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

Travel Time Prediction with LSTM Neural Network

Yanjie Duan[1,2,3], Yisheng Lv[1] and Fei-Yue Wang[1,2]

Abstract— **Travel time is one of the key concerns among travelers before starting a trip and also an important indicator of traffic conditions. However, travel time acquisition is time delayed and the pattern of travel time is usually irregular. In this paper, we explore a deep learning model, the LSTM neural network model, for travel time prediction. By employing the travel time data provided by Highways England, we construct 66 series prediction LSTM neural networks for the 66 links in the data set. Through model training and validation, we obtain the optimal structure within the setting range for each link. Then we predict multi-step ahead travel times for each link on the test set. Evaluation results show that the 1-step ahead travel time prediction error is relatively small, the median of mean relative error for the 66 links in the experiments is 7.0% on the test set. Deep learning models considering sequence relation are promising in traffic series data prediction.**

I. Introduction

Travel time is an important indicator of traffic capacity and traffic efficiency. As a kind of dynamic traffic information, real-time travel time is one of the key concerns among travelers. With real-time travel time, the intelligent transportation system (ITS) [1] can offer information services to decision makers for traffic control or guidance operations and to travelers for route selection and planning. However, real-time travel time cannot be real-time observed, since travel time counts from the start to the end of a trip. When travel time is observed, it has already been historical data rather than real-time data. Therefore travel time prediction is an effective way to obtain the real-time travel time before starting a trip.

There have been many researchers studying travel time prediction using various methods. These methods are mainly classified into two categories: data-driven methods and model-driven methods. Data-driven methods usually employ historical travel time [2], and other related variables e.g. speed, volume, occupancy [3], time of day, day of week etc. [4]. Among the models and algorithms applied in data-driven methods, the ARIMA model [5] uses the historical series of travel time to fit a time series model and then predicts the future travel time one by one. The linear model [6], [7] predicts the travel time of one trip departing at current time by combining the latest calculated travel time of the trip and the historical mean travel time of the same trip departing at the same time. The k nearest neighbors method [7], [8] finds the most similar historical k days to the present day, and takes the mean travel time at current time of that k days as the travel time at the current time of the present day. The Kalman filtering algorithm [9] estimates the predicted travel time and updates the prediction continually as a new observation becomes available. The support vector regression model [10] maps the historical travel time into a higher dimensional feature space from the lower input space and fits a function predicting the future travel time from the higher dimensional features. The gradient boosting regression tree method [4] combines simple regression trees to produce high accuracy travel time prediction with previous multiple time periods travel times and related variables as model inputs. Apart from the above models and algorithms, neural network model is applied to travel time prediction in various forms and structures. The spectral basis neural networks [11] perform a sinusoidal transformation to the input features including travel times of current, upstream and downstream links in previous time periods, obtain the spectral expansion of the input features, then utilize the conventional artificial neural network to predict the future multiple time periods travel times of the current link from the spectral expansion. The objected-oriented neural network approach [12] uses the current observed speed and flow data of the upstream and downstream stations on the freeway section as the network input, and predicts the future travel time on that section at the network output. The state-space neural network (SSNN) [13] with a topology of recurrent neural network (RNN) connects each hidden unit with the traffic flow and average speed data collected from one of the detectors on the concerned section and the hidden layer in previous time period. Then the SSNN predicts the future travel time in its output layer. The RNN considering temporal-spatial input dynamics [3] takes speed, volume and occupancy of current, upstream and downstream segments in the present time period as network input and predicts the travel time in the next time period at the network output. Model-driven methods typically need to build a virtual road network and perform simulation on it with the principle of dynamic traffic assignment. The travel time of one link or path can be detected from the virtual road network through the simulation. DynaSmart [14], DynaMIT [15], Vissim [16], Paramics [17], TransWorld [18] are transportation softwares for building virtual road networks and conducting traffic simulations, all of which need the travel demand data named origin-destination matrix or population data within the road network. However, these

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

[1]The State Key Laboratory of Management and Control for Complex Systems Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

[2]Qingdao Academy of Intelligent Industries, Qingdao, Shandong, 266109, China

[3]University of Chinese Academy of Sciences, Beijing, 100049, China

**Yisheng Lv is the corresponding author of this paper. E-mail: yisheng.lv@ia.ac.cn

travel demand data are difficult to acquire.

Data-driven methods [19], [20] have attracted more and more research interest and achieved inspiring results with the improvement of computing capability and the growth of traffic data quantity. Among them, the deep learning based methods lead the trend of big data processing. Exploring deep learning models in the application to travel time prediction is of great significance. In this paper, we explore the Long Short-Term Memory (LSTM) neural network model, which can automatically reserve historical sequence information in its model structure, for travel time prediction. The model has a deep structure in term of time but is with low dimension in term of single step prediction. To the best of our knowledge, it is the first time that the LSTM neural network is used for travel time prediction.

The rest of this paper is organized as follows. Section II introduces travel time and travel time acquisition approaches. Section III describes the LSTM Neural Network for travel time prediction. Section IV presents the experiments conducted on the LSTM neural network using real travel time data, analyses the experiments results and discusses the model performance. Section V concludes this paper.

II. Travel time

Travel time x_t shown in Fig. 1 is the time cost to complete a journey L from the starting point A to the end point B departing at time t. In this paper, we focus on the travel time of each link on highways. Travel time cannot be detected directly. There are multiple ways to acquire travel time. One way is to calculate travel time using the detected or estimated occupancy and speed, e.g. the PCSB method and the PLSB method [21]. The other way is to estimate travel time using Global Positioning Systems (GPS) data of vehicles [22]. Another way is to measure travel time from Automatic Vehicle Identification (AVI) stations [11] at ends of a journey.

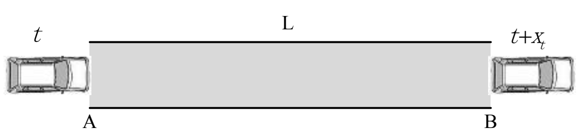

Fig. 1. Travel time

In this paper, we employ the travel time data provided by Highways England [23]. In this dataset, journey(travel) times are estimated using a combination of sources, including Automatic Number Plate Recognition (ANPR) cameras, in-vehicle GPS and inductive loops built into the road surface, and imputed using adjacent time periods or the same time period on different days. Specifically, the travel times in this data set are 15-minute interval average travel times for each link. We utilize the travel time data in Year 2013 of 66 links on highway M25, which almost encircles Greater London, England, in the United Kingdom. We focus on exploring the LSTM neural network model for link travel time prediction in the following sections.

III. Methodology

LSTM neural network [24] has been successfully applied in many real-world problems [25] involving sequence data, e.g. music generation [26], image captioning [27], speech recognition [28], machine translation [29]. In this paper, we explore the application of LSTM neural network in travel time prediction. LSTM neural network is closely connected with and can be seen as a specific RNN [30], which is proposed before LSTM neural network and also considers the sequence relation in data samples. Therefore we firstly introduce the structure of RNN, then describe the LSTM cell and the construction of the LSTM neural network for travel time prediction. Here travel time prediction is defined as predicting future time periods travel times $\{\widetilde{x}_{t+1}, \widetilde{x}_{t+2}, \cdots\}$ from the acquired historical travel times $\{\cdots, x_{t-1}, x_t\}$.

A. Recurrent Neural Network

RNN incorporates temporal dynamics in its structure shown in Fig. 2, which is a basic structure of regular RNN [31]. The hidden layer of each neural network is connected with the hidden layer of the next neural network later in time sequence. This connection style takes the influence of the former sample on the latter one into account. There is another connection style considering the relation between the former and the latter samples by connecting the hidden layer of each neural network with the hidden layers of its former and latter neural networks. RNN with this bidirectional connection style is called bidirectional recurrent neural network [32]. In this paper, travel time prediction adopts the structure of regular unidirectional RNN, since the prediction from history to future is what concerns us. Typically, the neural networks in RNN shown in Fig. 2 have the same parameters such as W_1, W_2 which are the weight matrices between layers and W_h which is the weight matrix between neural networks. Each historical sample is fed into RNN in order and the predicted future output will be obtained in sequence.

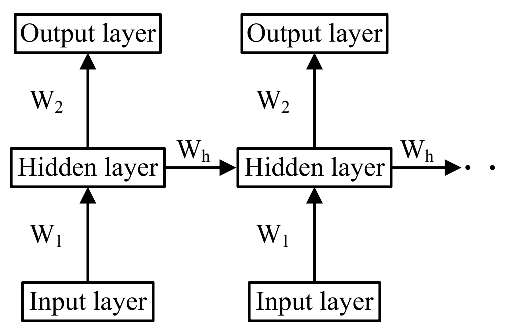

Fig. 2. The structure of RNN

B. Long Short-Term Memory

LSTM neural network as a specific RNN has a complex structure named LSTM cell in its hidden layer. The LSTM cell shown in Fig. 3 has three gates namely input gate, forget gate and output gate, which control the information flow through the cell and the neural network. At time t the input is x_t, the hidden layer output is h_t and its former output is h_{t-1}, the cell input state is $\widetilde{C}_t$, the cell output state is C_t and its former state is C_{t-1}, the three gates' states are i_t, f_t and o_t. The structure of the LSTM cell indicates that both C_t and h_t are transmitted to the next neural network in RNN. To calculate C_t and h_t, we use the following equations in order. Firstly, calculate the three gates' states and the cell input state, input gate:

$$i_t = \sigma(W_1^i \cdot x_t + W_h^i \cdot h_{t-1} + b_i), \tag{1}$$

forget gate:

$$f_t = \sigma(W_1^f \cdot x_t + W_h^f \cdot h_{t-1} + b_f), \tag{2}$$

output gate:

$$o_t = \sigma(W_1^o \cdot x_t + W_h^o \cdot h_{t-1} + b_o), \tag{3}$$

cell input:

$$\widetilde{C}_t = tanh(W_1^C \cdot x_t + W_h^C \cdot h_{t-1} + b_C), \tag{4}$$

where W_1^i, W_1^f, W_1^o, W_1^C are the weight matrices connecting x_t to the three gates and the cell input, W_h^i, W_h^f, W_h^o, W_h^C are the weight matrices connecting h_{t-1} to the three gates and the cell input, b_i, b_f, b_o, b_C are the bias terms of the three gates and the cell input, σ represents the sigmoid function $\frac{1}{1+exp(-x)}$ and $tanh$ represents the hyperbolic tangent function $\frac{exp(x)-exp(-x)}{exp(x)+exp(-x)}$. Secondly, calculate the cell output state:

$$C_t = i_t * \widetilde{C}_t + f_t * C_{t-1}, \tag{5}$$

where i_t, f_t, $\widetilde{C}_t$, C_{t-1} and C_t have the same dimension. Thirdly, calculate the hidden layer output:

$$h_t = o_t * tanh(C_t). \tag{6}$$

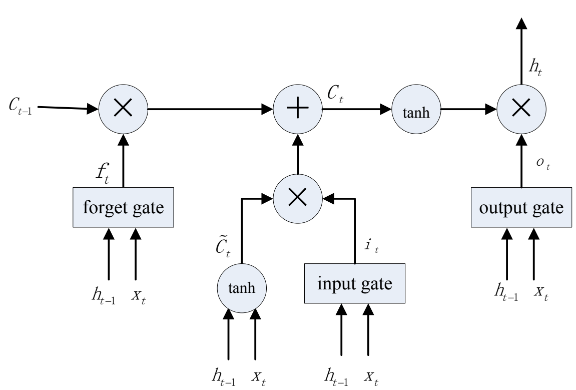

Fig. 3. The structure of LSTM cell

C. Series Prediction LSTM Neural Network

For the purpose of travel time prediction, we construct a LSTM neural network shown in Fig. 4 using RNN and LSTM cell. At time t, the input of the network is the observed history data x_t and the output is the predicted future data $\widetilde{x}_{t+1}$. Through the above LSTM calculation, h_t is obtained. Then calculate the network output:

$$\widetilde{x}_{t+1} = W_2 \cdot h_t + b \tag{7}$$

where W_2 is the weight matrix between the output layer and the hidden layer, b is the bias term of the output layer. In real applications, we can only use N historical data to feed the series prediction LSTM neural network shown in Fig. 4 since historical data are limited. In this way, the history information goes through the networks by recurrent calculation and the prediction is to absorb the long-short term memory from the network states.

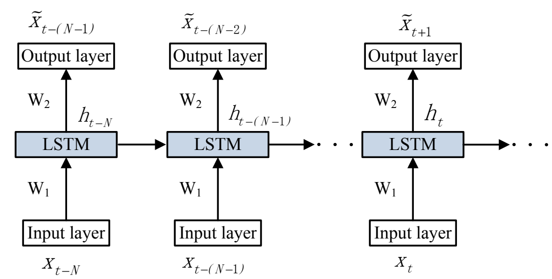

Fig. 4. The structure of series prediction LSTM neural network

IV. Experiments

We construct a series prediction LSTM neural network for each link in the travel time data set. The dimension of the network input and output at each time step equals the dimension of the travel time in each time period. Thus both the input and output dimensions equal 1. As for the dimension of the hidden layer at each step, we conduct experiments to determine the number of hidden units n_h ranging from 1 to 5 given the input dimension. Considering the time delay in travel time collection, we predict multiple time periods travel times ahead the latest observed one with the trained model to see how well the model performs. Here we predict 1,2,3,4 time periods travel times: $\{\widetilde{x}_{t+1}, \widetilde{x}_{t+2}, \widetilde{x}_{t+3}, \widetilde{x}_{t+4}\}$ with observed data $\{\cdots, x_{t-1}, x_t\}$. Thus the predicted travel time is to feed the model input when the observed one has not been acquired.

A. Model Training

Model training is the process of determining the parameters in the model structure. In the series prediction LSTM neural network model, the dimension of the hidden layer n_h and the number of iterations are hyper parameters which must be determined separately from the training process of the other parameters Θ, which includes all the weight matrices and biases in the model. We divide the whole data set into three parts: training set (80%), validation set (10%) and test set (10%). We adjust Θ relying on the training set, choose the hyper parameters by calculating the cost function on the validation set, and evaluate the prediction performance on the test set. Adjusting Θ is the process of solving the following optimization problem:

$$\Theta = \arg\min_{\Theta} L(X, \widetilde{X})$$

where X is the observed travel time sequence, $\widetilde{X}$ is the corresponding 1-step ahead predicted travel time sequence and L is the cost function. Define

$$L(X,\widetilde{X}) = \frac{\sum^{Num}(x_{t+1} - \widetilde{x}_{t+1})^2}{2Num}, \tag{8}$$

where Num is the number of predicted travel times. Then the gradient of L with respect to the weights are calculated by backward propagation through time using the following equations:

$$\frac{\partial L}{\partial \widetilde{x}_{t+1}} = -\frac{x_{t+1} - \widetilde{x}_{t+1}}{Num} \tag{9}$$

$$\frac{\partial L}{\partial W_2} = \sum^{Num} \frac{\partial L}{\partial \widetilde{x}_{t+1}} \cdot h_t^{\mathrm{T}} \tag{10}$$

$$\begin{aligned}\frac{\partial L}{\partial h_t} =& \frac{\partial L}{\partial \widetilde{x}_{t+1}} \cdot W_2^{\mathrm{T}} + \frac{\partial L}{\partial i_{t+1}} * i_{t+1} * (1 - i_{t+1}) \cdot (W_h^i)^{\mathrm{T}} \\ &+ \frac{\partial L}{\partial f_{t+1}} * f_{t+1} * (1 - f_{t+1}) \cdot (W_h^f)^{\mathrm{T}} \\ &+ \frac{\partial L}{\partial o_{t+1}} * o_{t+1} * (1 - o_{t+1}) \cdot (W_h^o)^{\mathrm{T}} \\ &+ \frac{\partial L}{\partial \widetilde{C}_{t+1}} * (1 - \widetilde{C}_{t+1}^2) \cdot (W_h^C)^{\mathrm{T}}\end{aligned} \tag{11}$$

$$\frac{\partial L}{\partial C_t} = \frac{\partial L}{\partial h_t} * o_t * [1 - tanh^2(C_t)] + \frac{\partial L}{\partial C_{t+1}} * f_{t+1} \tag{12}$$

$$\frac{\partial L}{\partial o_t} = \frac{\partial L}{\partial h_t} * tanh(C_t) \tag{13}$$

$$\frac{\partial L}{\partial f_t} = \frac{\partial L}{\partial C_t} * C_{t-1} \tag{14}$$

$$\frac{\partial L}{\partial i_t} = \frac{\partial L}{\partial C_t} * \widetilde{C}_t \tag{15}$$

$$\frac{\partial L}{\partial \widetilde{C}_t} = \frac{\partial L}{\partial C_t} * i_t \tag{16}$$

$$\frac{\partial L}{\partial W_1^C} = \sum^{Num} \frac{\partial L}{\partial \widetilde{C}_t} * (1 - \widetilde{C}_t^2) \cdot x_t^{\mathrm{T}} \tag{17}$$

$$\frac{\partial L}{\partial W_1^o} = \sum^{Num} \frac{\partial L}{\partial o_t} * o_t * (1 - o_t) \cdot x_t^{\mathrm{T}} \tag{18}$$

$$\frac{\partial L}{\partial W_1^f} = \sum^{Num} \frac{\partial L}{\partial f_t} * f_t * (1 - f_t) \cdot x_t^{\mathrm{T}} \tag{19}$$

$$\frac{\partial L}{\partial W_1^i} = \sum^{Num} \frac{\partial L}{\partial i_t} * i_t * (1 - i_t) \cdot x_t^{\mathrm{T}} \tag{20}$$

$$\frac{\partial L}{\partial W_h^C} = \sum^{Num} \frac{\partial L}{\partial \widetilde{C}_t} * (1 - \widetilde{C}_t^2) \cdot h_{t-1}^{\mathrm{T}} \tag{21}$$

$$\frac{\partial L}{\partial W_h^o} = \sum^{Num} \frac{\partial L}{\partial o_t} * o_t * (1 - o_t) \cdot h_{t-1}^{\mathrm{T}} \tag{22}$$

$$\frac{\partial L}{\partial W_h^f} = \sum^{Num} \frac{\partial L}{\partial f_t} * f_t * (1 - f_t) \cdot h_{t-1}^{\mathrm{T}} \tag{23}$$

$$\frac{\partial L}{\partial W_h^i} = \sum^{Num} \frac{\partial L}{\partial i_t} * i_t * (1 - i_t) \cdot h_{t-1}^{\mathrm{T}} \tag{24}$$

where the critical steps are equation (11) and (12) considering sequence relation. The gradient of L with respect to the bias terms are calculated similarly. Choosing hyper parameters is embedded in the training iterations of the LSTM neural network. The training and choosing process is summarized in Algorithm 1. After training we obtain the optimal n_h within the range,

$$n_h = \arg\min_{n_h} L_{best-val}^{n_h},$$

and the corresponding saved parameters Θ.

Algorithm 1 Training series prediction LSTM neural network

Require: training set $X_{training} = \{(x_t, x_{t+1}), t = 1, 2, \cdots, T_1\}$ and validation set $X_{validation} = \{(x_t, x_t + 1), t = T_1 + 1, T_1 + 2, \cdots, T_2\}$, the range of n_h: $1 \sim 5$, the $max - epoch$: 5000 and the $min - epoch$: 100.

for n_h = 1 to 5 **do**
  Initialization: initialize Θ randomly, $L_{best-val}^{n_h} = +\infty$
  Adjusting Θ**:**
  for $epoch$ = 1 to $max - epoch$ **do**
    Perform forward propagation recurrently using equation (1)-(7) to compute $\widetilde{x}_{t+1}, t = 1, 2, \cdots, T_1$
    Compute output error: $\widetilde{x}_{t+1} - x_{t+1}, t = 1, 2, \cdots, T_1$
    Perform backward propagation through time using equation (9)-(24) to compute $\Delta\Theta$
    Update Θ: $\Theta = \Theta + \Delta\Theta$
    Perform forward propagation recurrently to update the network states using equation (1)-(6)
    Perform forward propagation recurrently to compute $\widetilde{X} = \{\widetilde{x}_{t+1}, t = T_1 + 1, T_1 + 2, \cdots, T_2\}$
    Calculate the cost function on validation set $L_{this-val}$ using equation (8)
    if $L_{this-val} < L_{best-val}^{n_h}$ **then**
      if $L_{this-val} < L_{best-val} \times 0.995$ **then**
        $min - epoch = max(epoch \times 2, min - epoch)$
      end if
      $L_{best-val}^{n_h} = L_{this-val}$
      Save the current Θ and $L_{best-val}^{n_h}$
    end if
    if $epoch \geq min - epoch$ **then**
      break
    end if
  end for
end for

B. Model Evaluation

On the test set, we perform forward propagation recurrently using equation (1)-(7) to compute $\widetilde{x}_{t+1}, t = T_2 + 1, T_2 + 2, \cdots, T_{end}$. In order to evaluate the performance of the LSTM neural network for travel time prediction, we adopt three criteria to measure the prediction error. They are mean absolute error (MAE),

$$MAE = \frac{\sum^{Num} |x_{t+1} - \widetilde{x}_{t+1}|}{Num},$$

root mean square error (RMSE),

$$RMSE = \sqrt{\frac{\sum^{Num} (x_{t+1} - \widetilde{x}_{t+1})^2}{Num}},$$

and mean relative error (MRE),

$$MRE = \frac{\sum^{Num} \frac{|x_{t+1} - \widetilde{x}_{t+1}|}{x_{t+1}}}{Num},$$

where Num is the total number of test samples. Additionally, we predict $\widetilde{x}_{t+2}, \widetilde{x}_{t+3}, \widetilde{x}_{t+4}$ on the test set.

C. Experiments Results and Discussions

In the model training part, we obtain the optimal n_h within the set range and the corresponding saved parameters Θ for each link in travel time prediction. We show the distribution of the optimal n_h in Table I. The result in Table I indicates that the structure of the LSTM neural network varies with different links. Therefore model complexity of the LSTM neural network must adapt to different travel time patterns collected from different links.

TABLE I
THE DISTRIBUTION OF THE OPTIMAL n_h

The optimal n_h	1	2	3	4	5
The number of links	16	14	17	9	10

In the model evaluation part, we get the predicted multiple time periods travel times: 1-step ahead, 2-step ahead, 3-step ahead and 4-step ahead for each link. To see the prediction performance of the LSTM neural network, we present the MAE, RMSE, MRE results in Fig. 5-Fig. 7 and the median and 95%th values of them in Table II. Obviously, 1-step ahead predictions have relatively minor errors while the errors of multi-step predictions grow with the number of steps. The MRE of 4-step ahead predictions can even exceed 1 for two links illustrated in Fig. 7. Therefore timely historical travel time acquisition is important in obtaining accurate prediction of future travel time.

V. CONCLUSION

We explore a deep learning model, the LSTM neural network model, for travel time prediction. Employing the travel time data provided by Highways England, we construct 66 series prediction LSTM neural networks for the 66 links in the data set. We obtain the optimal structure within the setting range for each link after model training and validation. We predict multi-step ahead travel times for each link on the test set using the trained model. Evaluation results show that the 1-step ahead travel time prediction error is relatively small, the median of MRE for the 66 links is 7.0% on the test set. Deep learning models considering sequence relation are promising in traffic series data prediction. The LSTM neural network model for travel time prediction is a

TABLE II
MODEL PERFORMANCE STATISTICS

Step ahead	Median			95%th		
	MAE	RMSE	MRE	MAE	RMSE	MRE
1	16.9s	43.0s	0.070	60.4s	118s	0.173
2	23.1s	54.9s	0.092	89.1s	171s	0.298
3	27.8s	61.1s	0.113	136s	273s	0.446
4	27.8s	66.2s	0.108	193s	388s	0.770

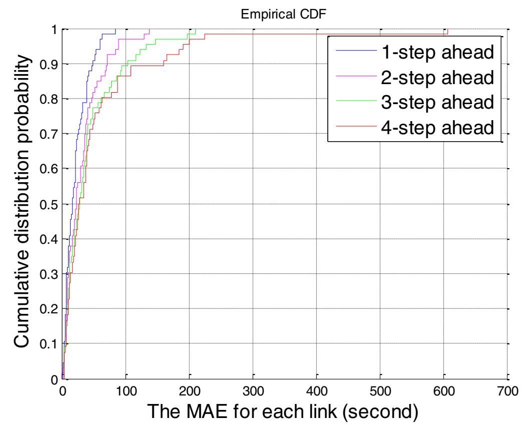

Fig. 5. The distribution of MAE for each link

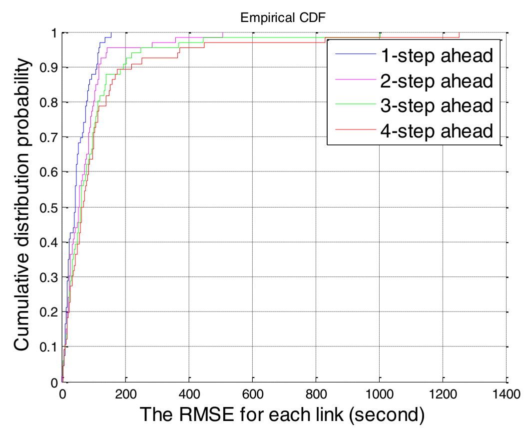

Fig. 6. The distribution of RMSE for each link

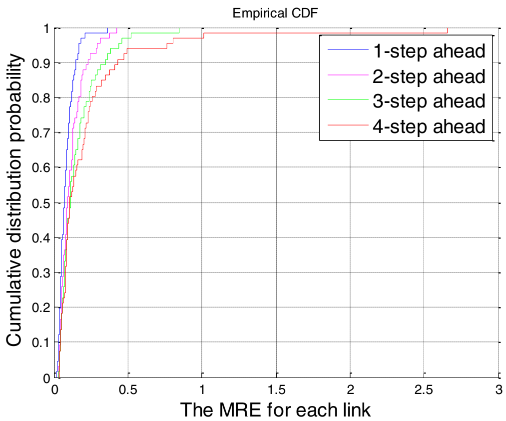

Fig. 7. The distribution of MRE for each link

good example. For future work, we will go on the research to improve prediction accuracy. One future direction is to apply ensemble methods including ensemble models and ensemble data sources to address bursts in travel time. Furthermore, with the rapid development of deep learning, we will try to apply more advanced models to solve prediction problems in transportation.

REFERENCES

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

[2] Y. J. Zou, X. X. Zhu, Y. L. Zhang, and X. S. Zeng, "A space-time diurnal method for short-term freeway travel time prediction," *Transportation Research Part C-Emerging Technologies*, vol. 43, pp. 33–49, 2014.

[3] X. S. Zeng and Y. L. Zhang, "Development of recurrent neural network considering temporal-spatial input dynamics for freeway travel time modeling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 5, pp. 359–371, 2013.

[4] Y. R. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C-Emerging Technologies*, vol. 58, pp. 308–324, 2015.

[5] D. Billings and J.-S. Yang, "Application of the arima models to urban roadway travel time prediction-a case study," in *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, vol. 3, pp. 2529–2534, IEEE.

[6] X. Y. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C-Emerging Technologies*, vol. 11, no. 3-4, pp. 187–210, 2003.

[7] J. Rice and E. van Zwet, "A simple and effective method for predicting travel times on freeways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 200–207, 2004.

[8] J. Myung, D.-K. Kim, S.-Y. Kho, and C.-H. Park, "Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2256, pp. 51–59, 2011.

[9] S. I. J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *Journal of Transportation Engineering-Asce*, vol. 129, no. 6, pp. 608–616, 2003.

[10] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[11] D. Park, L. R. Rilett, and G. Han, "Spectral basis neural networks for real-time travel time forecasting," *Journal of Transportation Engineering-ASCE*, vol. 125, no. 6, pp. 515–523, 1999.

[12] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, no. 2, pp. 253–261, 2001.

[13] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C-Emerging Technologies*, vol. 13, no. 5-6, pp. 347–369, 2005.

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

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

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

[17] G. D. Cameron and G. I. Duncan, "Paramics–parallel microscopic simulation of road traffic," *The Journal of Supercomputing*, vol. 10, no. 1, pp. 25–53, 1996.

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

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

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

[21] J. W. C. van Lint, N. J. van der Zijpp, and Trb, "Improving a travel-time estimation algorithm by using dual loop detectors," *Transportation Data Research: Planning and Administration*, no. 1855, pp. 41–48, 2003.

[22] C. A. Quiroga and D. Bullock, "Travel time studies with global positioning and geographic information systems: an integrated methodology," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 1, pp. 101–127, 1998.

[23] Highways Agency network journey time and traffic flow data, Highways England. https://data.gov.uk/dataset/dft-eng-srn-routes-journey-times, accessed Septemper 2014.

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

[25] A. Graves, *Supervised sequence labelling*. Springer, 2012.

[26] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with lstm recurrent networks," in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pp. 747–756, IEEE.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164.

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

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, pp. 3104–3112.

[30] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2342–2350.

[31] C. LEE GILES, G. M. Kuhn, and R. J. Williams, "Dynamic recurrent neural networks: Theory and applications," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 153–156, 1994.

[32] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.