Continuous Travel Time Prediction for Transit Signal Priority Based on a Deep Network

Gang Xiong, Wenwen Kang, Feiyue Wang
The State Key Laboratory of Management and Control for Complex Systems, and Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Fenghua Zhu, Yisheng Lv, Xisong Dong
Cloud Computing Center, Chinese Academy of Sciences, Dongguan, CHINA

Jukka Riekki, Susanna Pirttikangas
Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland

***Abstract*—It has been recognized by many researchers that accurate bus travel time prediction is critical for successful deployment of traffic signal priority (TSP) systems. Although there exist a lot of studies on travel time prediction for Advanced Traveler Information Systems (ATIS), this problem for TSP purpose is a little different and the amount of literature is limited. This paper proposes a deep learning based approach for continuous travel time prediction problem. Parameters of the deep network are fine-tuned following a layer-by-layer pre-training procedure on a dataset generated by traffic simulations. Variables that may affect continuous travel time are selected carefully. Experiments are conducted to validate the performance of the proposed model. The results indicate that the proposed model produces prediction with mean absolute error less than 4 seconds, which is accurate enough for TSP operations. This paper also reveals that, except for obvious factors like speed, travel distance and traffic density, the signal time when the prediction is made is also an important factor affecting travel time.**

Keywords—travel time prediction; transit signal priority; deep learning; deep network

I. INTRODUCTION

The segment travel time prediction problem this paper focuses on is derived from another problem called active transit signal priority. Transit signal priority (TSP) is a kind of traffic signal control strategy that gives priority of right-of-way to public transit vehicles (buses) to facilitate their pass through signalized intersections [1] meanwhile minimums the adverse impact on the non-priority traffic streams. This strategy can improve the service quality of public transport and thus brings many advantages, including attracting ridership to public transit, alleviating urban traffic congestion, reducing vehicle exhaust emission and benefiting the health of citizens. There are four functional components of a TSP system, bus detection, priority request generation (PRG), priority request server (PRS) and TSP control. They co-work with each other as following [2]. The bus detector is installed a distance upstream to the intersection. On detection of a bus, the PRG is notified with detected vehicle data (identity, location, arrival time, passing speed etc.) and then generate a priority request. The PRS receives the generated priority request and decides whether to grant priority based on the defined conditions. Once granted, the TSP controller initiates action to provide priority for the bus based on the priority control strategies implemented in the system. When a signal is received from the downstream bus detector indicating that the bus has cleared the intersection, the TSP controller restores the normal signal timing and the priority operation is finished.

There are three most commonly used signal action in TSP operation. They are green extension, red truncation and phase insertion. According to the predicted arrival time of the bus, different action should be activated. To be more specific, if the bus is predicted to arrive at the intersection during green phase but the left green time is not long enough for the bus to pass through, the current green phase should be extended. If the predicted arrival time of the bus is at red phase which is just prior to the bus green phase, the current red phase should be truncated to switch to bus green phase in advance such that the bus can go through the intersection continuously. Otherwise the bus green phase should be inserted when the bus arriving the intersection. From this we can see that a critical issue encountered in implementing TSP is to predict the time it takes for the bus to travel from the detection location to the intersection. If the prediction is not accurate enough, it may happen that the bus is granted priority when it is not needed or is not granted when it is needed, both of which could produce badly adverse impact on the bus or non-priority traffic. Suppose that, for example, a bus is detected by the upstream detector when the bus phase is green. It is predicted that there will be not enough green time when the bus arrives at the intersection such that the green extension actin is activated. Since the prediction is not accurate, the bus actually arrives at the intersection early that the extended green time is not used. This granted while unnecessary priority operation makes bad impact on the non-priority traffic streams.

Many researchers and traffic engineers have recognized that accurate prediction of bus travel time from the detection location to the intersection is the key to success in implementing TSP control [3][4]. This can be modelled as a travel time prediction problem which many researchers have contributed their efforts to. Mori et al. divide these models into four categories, naive models, traffic theory based models, data-based models, and combined or hybrid models [5]. Naive models are simple, and do not need any training or parameter estimation. They simply predict the travel time as the most up-to-date travel time [6], simple filtering or weighting of historical travel time [7], or integration of both [8]. Traffic theory based models try to recreate the traffic status at each time step repeatedly and derive travel times from predicted state variables. They are usually implemented in traffic

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.92

simulation tools, like CORSIM [9], PARAMICS [10], AND DynaMIT [11] etc. These models have powerful representation ability of traffic details but need more expertise to be built. The accuracy of travel time prediction relies heavily on the similarity between the recreated and actual traffic situations, which is not always assured [5].

Benefiting from the development of statistics and machine learning methods, a lot of data based prediction models are proposed. The target of this kind of models is to find an appropriate data interpretative structure f and its parameters which maps the input and output data with minimal errors. Rice and van Zwet use a linear regression model with time-varying coefficients and the work gives good results [12]. Kalman filter method [13] and autoregressive-integrated moving average (ARIMA) models [14] are also proved to be effective for this problem. Instead of taking the input data directly, support vector regression (SVR) method firstly maps the input data into higher dimensional space with a specifically designed kernel such that the relationship between modified input data and target variable is linear [15]. Among those data based models, artificial neural network (ANN) is one of the most widely explored since its powerful capability of capturing nonlinear relationships or patterns underlying the data. There is a variety of types of ANNs to be chosen from, including conventional multilayer feed forward neural network [16][17], counter propagation neural network [18], object-oriented neural network [19] and spectral basis neural network [20]. Data based models can be easily built by researchers with little expertise in traffic theory. But it usually requires much larger amount of data to train these models which are not always available. And the trained models can only "get used to" the study site where the training data are collected As a result they are not always transferable to others sites [5].

To take advantages of different models, they are combined to improve the prediction accuracy, which are called combined or hybrid models. Xia et al. propose a multistep predictor combing a seasonal ARIMA model with and adaptive Kalman filter [21]. Zheng et al. build a neural network (NN) model whose prediction is combined based on Bayesian rule from two single NN predictors, i.e., a back propagation NN and a radial basis function NN [22]. In [23], the extended Kalman filter is applied to train the state-space neural network which is used to predict the urban arterial travel time. There are also researchers exploring the integration of neural network with fuzzy logic system, like [24] as an example. These combined or hybrid models are reported to perform better than conventional single predictors in their papers. This could be a promising direction in travel time prediction problem, but more and further research is needed.

Most of the existing literature on travel time prediction reviewed above is intended for Advanced Traveler Information Systems (ATIS). For TSP purpose, the characteristics and requirements of the problem are a little different. For ATIS application, the task is to predict the travel time of a link, or a series of links. But for TSP operation, the prediction space span is much shorter, i.e., just from the upstream detector to the intersection, which is usually less than 500 meters. Little variance in prediction can make a big difference. For this reason, the accuracy of prediction must be stricter for TSP operation, i.e. within second level, instead of minute level for ATIS. Although the amount of literature is limited, there are some researchers trying to address this problem. Tan et al. propose an optimal a posterior parameter estimation model whose prediction is derived from both historical and real-time GPS probe data [4]. Li et al. apply the model of Tan et al. to their predictive TSP strategy. Traffic simulations suggest that this predictive strategy has better performance than conventional TSP strategies. Lee et al. develop a microscopic traffic simulation model to predict the transit travel time along an intersection approach [25]. These work integrate transit travel time prediction with TSP in different contexts, but both indicate that this is a promising direction for improving TSP performance further.

ANNs have shown powerful capabilities in prediction problems since they are firstly proposed. Recent breakthroughs in deep leaning networks [26], which are also ANNs but with multiple-layer architectures or deep architectures, have been proved to be successful in prediction [27] and other problems. In this paper, we propose a deep network structure which is composed of a stacked autoencoder (SAE) and a predictor to address the transit travel time prediction problem for TSP. The rest of the paper is organized as follows. In section II the proposed prediction model based on deep learning network is demonstrated. In section III the experiment procedure is described, including summary and analysis of the experiment results. Finally in section IV some conclusions are drawn.

II. Methodology

A. The structure of SAE

A SAE, inferred from its name, is built by stacking a series of autoencoders (AEs). An autoencoder is a neural network which has the same number of nodes in the input and output layers. The simplest form is a NN with only one hidden layer. Its structure is shown in fig. 1. Given an input sample x, the AE first encodes it to its encoding representation y, as seen in equation 1. The encoding y is then decoded to the output z, as seen in equation 2. The encoding and decoding information is stored in matrices W_1 and W_2. An AE's responsibility is to ensure that the reconstructed output z is as similar to the input x as possible.

$$y(x) = f(W_1 x + b) \tag{1}$$

$$z(x) = g(W_2 y + c) \tag{2}$$

where W_1 and W_2 are the encoding and decoding matrices respectively, b and c are encoding bias vector and decoding bias vector respectively, $f(x)$ and $g(x)$ are the activation functions where the sigmoid function $1 / (1 + exp(-x))$ is applied in this paper.

The AE can be trained using the conventional back-propagation (BP) approach. Different from other NN training that requires both input data and labels (or target output) data, training of AE only requires input data since it takes each input sample itself as its label. The target is to minimize the reconstruction error $L(X, Z)$ where

$$L(X, Z) = \frac{1}{2}\sum_{i=1}^{N}\left\|x^{(i)}\text{-}z^{(i)}\right\|^2. \tag{3}$$

In equation 3, X is the set of N input samples $\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\}$ and Z is the set of corresponding

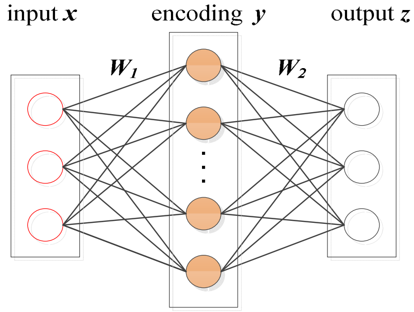

Fig. 1. The structure of a one-hidden-layer autoencoder.

reconstructed output $\{z^{(1)}, z^{(2)}, \cdots, z^{(N)}\}$. And the notation $\|\cdot\|$ represents the 2-norm of a vector.

An SAE is created by stacking a series of AEs to form a deep network following the consistency constraint that the number of nodes in the previous AE's hidden layer must be equal to that in the next AE's input layer. To be clearer, considering an SAE stacked by l AEs as shown in fig. 2. These AEs are stacked in such a way that, for each AE, the decoding part is dumped and its hidden nodes are connected directly to the hidden layer of the next AE acting as the next input layer.

Given a set of input samples, the training of an SAE is straightforward. Firstly the training data set is applied to train the first AE as described above such that $W_1^{\,1}$ and $W_2^{\,1}$ are obtained. Then for each of the i^{th} (i = 2, 3, … , l) AE, the encoding of the $(i\text{-}1)^{th}$ AE y^{i-1} is taken as the input to train this AE such that $W_1^{\,i}$ and $W_2^{\,i}$ are obtained. In this way the training of the SAE is completed.

B. SAE based Deep Network

The reason why SAE is effective in data modelling is that it extracts and represents features underlying the data layer by layer. To perform the prediction functionality, a predictor is needed to be added on the top of the SAE to form a deep architecture model for travel time prediction. The structure of the proposed deep network is shown in fig. 3. In this paper, we choose the logistic regression model as the predictor. The predictor takes the output of the SAE's last layer as its input and gives a prediction value corresponding to the input sample. But before this, the SAE together with the predictor needs to be trained again to fine-tuning the parameters of the deep network. The training procedure is based on the work of Hinton et al. [26] which can be summarized as following:

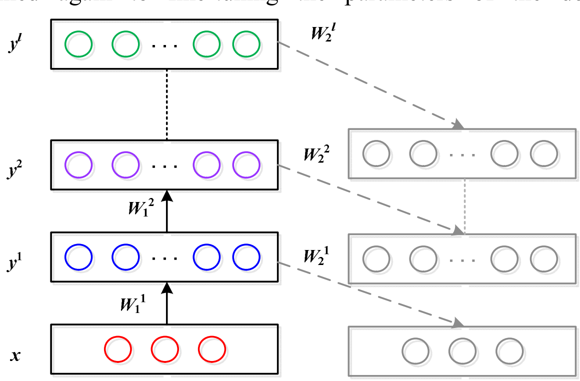

Fig. 2. The structure of an SAE with l layers.

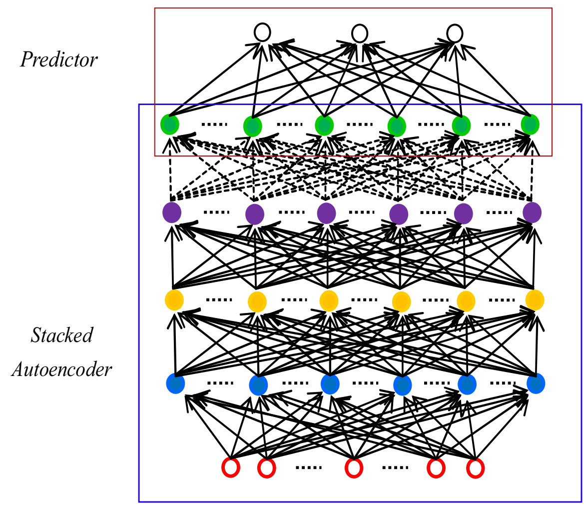

Fig. 3. Combine an SAE and a predictor to form a deep network.

1). Train the first layer as an autoencoder by minimizing the reconstruction error with the training data as the input.

2). Train the i^{th} (i = 2, 3, … , l) layer as an autoencoder taking $(i\text{-}1)^{th}$ layer's output as the input.

3). Iterate as in 2) until $i = l$ where l is the total number of AEs in the SAE.

4). Use the output of the SAE's last layer as the input for the prediction layer, and initialize its parameters randomly or by supervised training.

5). Fine-tune the parameters of all layers in a supervised way taking training data as the input and their labels as the target output. The fine-tuning process can be accomplished using the conventional back-propagation method with gradient-based optimization technique.

Steps 1 ~ 3 make up the pre-training procedure as described in the previous section. The target is to obtain reasonable initial values for the parameters of the SAE. Steps 4 ~ 5 retrain the SAE and prediction layer as a whole to fine-turn the parameters of the model.

III. EXPERIMENTS

A. Data acquisition

The study site is a typical four-approach signalized intersection. Each approach has one right turn lane, two straight through lane and one left turn lane. The open end of each approach is connected to a zone numbered from 1 to 4 as traffic sources and sinks. We modelled the intersection in a microscopic traffic simulation software called Paramics [10], as seen in fig. 4. The signal cycle length is 112 seconds and it is split into four phases. The split length and allowed movements for each phase are depicted in table I. Since right

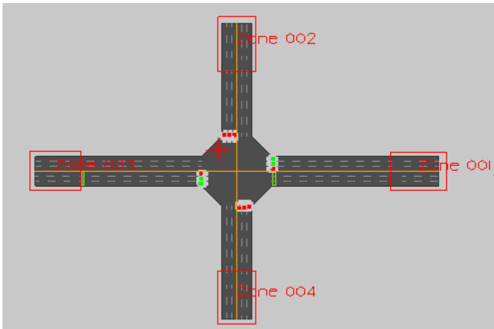

Fig. 4. The study site modelled in Paramics.

turn movements are always allowed, right turn traffic is excluded for simplicity. The temporal span of the simulation is 6 hours, from 6:00 a.m. to 11:00 a.m. Total traffic count of straight through and left turn vehicles for each approach is shown in table II. To simulate the temporal variance of traffic volume distribution, the total traffic count is released according to the distribution profile in table III. From this table we can see that nearly half (43%) of the vehicles are released during morning peak hours 7:00 a.m. ~ 9:00 a.m.

TABLE I. PHASES AND SPLITS OF THE STUDY INTERSECTION.

Phase number	**Allowed movements**	**Green length (s)**	**Ampler length (s)**
1		30	3
2		20	3
3		30	3
4		20	3

* All right turns are always allowed.

TABLE II. TRAFFIC VOLUME FOR EACH OD PAIR (UNIT: VEHICLES).

To / **From**	Zone 1	Zone 2	Zone 3	Zone 4	Total
Zone 1	0	0	3520	1200	4720
Zone 2	1200	0	0	2400	3600
Zone 3	3840	1200	0	0	5040
Zone 4	0	3040	1000	0	4040
Total	5040	4240	4520	3600	17400

TABLE III. TEMPORAL DISTRIBUTION OF TRAFFIC VOLUMES.

Time	6:30	7:00	7:30	8:00	8:30	9:00
C. P. (%)	3	9	18	30	42	52
Time	9:30	10:00	10:30	11:00	11:30	12:00
C. P. (%)	61	69	75	82	90	100

* C. P.: cumulative percentage.

Various types of vehicle data can be collected through the API (Application Programming Interface) functions provided by the software. But only those that can be collected in real world should be used as experiment input considering practical application. Data are collected at two points in out simulations. The start point of data collection is triggered when a vehicle passes some place upstream to the intersection. At this point a part of data are collected including current time, passing speed, traffic flow and queue length of the lane. The determination of end point of data collection is a little different from previous studies. In previous studies, end point of data collection is triggered when a vehicle passes the stop line. But if the traffic light is red and there exists queues at the stop line when a bus arrives, it will not be able to cross the stop line until the next green phase comes and the queues are cleared. As we know, TSP operation is applied to ensure continues travelling through the intersection for buses. Thus in this study the end point of data collection is triggered when a vehicle stops at the tail of a queue because of red light, or when it passes the stop line without any stops because of green light. That's why it's called "continuous" travel time prediction. At this point, current time is recorded thus the travel time from the start point to the endpoint of data collection is inferred. This travel time is called continuous travel time (CTT) in this paper to differ from the definition of travel time in previous studies. All the data collected at the start and end point can be obtained by Global Positioning Systems (GPS) in real world.

The start point of data collection is set 250 meters upstream to the intersection which is nearly half the distance between two consecutive intersections. Since there are no stations between the upstream detector and the intersection, we assume that travel times of buses differ nothing from those of normal vehicles. Traffic data of all vehicles are collected to obtain a large data set. There are about 5,000 vehicles passing the study site in one simulation. Data are collected after 30 minutes warm-up period. To get enough number of samples we conducted 30 times of simulations with different random seeds. Total 15,0694 samples are collected and the data set is separated into a training set with 80% samples and a testing set with 20% samples. The training set is used to train the deep network model to get optimal structure and parameters. The testing set is used to test the performance of the trained model. We use mean absolute error (MAE) of prediction as measurement of effectiveness (MOE). It is defined as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|t_i - \hat{t}_i|$$

where n is the total number of samples in testing set, t_i and $\hat{t}_i$ are the actual CTT and predicted CTT of the i^{th} sample respectively.

B. Experiments and results

To identify variables affecting CTT, we start by using the most direct and intuitive variables firstly and gradually adding other variables that possibly related. Based on the most naive idea, the CTT can be calculated as

$$\mathrm{CTT} = \frac{distance\ travelled}{travelling\ speed} = \frac{l - q}{v}$$

where l is the distance between the start point and the stop line which is constant, q and v are the queue length and travelling speed when the vehicle passing the start point respectively. As can be imagined, the travelling speed of a passing vehicle can not keep constant especially when getting close to the intersection. It is affected by many factors among which traffic density is one of the most relevant. Traffic density can be roughly expressed by traffic count which is denoted by c at a specific intersection approach. Thus we use q, v and c as the input of the proposed deep network model firstly (l is constant thus can be excluded in ANN based models).

TABLE IV. OPTIMAL STRUCTURES AND MAES FOR DIFFERENT GROUPS OF INPUT VARIABLES.

# of AEs	*q, v, c*		*q, v, c, s*		*q, v, c, s, t*		*q, v, c, s, f*		*q, v, c, s, t, f*	
	Optimal structure	***MAE (s)***	***Optimal structure***	***MAE (s)***	***Optimal structure***	***MAE (s)***	***Optimal structure***	***MAE (s)***	***Optimal structure***	***MAE (s)***
1	[~, 8]	4.84	[~, 5]	4.25	[~, 6]	4.29	[~, 15]	4.29	[~, 19]	4.30
2	[~, 9, 19]	**4.83**	[~, 3, 4]	4.19	[~, 10, 8]	4.21	[~, 5, 20]	4.18	[~, 8, 13]	4.22
3	[~, 12, 7, 18]	4.84	[~, 17, 15, 10]	**4.13**	[~, 16, 10, 18]	4.17	[~, 17, 15, 5]	4.16	[~, 4, 12, 8]	4.17
4	[~, 9, 5, 20, 9]	4.87	[~, 6, 6, 4, 7]	4.20	[~, 11, 12, 19, 6]	**4.15**	[~, 5, 7, 10, 19]	4.21	[~, 4, 19, 17, 10]	4.20
5	[~,9,8,15,16,11]	4.86	[~,5,3,13,13,15]	4.16	[~,10,9, 9, 8, 12]	4.22	[~,15,6,7,17,15]	**4.13**	[~,11,12, 8, 5, 9]	**4.15**

Another problem is to determine the optimal structure of the deep network. Different combination of number of layers and number of nodes in each layer makes different structure of a deep network. In this paper, according to practical experience, we limit the number of AEs between 1 and 5, and the number of hidden nodes in each AE between 3 and 20. To avoid combination explosion problem in grid search approach which is too time consuming, for cases where the number of AEs is 2, 3, 4 and 5, the number of hidden nodes in each AE is randomly generated within its limitation for 300 iterations. That is, among all the different possible structures, exactly 18 + 300 * 4 = 1218 are randomly selected to conduct the simulations, and the structure with the least MAE is determined as the optimal structure for this group of input variables. For different groups of input variables, we perform the process above to determine the according optimal structure.

According to out simulation results, for input of q, v and c, the least MAE is 4.83 seconds and the optimal structure is [~, 9, 19] (see table IV), which means the deep network has two AEs stacked whose number of hidden nodes are 9 and 19 respectively, and ~ represents the number of input variables.

As for CTT prediction problem, another important but easily ignored variable that affects vehicle travelling speed is the signal time when vehicle passes the start point. As can be imagined, if a vehicle passes the start point when the traffic light is red and the left red time is long enough, the vehicle has to decelerate and possibly stops advanced to the stop line. If a vehicle passes the start point when the traffic light is green and the left green time is long enough, the vehicle can keep going through the intersection without deceleration. If otherwise, i.e., the traffic light is turning from red to green or the opposite, the vehicle has to firstly stop and go or the opposite. We denote the signal time variable as s. It is divided by the signal cycle length to normalize it in range [0, 1]. This time s together with q, v and c is used as input. From the results in table IV we can see that the least MAE is 4.13 seconds and the optimal structure is [~, 17, 15, 10]. The MAE is decreased by 14.5% and this supports out opinion that the signal time when vehicle passes the start point is an import factor in CTT prediction problem.

Another two variables, traffic flow (denoted as f) and time of day (denoted as t) when the prediction is made, are also thought to be related to CTT. We conducted additional experiments with another three groups of input variables, i.e., with single t, single f, both t and f added each. The results are displayed in table IV. The overall least MAE is 4.13 seconds and the optimal structure is [~, 17, 15, 10] when the input variables are q, v, c, and s. The results also indicate that adding t or f as input variables can not actually improve the performance of CTT prediction which is uncommon with previous studies. This is possibly because of the fact that the experiment data in this paper are generated by traffic simulations. There are much less noise in the data. Relationships between different variables are determined and reflected finely in the data. For example, the characteristics of f is determined by t (according to the distribution profile) and t is reflected by signal time s, given the static simulation configuration. This makes the trial of adding t or f as input makes no difference with the prediction performance.

TABLE V. MAES AND MAPES VARY WITH PREDICTION SPATIAL SPAN.

Distance (m)	**MAE (s)**	**MAPE (%)**
150	3.72	28.49
250	4.13	20.36
350	4.53	18.36
500	4.55	13.89
1000	4.58	7.96
1500	4.80	5.79

Although the MAE performance of the proposed mode is acceptable, some researchers may argue that how about the prediction performance of the mean absolute percentage error (MAPE). It is defined as

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|t_i - \hat{t}_i|}{t_i}.$$

Indeed, the resulting MAPE is between 20% ~ 26%, which is relatively high. This is because the start point is too close to the intersection (about 250 meters). According to the collected data, average CTT is 21.4 seconds. Little error in prediction contributes much to MAPE. We conducted another group of simulations varying the distance between the start point and the intersection from 150 meters to 1500 meters. The optimal structure [~, 17, 15, 10] is used and variables q, v, c, and s are taken as model input. Resulting MAEs and MAPEs are shown in table V. From the results we can see that when the prediction spatial span is long enough the MAPE performance of the proposed model is comparable with state-of-the-art prediction models.

IV. CONCLUSIONS AND FUTURE WORK

Transit signal priority is an effective strategy to promote vehicles' pass through intersections. However, accurate prediction of bus arrival time remains a critical problem to be addressed. And this problem has not attracted too much attention from field researchers. We proposed a deep learning based model to predict the time it takes to travel some distance before the stop line continuously. Simulation experiments indicated that the MAE performance of the proposed model can meet the requirement of practical application, and the MAPE performance is comparable with state-of-the-art prediction models if the prediction span is long enough. The results also revealed that the continuous travel time is highly related with the traffic light status when the prediction is made. The limitation is that the experiment data used in the paper is generated by traffic simulations instead of being collected in real world. This limits the explanation power of the results.

For future work, the proposed model should be validated against data set collected in real world. Furthermore, the predictor applied in our paper is just a logistic regression model. Extending it to more powerful predictors may make further performance improvement. Relationships between CTT and other factors are also worth to be explored. For example, how can different phase plans affect prediction accuracy, how the prediction accuracy differs with different levels of traffic flow, i.e., high, medium and low flow, and how the spatial and temporal structures underlying in data change the prediction performance when the model is applied to multiple consecutive intersections.

ACKNOWLEDGMENT

This work is supported in part by NSFC (Natural Science Foundation of China) projects (71232006, 61233001, 61174172), Finnish TEKES program: From Data to Intelligence (D2I), and Chinese Dongguan's Innovation Talents Project (Gang Xiong).

REFERENCES

[1] Smith, Harriet R., Brendon Hemily, and Miomir Ivanovic. "Transit signal priority (TSP): A planning and implementation handbook." (2005).

[2] O'Brien, W. "Design and Implementation of Transit Priority at Signalized Intersections: A Primer for Transit Managers and a Review of North American Experience". Canadian Urban Transit Association STRP Report 15, Toronto, Canada, 2000, p.31.

[3] Li, Jie, et al. "Predictive Strategy for Transit Signal Priority at Fixed-Time Signalized Intersections." Transportation Research Record: Journal of the Transportation Research Board 2311.1 (2012): 124-131.

[4] Tan, Chin-Woo, et al. "Prediction of transit vehicle arrival time for signal priority control: Algorithm and performance." Intelligent Transportation Systems, IEEE Transactions on 9.4 (2008): 688-696.

[5] Mori, Usue, et al. "A review of travel time estimation and forecasting for Advanced Traveller Information Systems." Transportmetrica A: Transport Science ahead-of-print (2014): 1-39.

[6] Schmitt, Erick J., and Hossein Jula. "On the limitations of linear models in predicting travel times." Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE. IEEE, 2007.

[7] Hoffmann, G., and J. Janko. "Travel times as a basic part of the LISB guidance strategy." Road Traffic Control, 1990., Third International Conference on. IET, 1990.

[8] Wunderlich, Karl E., David E. Kaufman, and Robert L. Smith. "Link travel time prediction for decentralized route guidance architectures." Intelligent Transportation Systems, IEEE Transactions on 1.1 (2000): 4-14.

[9] Halati, Abolhassan, Henry Lieu, and Susan Walker. "CORSIM-corridor traffic simulation model." Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities. 1997.

[10] Cameron, Gordon DB, and Gordon ID Duncan. "PARAMICS—Parallel microscopic simulation of road traffic." The Journal of Supercomputing 10.1 (1996): 25-53.

[11] Ben-Akiva, Moshe, et al. "DynaMIT: a simulation-based system for traffic prediction." DACCORS Short Term Forecasting Workshop, The Netherlands. 1998.

[12] Rice, John, and Erik Van Zwet. "A simple and effective method for predicting travel times on freeways." Intelligent Transportation Systems, IEEE Transactions on 5.3 (2004): 200-207.

[13] Yang, Jiann-Shiou. "Travel time prediction using the GPS test vehicle and Kalman filtering techniques." American Control Conference, 2005. Proceedings of the 2005. IEEE, 2005.

[14] Yang, Jiann-Shiou. "A study of travel time modeling via time series analysis." Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on. IEEE, 2005.

[15] Wu, Chun-Hsin, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression." Intelligent Transportation Systems, IEEE Transactions on 5.4 (2004): 276-281.

[16] Park, Dongjoo, and Laurence R. Rilett. "Forecasting freeway link travel times with a multilayer feedforward neural network." Computer‐Aided Civil and Infrastructure Engineering 14.5 (1999): 357-367.

[17] Li, Chi-Sen, and Mu-Chen Chen. "Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks." Neural Computing and Applications 23.6 (2013): 1611-1629.

[18] Dharia, Abhijit, and Hojjat Adeli. "Neural network model for rapid forecasting of freeway link travel time." Engineering Applications of Artificial Intelligence 16.7 (2003): 607-613.

[19] Dia, Hussein. "An object-oriented neural network approach to short-term traffic forecasting." European Journal of Operational Research 131.2 (2001): 253-261.

[20] Park, Dongjoo, Laurence R. Rilett, and Gunhee Han. "Spectral basis neural networks for real-time travel time forecasting." Journal of Transportation Engineering 125.6 (1999): 515-523.

[21] Xia, Jingxin, Mei Chen, and Wei Huang. "A multistep corridor travel-time prediction method using presence-type vehicle detector data." Journal of Intelligent Transportation Systems 15.2 (2011): 104-113.

[22] Zheng, Weizhong, Der-Horng Lee, and Qixin Shi. "Short-term freeway traffic flow prediction: Bayesian combined neural network approach." Journal of transportation engineering 132.2 (2006): 114-121.

[23] Liu, Hao, et al. "Predicting urban arterial travel time with state-space neural networks and Kalman filters." Transportation Research Record: Journal of the Transportation Research Board 1968.1 (2006): 99-108.

[24] Zhang, Yunlong, and Hancheng Ge. "Freeway Travel Time Prediction Using Takagi–Sugeno–Kang Fuzzy Neural Network." Computer‐Aided Civil and Infrastructure Engineering 28.8 (2013): 594-603.

[25] Lee, Jinwoo, et al. "Advanced transit signal priority control with online microsimulation-based transit prediction model." Transportation Research Record: Journal of the Transportation Research Board 1925.1 (2005): 185-194.

[26] Hinton, Geoffrey, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

[27] Lv, Yisheng, et al. "Traffic Flow Prediction with Big Data: A Deep Learning Approach."