more than one hidden layer, it is considered as a 'deep' architecture. A gradient descent algorithm called backpropagation (BP) [26] was proposed in 1980s to train neural network and has played an important role in neural network training since then. However, one of the major reasons that ANNs with multiple fully connected layers have not gained popularity in many real-world applications for decades is the computation complexity. In 2006, a breakthrough research [27] first showed that training deep neural network in an unsupervised manner (pre-training), followed by a supervised fine-tuning, could result in good performance. In 2012, the research group led by Hinton won the ImageNet competition by using convolutional neural network that almost halve the classification error rate [14]. Since then, the study of deep learning has achieved a series of milestones in various domains and has been applied to transportation research. The work of Huang et al. [16] was the first study to apply deep learning for the traffic flow prediction problem by using Deep Belief Network (DBN) to learn effective features in an unsupervised fashion. In addition, it introduced a multitask regression layer on the top of DBN for supervised prediction and reported around 5% improvement over the state of the art. As traffic is usually affected by other factors such as weather condition, the work in [17] developed a DBN based deep learning model to fuse traffic data and weather data for more accurate prediction. The studies in both [18] and [19] applied Stacked Auto-Encoder (SAE) for traffic flow prediction. While the work in [18] used a sparse Auto-Encoder for better feature extraction, the one in [19] used the Levenberg-Marquardt algorithm to train SAE for more stable convergence. As SAE and DBN cannot model temporal dependency in the data, Recurrent Neural Network (RNN) has also been applied for traffic flow/speed prediction [20-21]. Both work used LSTM based RNN for short-term traffic flow/speed prediction and reported better performance over other deep learning models.

3 Attention-based Recurrent Neural Network Architecture

In this section, we first briefly describe the technique background of RNN architecture, Long Short-Term Memory (LSTM) and encoder-decoder architecture. Then, we present the proposed Attention Based RNN (ABRNN) model.

3.1 Recurrent Neural Network (RNN)

A RNN contains links among neurons, and after unfolding it forms a directed graph along a sequence as shown in Figure 1. This allows RNN to process any type of data that can be modelled as temporal sequences of variable lengths, $X = (x_1, x_2, \ldots, x_T)$. At each time step t, the hidden state h_t of the RNN is updated according to Equation (1).

$$x_t = f(h_{t-1}, x_t) \quad \textbf{(1)}$$

where f is a non-linear function, which can be as simple as a sigmoid function and as complex as a LSTM unit. RNNs use the internal states to capture dependency among input data in a sequence, which makes them suitable to tasks such as natural language processing, speech recognition and traffic flow prediction where data demonstrates strong temporal correlations. As vanilla RNNs suffer from various limitations, they have not been used in real-world applications. In practice, two RNN units: LSTM [28] and Gated Recurrent Unit (GRU) [29] have been widely used. In this paper, LSTM is chosen as the RNN unit.

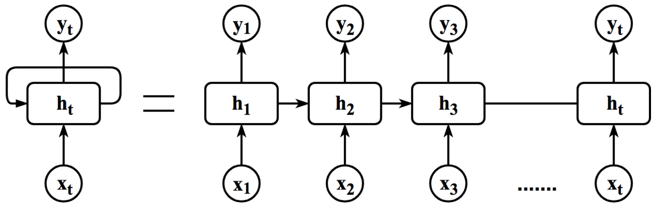


Figure 1. Illustration of a typical RNN architecture before and after unfolding

3.1.1 Long Short-Term Memory (LSTM)

As vanilla RNN suffers from the so-called exploding and vanishing gradient problem during training, a recurrent unit called LSTM [28] was proposed to solve the problem. The architecture of a standard LSTM unit is visualised in Figure 2.

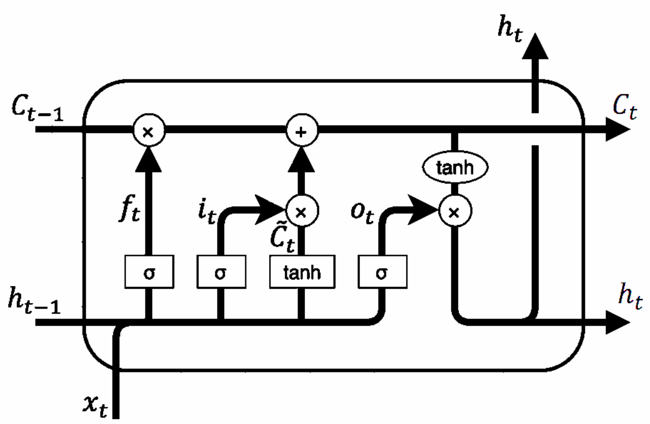


Figure 2. Long Short-Term Memory unit

A LSTM unit contains a cell C, an input gate i, an output gate o and a forget gate f. The cell is responsible for remembering values over time steps. The three gates allow LSTM memory cell to store and access information over long periods of time, thereby mitigating the vanishing gradient problem. Each of the three gates can be considered as a neuron that computes a value using an activation function σ. Given a sequence of input $X = (x_1, x_2, \ldots, x_T)$, a standard LSTM computes a sequence of outputs $Y = (y_1, y_2, \ldots, y_T)$ by iterating the following equations from (2) to (7):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \textbf{(2)}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \textbf{(3)}$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \textbf{(4)}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \textbf{(5)}$$

$$h_t = o_t * \tanh(C_t) \quad \textbf{(6)}$$

$$y_t = W_y \cdot h_t + b_y \quad \textbf{(7)}$$

Where f_t represents the output of the forget gate, i_t denotes the output of the input gate, and o_t is the output of the output gate. The cell state and hidden state are denoted as C_t and h_t, respectively. The weight matrices W, bias vectors b and sigmoid functions σ are utilised to build connections between input layer, hidden layer and output layer.

3.1.2 Encoder-Decoder Architecture

In some applications, e.g., machine translation, the input sentence and the desired target sentence usually have different lengths, which cannot be modeled with the standard RNN architecture. As such, the RNN based encoder-decoder architecture [29-30] has been proposed. It contains two RNNs, one learns to encode the source sequence into a vector representation and the other decodes the vector into the target sequence. This architecture allows to produce a sequence of predictions for time series data, e.g., traffic flow. Since each input time step may be of different importance to different output time step, an important extension is to add the attention mechanism [13], which adaptively selects the relevant hidden states from the encoder in order to produce the output at a particular time step.

3.2 Attention-Based Recurrent Neural Network (ABRNN)

The work in [20] and [21] using RNN for traffic flow prediction have three issues: (1) the last hidden state h_t is used to predict the traffic flow at next time step only, so it is difficult to be applied to multi-step prediction; (2) traffic flow at different time steps may be of different importance for predicting results; (3) time information may significantly change traffic flow patterns, but it's hard to be fed into the network directly. Therefore, we developed an Attention-Based RNN architecture with Temporal Component (ABRNN_TC) for traffic flow prediction (shown in Figure 3). The proposed architecture includes three major components: encoder-decoder architecture, attention mechanism, and temporal component, to address the above issues, respectively.

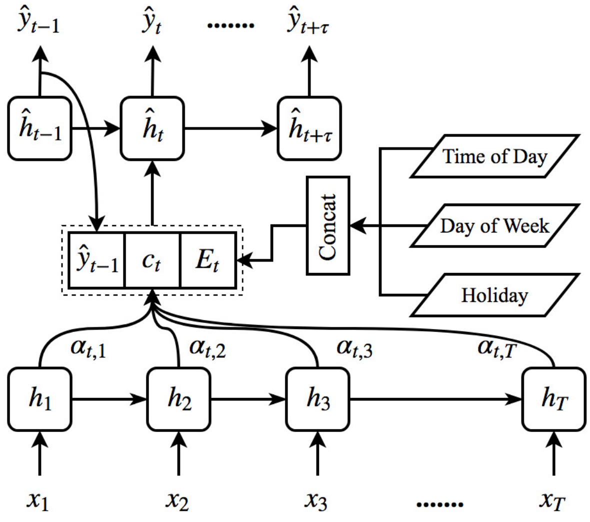


Figure 3. Attention-Based RNN architecture with Temporal Component

The encoder is a LSTM layer that reads an input sequence $X = (x_1, x_2, \ldots, x_T)$ and generates a dynamic context vector *c* from the input sequence for each time step in the decoder, where c_t is computed as a weighted sum of all hidden state of the encoder with Equation (8):

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j \quad \textbf{(8)}$$

the weight α_{tj} of each hidden state h_j is computed by Equation (9) and Equation (10):

$$\alpha_{tj} = \frac{e_{tj}}{\sum_{k=1}^{T} \exp(e_{tk})} \quad \textbf{(9)}$$

$$e_{tj} = a(\hat{h}_{t-1}, h_j) \quad \textbf{(10)}$$

where e_{tj} indicates how well the input at time step *j* and the output at time step *t* match, and *a* is modeled as a feedforward neural network which can be jointly trained with all the other components in the architecture.

Traffic flow has a strong correlation with the temporal factors, e.g., time of the day, day of the week and national holidays. The traffic pattern during weekdays, weekends and national holidays are usually very different. Previous work divided traffic flow into two different groups (e.g., weekdays and weekends) for prediction. In our model, a temporal component is added to handle these factors. As shown in Figure 3, we concatenate the temporal information (denoted as E_t), the time of the day (values from 0 to 95), the day of the week (from 0 to 6) and whether it is national holiday (0 or 1), to the context vector c_t.

Finally, another LSTM layer is used as decoder to generate the output sequence $\hat{Y} = (\hat{y}_{T+1}, \hat{y}_{T+2}, \ldots, \hat{y}_{T+\tau})$. Unlike the LSTM described in Section 3.1.1, the decoder LSTM also includes the context vector *c* and the temporal component output E_t as input while updating the hidden state. Hence, the hidden state of the decoder at time *t* is calculated using Equation (11) below.

$$\hat{h}_t = f(\hat{h}_{t-1}, y_{t-1}, c_t, E_t) \quad \textbf{(11)}$$

where *f* represents a LSTM unit here.

4 Experiments & Evaluation

4.1 Dataset

The Caltrans Performance Measurement System (PeMS) [31] is a widely used dataset for traffic flow prediction [16-20]. The traffic data was collected every 30 seconds from various types of vehicle detector stations throughout the state of California in the United States. Then, it was aggregated at a 5-min interval for each detector station. We further aggregated the data into 15-min interval, as suggested by the Highway Capacity Manual [32]. In this paper, we used the dataset that was collected from 243 vehicle detector stations in district 5 (including Monterey, San Benito, etc.) from May 1, 2017 to Feb 28, 2018. The data of the first nine months was used for training, and the data of the remaining one month was used for testing.

4.2 Experiments

Input and output. For the input layer, we collected data from all 243 vehicle detector stations at previous *r* time steps, i.e., $x_{t-r}, x_{t-r+1}, \ldots, x_t$. The data includes both the relationship of 243 detector stations and temporal correlations. For the output layer, we predicted the traffic flow of the 243 stations at the next τ time steps, i.e. $x_{t+1}, x_{t+2}, \ldots, x_{t+\tau}$. The dimension of the input shape is $243 \times \gamma$; the dimension of the output shape is $243 \times \tau$. As traffic flow volume of different stations may have different scales, the input data was further normalised to the range of [0, 1] by Equation (12).

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad \textbf{(12)}$$

We chose *r* from {2, 4, 8, 12, 24, 48, 96}. After performing grid search, the best number of input time steps was 4, which means the traffic flow prediction mostly depends on the traffic flow of the previous one hour. The prediction performance dropped rapidly when *r* increased due to the difficulty in modelling long historical time sequences. We used the proposed method to predict traffic flow volumes in next 1 hour, 2 hours, and 3 hours, where τ is 4, 8, and 12 respectively. As the traffic flow of weekdays, weekends, and national holiday may have different patterns, we further collected time information as an additional input for the temporal component E_t in the proposed model.

Model parameters. With regard to the attention based RNN architecture, we need to determine the number of hidden layers, the number of hidden units, batch size, epochs, optimiser, and etc. Because training the attention based RNN model is time consuming, the number of hidden layers is set to 1 in this study. We chose the number of hidden units from {128, 256, 512, 1024} and the number of batch size from {64, 128, 256, 512}. We used early stopping to avoid overfitting. After performing grid search, the best model parameters were determined and shown in Table 1. These experiments were run using Keras 2.1.5, Tensorflow 1.3, python 3.6, and Windows 10 on a laptop with a i7-6700HQ CPU, 8GB RAM and GTX-970M GPU.

Table 1. Model parameter settings

LSTM-ED model parameters			
Input length	4	Batch size	256
Output length	{4, 8, 12}	Dropout	0
Hidden layers	1	Epochs	100
Hidden units	512	Optimiser	Adam

4.3 Evaluation

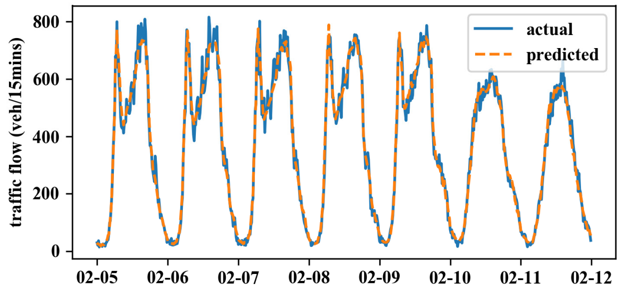
Figure 4 presents the predicted results for three different vehicle detector stations at Freeway 1, Freeway 101 and Freeway 156. The actual traffic flow data is also plotted for comparison. The figure shows that the proposed model is able to learn the traffic flow patterns and provides accurate traffic predictions. To evaluate the effectiveness of the proposed model, we used the root mean square error (RMSE) and mean absolute error (MAE), which are calculated by Equations (13) and (14), respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |^2} \quad \textbf{(13)}$$

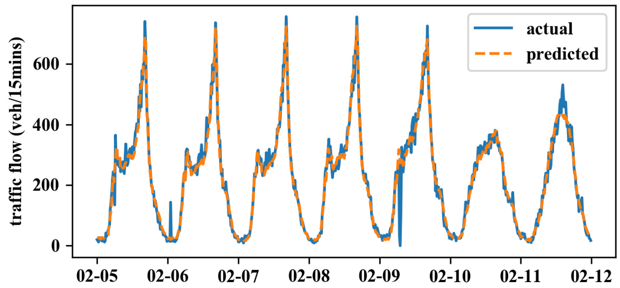
$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \quad \textbf{(14)}$$

where y_i is the observed traffic flow, and $\hat{y}_i$ is the predicted traffic flow.

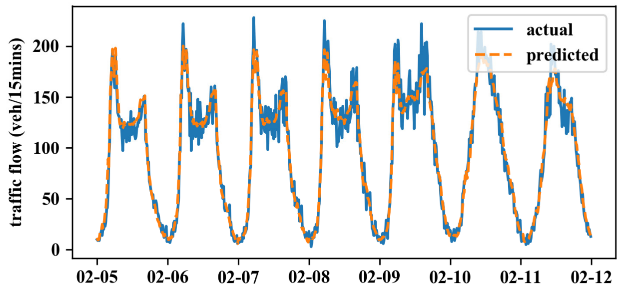
The performance of the proposed model was also compared with four other methods as briefly explained below.



(a) Station at Freeway 1



(b) Station at Freeway 101



(c) Station at Freeway 156

Figure 4. Traffic flow prediction results of three different stations.

- **AVG.** It is a simple method that calculates the average traffic flow of each station at specific time (e.g. 9 AM on Monday).
- **k-NN [23].** It finds the k most similar traffic flow patterns to the current traffic flow and predicts future time steps on the basis of the averaged future data of the k patterns.
- **Seq2Seq [29].** It uses a standard RNN with LSTM units to encoder the input sequence into a context vector and another RNN to make predictions iteratively.
- **Attention-based RNN [13].** It introduces an attention mechanism to adaptively select the weight of hidden states from the encoder to produce the output sequence.

Table 2. Performance comparison of the RMSE and MAE for differnet models

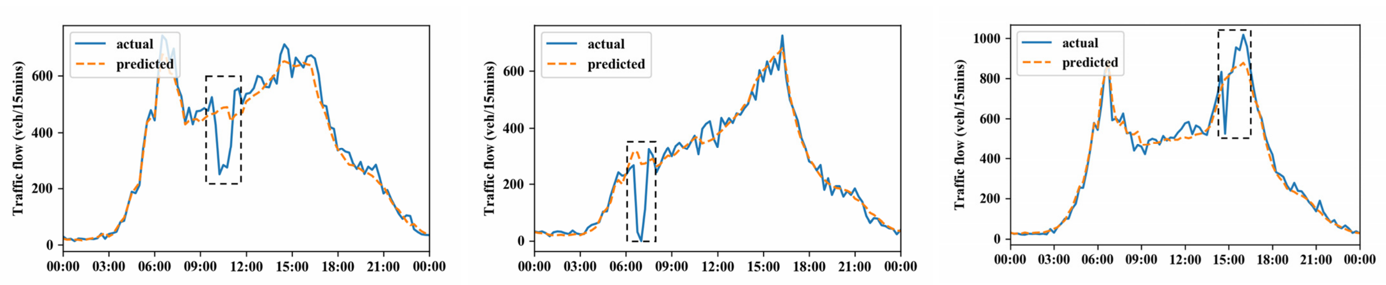
Method	1-hour traffic flow prediction		2-hours traffic flow prediction		3-hours traffic flow prediction	
	RMSE (10^{-2})	MAE (10^{-2})	RMSE (10^{-2})	MAE (10^{-2})	RMSE (10^{-2})	MAE (10^{-2})
AVG	6.86	4.42	6.86	4.42	6.86	4.42
k-NN	4.95	3.26	4.86	3.15	4.89	3.14
Seq2Seq	3.94	2.78	4.34	3.03	4.57	3.16
ABRNN	3.92	2.74	4.29	2.99	4.53	3.15
ABRNN_TC	**3.89**	**2.71**	**4.26**	**2.96**	**4.48**	**3.08**

The best results of each method under different parameters were reported in Table 2. As expected, simply using the statistical AVG method led to inaccurate results. Following the idea of [23], using k-NN provided better and robust traffic prediction results. In general, the deep learning models, i.e., Seq2Seq and ABRNN, outperformed other methods, which has also confirmed by other previous studies [16-21]. Compared to the standard Seq2Seq model, ABRNN models further improved the prediction accuracy due to the positive effects of the attention mechanism. Since there is not long temporal dependency in traffic flow data, only limited improvement could be observed with the addition of the attention mechanism. We expect attention-based mechanisms will be able to provide more notable improvement when applying to data with longer temporal correlations, e.g., air quality, water quality, etc. When adding the temporal information to the model, the MAE of attention based RNN model further decreased by around 2.2%. It should be noted that we only use the temporal information as the external data in this study. It is also possible to include more external factors that are relevant to traffic flow for further improvement, e.g., traffic speed, weather condition, accidents, and so on.

4.4 Use Case: Traffic Anomaly Detection

Traffic flow data follows a more or less recurring pattern. Meanwhile, it may vary abnormally due to various traffic events, road conditions, and other external factors. In these cases, the predicted traffic flow and the actual traffic flow may have significant differences. One interesting application is to analyse traffic anomalies to detect real-time traffic incidents or events. We demonstrate a use case that applies the proposed method together with social media information to detect traffic incidents and traffic events.

Since the proposed method is able to capture traffic flow patterns, an unusually high prediction error is likely to indicate a real-world traffic anomaly. For example, in Figure 5 it shows that there are three traffic anomalies on highway 101 in February 2018. Although a threshold can be set to extract the temporal and spatial information of traffic anomaly, details (i.e., event type, cause and impact) of the traffic event is still missing.



(a) (b) (c)

Figure 5. Three Traffic events on Highway 101 detected from sensor data

While social media data published by the citizen sensors differs semantically from sensing data generated by the physical devices to a great extent, it can be used as an important complementary source for traffic related applications. We further used the temporal and spatial information collected from the detected traffic anomalies to filter traffic related tweets posted by trusted organisations on Twitter. In Table 3, details of the traffic accidents, e.g., overturned truck or car crash, can be collected from tweets published by CaltransD5.

Table 3. Traffic event description from sensor and social meida data

	Time	Location	Event
(a)	2018-02-02 9: 00 - 12: 00	Highway 101 off-ramp at Traffic Way in Atascadero	Overturned truck
(b)	2018-02-09 6: 00 - 8: 00	Highway 101 near Highway 58	Overturned semi-truck
(c)	2018-02-13 14: 00 - 17: 00	Highway 101 on Santa Maria Bridge	2-car crash

This is a simple traffic anomaly detection example based on historical traffic data and official tweets. A real-time traffic anomaly detection system or more generally a physical and social data analysis system can be developed as illustrated in Figure 6. For a traffic event detection system, we can set a threshold for a particular case to be considered as traffic anomaly (e.g., difference between the actual and predicted values is 3 times of the standard deviation). Meanwhile, we can use openly available city traffic event tool [33] to extract traffic event from twitter. Then, the anomalies can be explained by the social media data with temporal (within 1 hour) and spatial (within *r* km) overlapping with the detected event. With the development of IoT technologies and data analysis methods, the physical and social data may become more accessible and further applied to various city domains, e.g. transportation, energy, environment, public safety, etc.

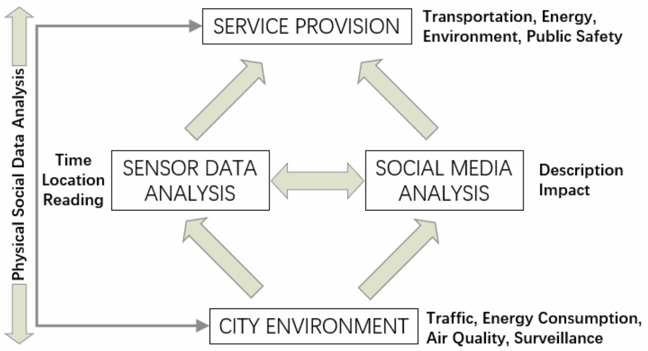


Figure 6. A physical and social data analysis framework

5 Conclusion and Future Work

We propose an Attention Based Recurrent Neural Network with Temporal Component (ABRNN_TC) model for predicting future traffic flow. The experimental results show that with the addition of the attention mechanism and the temporal component, the deep model can capture traffic patterns accurately and produce superior prediction results over other baseline methods. We also presented a use case of the proposed method for traffic anomaly detection with information found on the social media websites. For future work, we plan to incorporate more factors relevant to traffic flow into the proposed method and further evaluate their impact on the prediction accuracy. Another future work is to design a new deep learning model that can simultaneously process sensor data and social media data for traffic flow prediction and anomaly detection.

Acknowledgements

The authors wish to thank for the continuous support from AI University Research Centre (AI-URC) and Research Institute of Big Data Analytics (RIBDA), Xi’an Jiaotong-Liverpool University, China. The research presented in this paper is supported by XJTLU Key Programme Special Fund (#KSF-P-02) and RIBDA Internal Research Grant.

References

[1] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A Survey, *Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, October, 2010.

[2] G. Pan, G. Qi, W. Zhang, S. Li, Z. Wu, L. T. Yang, Trace Analysis and Mining for Smart Cities: Issues, Methods, and Applications, *IEEE Communications Magazine*, Vol. 51, No. 6, pp. 120-126, June, 2013.

[3] U. Varshney, Pervasive Healthcare and Wireless Health Monitoring, *Mobile Networks and Applications*, Vol. 12, No. 2-3, pp. 113-127, June, 2007.

[4] B. T. Ong, K. Sugiura, K. Zettsu, Dynamically Pre-trained Deep Recurrent Neural Networks Using Environmental Monitoring Data for Predicting PM2.5, *Neural Computing and Applications*, Vol. 27, No. 6, pp. 1553-1566, June, 2016.

[5] Y. Tang, C. Zhang, R. Gu, P. Li, B. Yang, Vehicle Detection and Recognition for Intelligent Traffic Surveillance System, *Multimedia Tools and Applications*, Vol. 76, No. 4, pp. 5817-5832, March, 2017.

[6] N. Zhang, F.-Y. Wang, F. Zhu, D. Zhao, S. Tang, DynaCAS: Computational Experiments and Decision Support for ITS, *IEEE Intelligent Systems*, Vol. 23, No. 6, November, 2008.

[7] J. Zhang, D. He, N. Kumar, K.-K. R. Choo, An Efficient and Secure Authentication Scheme for Vehicle Sensor Networks, *Journal of Internet Technology*, Vol. 20, No. 2, pp. 617-627, March, 2019.

[8] R. Li, H. Asaeda, MWBS: An Efficient Many-to-Many Wireless Big Data Delivery Scheme. *IEEE Transactions on Big Data*, October, 2018, DOI: 10.1109/TBDATA.2018.2878584.

[9] R. Li, H. Asaeda, J. Wu, DCAuth: Data-Centric Authentication for Secure In-Network Big-Data Retrieval, *IEEE Transactions on Network Science and Engineering*, September, 2018, DOI: 10.1109/TNSE.2018.2872049.

[10] R. Li, H. Asaeda, J. Li, A Distributed Publisher-Driven Secure Data Sharing Scheme for Information-Centric IoT, *IEEE Internet of Things Journal*, Vol. 4, No. 3, pp. 791-803, June, 2017.

[11] R. Li, H. Harai, H. Asaeda, An Aggregatable Name-Based Routing for Energy-Efficient Data Sharing in Big Data Era, *IEEE Access*, Vol. 3, pp. 955-966, July, 2015.

[12] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, *Nature*, Vol. 521, No. 7553, pp. 436-444, May, 2015.

[13] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, *International Conference on Learning Representations*, San Diego, California, USA, 2014.

[14] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classication with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, 2012, pp. 1097-1105.

[15] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, B. Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp. 82-97, November, 2012.

[16] W. Huang, G. Song, H. Hong, 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, October, 2014.

[17] A. B. Koesdwiady, R. Soua, F. Karray, Improving Traffic Flow Prediction with Weather Information in Connected Cars: A Deep Learning Approach, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 12, pp. 9508-9517, December, 2016.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, 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.

[19] H. F. Yang, T. S. Dillon, Y. P. Chen, Optimized Structure of the Traffic Flow Forecasting Model with a Deep Learning Approach, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 10, pp. 2371-2381, October, 2017.

[20] Y. Tian, L. Pan, Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network, *IEEE International Conference on Smart City/SocialCom/SustainCom*, Chengdu, China, 2015, pp. 153-156.

[21] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long Short-Term Memory Neural Network for Traffic Speed Prediction using Remote Microwave Sensor Data, *Transportation Research Part C: Emerging Technologies*, Vol. 54, pp. 187-197, May, 2015.

[22] M. S. Ahmed, A. R. Cook, Analysis of Freeway Traffic Time Series Data by Using Box-Jenkins Techniques, *Transportation Research Record*, Vol. 773, No. 722, pp. 1-9, January, 1979.

[23] H. Chang, Y. Lee, B. Yoon, S. Baek, Dynamic Near-Term Traffic Flow Prediction: System Oriented Approach Based on Past Experiences, *IET Intelligent Transportation Systems*, Vol. 6, No. 3, pp. 292-305, September, 2012.

[24] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, S. M. Easa, Supervised Weighting-Online Learning Algorithm for Short-Term Traffic Flow Prediction, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 4, pp. 1700-1707, December, 2013.

[25] J. Ding, Investigation on the Traffic Flow Based on Wireless Sensor Network Technologies Combined with FA-BPNN Models, *Journal of Internet Technology*, Vol. 20, No. 2, pp. 589-597, March, 2019.

[26] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning Representations by Back-Propagating Errors, *Nature*, Vol. 323, No. 6088, pp. 399-421, October, 1986.

[27] G. E. Hinton, S. Osindero, Y.-W. Teh, A Fast Learning Algorithm for Deep Belief Nets, *Neural Computation*, Vol. 18, No. 7, pp. 1527-1554, August, 2006.

[28] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, December, 1997.

[29] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724-1734.

[30] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3104-3112.

[31] Caltrans, Performance Measurement System (PeMS), http://pems.dot.ca.gov.

[32] Highway Capacity Manual, Transportation Research Board, National Research Council, Washington, D.C., 2000.

[33] P. Anantharam, P. Barnaghi, K. Thirunarayan, A. Sheth, Extracting City Traffic Events from Social Streams, *ACM Transactions on Intelligent Systems and Technology*, Vol. 6, No. 4, pp. 1-27, July, 2015.

Biographies

Qi Chen received his M.S. degree in computer science from Stevens Institute of Technology, USA, in 2017. He is currently working toward the Ph.D. degree in computer science from University of Liverpool, UK. His research interests include deep learning, smart city, and data mining.

Wei Wang received his Ph.D. in Computer Science from University of Nottingham, in 2009, and is currently an associate professor at the Department of Computer Science and Software Engineering, Xi'an Jiaotong Liverpool University, China. His research interests lie in the broad area of data and knowledge engineering.

Xin Huang received his Ph.D. degree from University of Oxford. He is currently a professor with Taiyuan University of Technology, China; he is also working as a post-doctoral researcher with Soochow University and Suzhou L. J. T Intelligent Technology Co., Ltd., Suzhou, China. His research interests include information security, Internet of Things, and blockchain.

Hai-Ning Liang is a Senior Associate Professor with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, China. He completed his Ph.D. in computer science from Western University, Canada. His research lies primarily in the area of human-computer interaction with a focus on VR/AR and other interactive technologies.