Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 155 (2019) 154–167
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Utilizing data mining techniques to predict expected freeway travel time from experienced travel time

Hasan M. Moonam[a], Xiao Qin[a,*], Jun Zhang[b]

[a] *Civil and Environmental Engineering, University of Wisconsin-Milwaukee, 3200 N Cramer St., Milwaukee, WI 53211-3314, United States*
[b] *Electrical Engineering & Computer Science, University of Wisconsin-Milwaukee, United States*

Received 9 October 2017; received in revised form 27 January 2018; accepted 29 January 2018
Available online 7 February 2018

Abstract

As the most important real-time traveler information, travel time can be either experienced or expected (i.e. to be experienced). When a vehicle completes a trip, the travel time refers to the experienced travel time. In contrast, when a vehicle starts its journey, the travel time is unknown but can be predicted, which is the expected travel time. Although the experienced travel time is termed as the real-time travel time, a traveler may encounter a somewhat different travel time (from expected travel time) due to the changing traffic conditions. Therefore, expected travel time needs to be predicted. In this study, the expected travel time was predicted from the experienced travel time using the data mining techniques such as k-nearest neighbor (k-NN), least squares regression boosting (LSBoost) and Kalman filter (KF) methods. After comparing the performances of KF to corresponding modeling techniques from both link and corridor perspectives, it is concluded that the KF method offers superior prediction accuracy in a link-based model. Moreover, the effect of different noise assumptions was examined and it is found that the steady noise computed from the full-dataset had the most accurate prediction. A data processing algorithm, which processed more than a hundred million records reliably and efficiently was also introduced.

Keywords: Experienced and expected travel time; Arrival and departure time based travel time; Travel time prediction; Data mining; Kalman filter

1. Introduction

Travel time is an important component of Advanced Traveler Information Systems (ATIS), as it is a key factor for travelers who are faced with non-recurring congestion [33]. Aside from measuring transportation system performance, travel time has been used to predict future travel time and traffic state, which help the traffic operations room in versatile ways. Amongst all available techniques, Bluetooth has emerged as one of the fastest growing data collection technologies whose market share is continuing to rise, mainly due to its cost effectiveness [5,38]. Bluetooth is a probe-based [45] Automatic Vehicle Identification (AVI) technique used for collecting travel time data. Each Bluetooth

* Corresponding author.
E-mail addresses: hmoonam@uwm.edu (H.M. Moonam), qinx@uwm.edu (X. Qin), junzhang@uwm.edu (J. Zhang).

https://doi.org/10.1016/j.matcom.2018.01.006

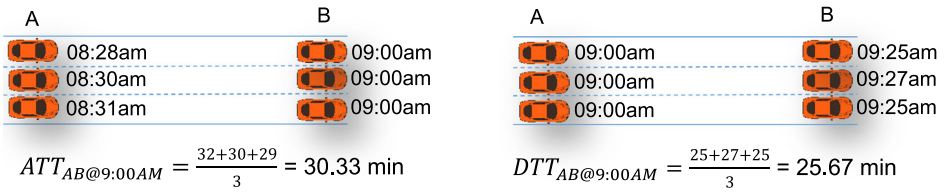

Fig. 1. Estimating ATT and DTT of link AB at 9 am.

device contains a unique electronic identifier known as a Media Access Control (MAC) address/identifier. Devices with MAC addresses that are in range can be logged as long as a simple antenna is mounted adjacent to the roadway. Travel time and corresponding traffic speed can be estimated by the timestamp difference between two consecutive stations with matching MAC addresses [2].

Travel time is measured by the time elapsed when a traveler moves between two distinct spatial positions [7]. Over a decade, various studies have attempted to define travel time estimation (e.g. instantaneous travel time, experienced travel time [55], and predicted travel time [4]), but the definitions lack clarity, which has created confusion and inconsistency in data collection and analysis [4,50]. Recently, one study made clear the distinction between arrival time-based link travel time and departure time-based link travel time [34].

Arrival time-based link travel time (ATT) and departure time-based link travel time (DTT) have two different estimation algorithms. ATT refers to the travel time associated with arrival at the destination, while DTT refers to the travel time associated with departure from the origin. In practice, ATT is the experienced travel time (t_{expn}) that is calculated using arrival and departure times of the vehicles when both are available. On the other hand, DTT is the expected travel time (t_{expt}) that is predicted at the time of departure when arrival time is unavailable. To get ATT or t_{expn} and DTT or t_{expt} , assume that two vehicles (V_1 and V_2) start at 8:30 am and 9:00 am from point A, respectively. If the assumed clock time is now 9:00 am and the first vehicle (V_1) has just arrived at point B. Then, the arrival time based travel time for link AB, $ATT_{AB@9:00\text{AM}} = 30$ min (based on the experienced travel time of V_1). The departure time based link travel time, $DTT_{AB@9:00\text{AM}} = unknown$. If the arrival time for V_2 at point B could be predicted (say, 9:25 am), the departure time based travel time would be, $DTT_{AB@9:00\text{AM}} = 25$ min (based on the expected travel time of V_2). Since the DTT at 9:00 am is unavailable until a later time, i.e., until the V_2 travels the link AB, it is understandable that the DTT at 9:00 am requires a prediction of travel time. Fig. 1 illustrates the concept of estimating ATT or t_{expn} and DTT or t_{expt} for a route considering multiple vehicles.

Practitioners usually treat ATT (or t_{expn}) as the travel time due to the lack of available DTT (or t_{expt}); however, this reported ATT is one-step (step interval = travel time) earlier than the actual travel time to be experienced (DTT) by drivers. Although ATT and DTT differ slightly in a free-flow condition, the difference can sharply escalate at the onset and end of traffic congestion. ATT usually lags behind DTT during a transition of traffic state, and the difference starts to decrease when the traffic state becomes stable. Information on travel time during a transitional state, as opposed to a stable state, is more important to travelers. Ideally, the travel time should be the predicted travel time that will be experienced by a traveler, or DTT. This predicted travel time also helps ensure proper and proactive operations and management of traffic in a network. Unfortunately, few studies have distinguished between DTT and ATT or attempted to estimate DTT [34].

The overarching goal of this study is to develop a comprehensive model for short-term freeway travel time prediction using Bluetooth data. A dynamic filtering algorithm was proposed to accurately estimate ATT and thus, reliably predict DTT. An efficient computer algorithm was developed to process, refine, and integrate a massive Bluetooth dataset, which filtered the travel time. Finally, prediction algorithms were examined to predict DTT from real-time ATT, and the better performance of the proposed prediction was observed.

2. Literature review

Travel time data are subject to outliers. The main purpose of outlier detection algorithms is to detect extreme travel times that result from sampling bias. Fixed-range outlier filtering methods are not suitable for travel time filtering due to local travel time turbulences, especially when they occur at the onset or end of congestion. To avoid imposing arbitrary fixed-bound, researchers have introduced moving average speed based lower- and upper-bound [24] and

data-driven real-time adaptive-bound methods [15]. Dion and Rakha [15] incorporated a few simple yet significant alterations in their proposed adaptive method, which offers an alternative to conventional algorithms like percentile, deviation, and traditional (modified) z- or t-statistical test [12,35]. The main alteration includes expanding the data validity window when three consecutive observations fall either above or below (same side) the window. While this key adjustment helps capture sudden changes in travel time trends, it is prone to the inclusion of extreme outliers, and therefore compromises the accuracy of travel time estimation. In response to Dion and Rakha [15] method, Moghaddam and Hellinga [39] proposed a proactive method that uses a pattern recognition model, which showed superior performance. But the author acknowledged that the performance of outlier detection algorithms cannot be objectively quantified when the algorithms are applied to field data. Appropriate estimation of travel time is possible only when an effective outlier filter is used. Many studies have examined accurate estimation of travel time in a real-time fashion [15,39,46,47]. In most cases, the sophistication of filtering algorithms to maximize the accuracy led to a certain level of complexity in real-time applications. Therefore, a simplified version of these proposed algorithms is preferred.

Broadly, travel time prediction methods can be classified into the classical approach [43] including statistical [46] and time series models [1,25], and the data mining approach [41,52,54,59,60]. Due to the instability of traffic states, most classical approaches have shown to be incapable of better prediction, especially with regard to structured and unstructured data [52]. Therefore, advanced data mining methods have become popular to predict travel time. As such, Jenelius and Koutsopoulos [29] proposed a multivariate probabilistic principal component analysis method that predicts travel time based on the expected distribution of link travel times. The method provides superior results to the k-NN method. Zhong et al. [61] introduced an online travel time prediction system without sacrificing computational efficiency. The system adopts functional principal component analysis framework and utilizes historical and real-time travel time data to predict link travel time. Zhang et al. [58] applied a two-component generalized autoregressive conditional heteroskedasticity (GARCH) model that captures trend and seasonal components to improve prediction results. Fei et al. [18] employed a Bayesian inference-based dynamic linear model (DLM) to predict online short-term travel time. Sumalee et al. [49] estimated dynamic stochastic journey time distribution and predicted travel time based on stochastic cell transmission model. Zhan et al. [57] extracted travel time data from origin–destination dataset by minimizing the least squared error between the observed and expected path travel times. Zou et al. [62] exploited a space–time diurnal method to predict travel time that considers spatial and temporal correlation and diurnal pattern of travel times. In addition to vehicle trajectory-based methods, Celikoglu [8] used flow model to predict travel time that not only increases accuracy but also reduces computational complexity. Since data mining approaches fit easily with massive datasets, researchers have also applied neural networks [60], fuzzy and evolutionary techniques [59], support vector regression [54], and the k-nearest-neighbor (k-NN) [41] model to directly or indirectly predict travel time. According to Myung et al., the use of non-representative samples to train the artificial neural-network (ANN) model may lead to a non-negligible error in prediction [41]. These data mining approaches require representative samples [48] and sometimes suffer from a lack of interpretability and transferability. On the other hand, simple methods (e.g. instantaneous, historic average and clustering over specific days) exhibit low accuracy [51]. Although LSBoost, a data mining technique, has been widely used [3,14,30–32] in different fields for many years, only recently it is being popular in transportation studies such as perception reaction time [17], congestion duration [20], freight flow [40] and emission [44] prediction.

KF, an optimal recursive data processing algorithm, has been widely used with various modifications (e.g. adaptive KF [22] and extended KF [36]) in several studies including those on travel time prediction [9,11,42,56]. KF incorporates all information that can be provided and processes all available measurements to estimate the current value of the variables of interest [37]. The KF method has two components: process/system/state–space and measurement/observation. Nanthawichit et al. formed their state–space equation by declaring traffic density and space mean speed as state variables, developed their observation equation by declaring traffic volumes and spot speeds as observation variables [42]. Chen and Chien used travel time as the input variable in both of these equations; previous step travel time was multiplied by a transition matrix to obtain the state update equation [9]. A similar study was conducted using field data [11] rather than simulated data [9]. Despite promising results, these studies lack details about sources of process and measurement (variables') values.

Researchers [9–11,42,56] have modeled the state–space as a linear system to which KF was applied. The KF model uses both a priori (derived by state–space/process) and the observation of the same timestamp to get a posterior by using the update equation. Within this procedure, the model combines all available observations and prior knowledge

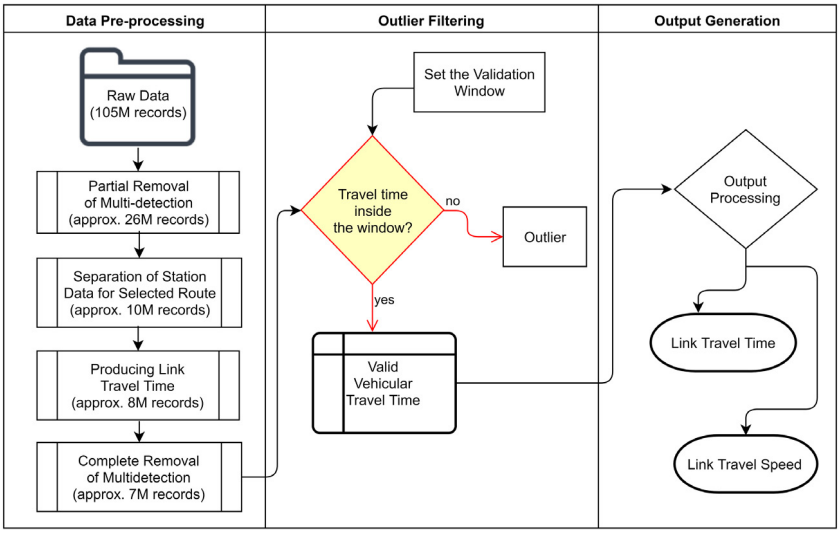

Fig. 2. Data processing procedures.

of a system in a way that statistically minimizes errors [37]. The model can be directly applied to signal processing and control systems (when observations are available) in order to filter out the noise. KF has also been widely applied to time series forecasting with a state–space model [16,26,27] where the update equation [13,16] clearly states a time latency; the next step is predicted using the observations from the current step. Hence, predicting DTT from ATT is more advantageous over a single source time series data models.

3. Data preparation and reduction

Bluetooth data contains three variables: the MAC ID of the detector, MAC IDs of the detected devices, and the detection timestamp. In spite of a simple data format, a complex processing algorithm is required to produce the final dataset from the source, which stores the entire network data in a single table. For a logged MAC ID, recorded timestamps at two consecutive stations are processed to estimate travel time and the corresponding traffic speed. The detailed description of the processing algorithm is beyond the scope of this paper; therefore, a brief description is included below with a limited description of the data characteristics.

The selected study area consists of a 62.8-mile long route, or approximately 47.5 miles on I-90 and the remaining on the Beltline Highway in Madison, Wisconsin. The route is equipped with 41 unequally spaced Bluetooth stations, resulting in 40 links. The first 21 links are on I-90, the 22nd link is on both corridors, and the remaining links are on the Beltline. The spacing varies from 1.3–3.4 miles on I-90 and 0.4–1.3 miles on the Beltline Highway. Forty-seven days' worth of data (11/16/2015–01/01/2016) containing more than 100 million records was collected from traffic in both directions. Half of the records were from outside the study-area. Each station of the one hundred stations selected captured around one million records for 47 days, or 67,680 min. However, a large portion (approx. three-fourths) of the data are either corrupted or contaminated due to multiple detections and unsuccessful detections (i.e. not detected in two consecutive stations). Fig. 2 shows the complete procedure of data processing.

A Bluetooth station usually detects a Bluetooth device in its range more than once. The number of such detections can increase significantly due to planned or unplanned slowing down/stopping of vehicles. A general inspection of the dataset revealed that such detections usually vary two to four times. Oracle queries helped clean up the multi-detection, resulting in the total number of records decreasing from 105 million to 26 million. The data was then separated by each station for the selected routes, further reducing the records to 10 million. Unsuccessful detections were automatically ignored due to the vehicle's detection timestamps from two adjacent stations. Travel times were calculated. Next, the reduced dataset of 8 million samples was processed through a robust Java-based pre-processing module that investigated each record individually and cleaned all redundant records based on the following principle:

Two detections of a vehicle at a station (STA_1) are valid separate detections if the vehicle is detected at least once at its upstream station (STA_2) within the time gap of two detections at STA_1.

For example, a vehicle detected on 08:59 am, 09:01 am and 09:08 am at STA_1, and on 09:04 am at STA_2. Detection on 09:01 am is redundant since there is no detection at *upstream station* STA_2 in between 08:59 am and 09:01 am. In addition, neither 09:01 am nor 09:08 am is a redundant detection since there is a detection at *upstream station* STA_2 on 09:04 am. Note that the Bluetooth stations were capturing both directions of traffic and the vehicle used as an example made a U-turn/return-trip. The pre-processed dataset of 7 million records was further processed to filter outliers. Finally, a Java-based programming module produced the travel time and speed data using the outlier-filtered data. Since travel direction is pertinent to travel time, this study used northbound data.

4. Methodology

The methodology section details the algorithms for outlier filtering, ATT, DTT, travel speed estimation, and travel time prediction.

4.1. Outlier filtering

A preset upper and lower boundary helped filter out the outliers. The following equation defines the lower boundary:

$$tt_{lowr} = {}^{tt_{ff}}/_{2} \tag{1}$$

where, tt_{lowr} and tt_{ff} stand for lower bound and free flow travel time, respectively. A vehicle was considered to be an outlier if its speed exceeded more than double the posted speed limit.

A dynamic validation window works best in outlier filtering [15]. The upper boundary is defined by the following equation:

$$tt_{uppr} = tt_e + n \cdot \sigma_e \tag{2}$$

where tt_{uppr}, tt_e and σ_e are upper bound, expected travel time, and expected standard deviation of travel time (samples), respectively. $n = 1, 2, 3, \ldots$.

4.2. Speed Estimation

The following equation estimates the space mean speed of a link AB with length L:

$$v = \frac{L}{\frac{1}{n}\sum_i tt_i} \tag{3}$$

where n is the observation count in a defined interval and tt_i is the travel time of ith observation.

4.3. Travel time prediction

4.3.1. k-Nearest Neighbor (k-NN) method

Travel time at any timestamp is related to the travel time of its close temporal proximity. Therefore, DTT is modeled by the nearest ATT.

$$DTT_t = ATT_t + \Delta tt_t \tag{4}$$

where Δtt_t = predicted difference of ATT and DTT at a time-step t.

Travel time difference, Δtt_t is predicted by the distance weighted k-NN method using historic daily data:

$$\Delta tt_t = \frac{\sum_d w_{d.t}\Delta tt_{d.t}}{\sum_d w_{d.t}} \tag{5}$$

where $\Delta tt_{d.t}$ is the difference between ATT and DTT on a historic day, d at a time-step, t and $w_{d.t}$ is the corresponding weight which is the measure of the similarity between two traffic patterns: the traffic pattern of the present day and of

$Initialize, F_0(x) = \bar{y}$

$for\ m = 1\ to\ M\ do$:

$\tilde{y}_i = y_i - F_{m-1}(x_i),\ i = 1, N$

$(\rho_m, \alpha_m) = \arg\min_{\rho,\alpha} \sum_{i=1}^{N} [\tilde{y}_i - \rho h(x_i; \alpha)]^2$

$F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$

$endfor$

Output the final regression function $F_m(x)$.

Fig. 3. The least square regression boost algorithm [19].

the historic day, d. This similarity is the reciprocal of the measure of the variation between those two traffic patterns. This variation is measured by the Euclidian squared distance of two n-dimensional vectors representing the latest travel times (ATT of n-steps) from the present day, p and historic day, d: $[ATT_t ATT_{t-1} ATT_{t-2} \ldots\ldots\ldots ATT_{t-n+1}]_p$ and $[ATT_t ATT_{t-1} ATT_{t-2} \ldots\ldots\ldots ATT_{t-n+1}]_d$. Since, the n-steps should be the steps that have the most significant impact on the current step t to reflect the traffic pattern, the determination of n is a heuristic approach. In this study, travel time was predicted for $n = 1, 2, 3, \ldots, 10$.

4.3.2. Boosting: least square regression (LSBoost) method

LSBoost is a least square regression boost approach that fits regression ensembles to minimize mean-squared error. At each step, a new learner is fitted to the difference between the observed response and the aggregated prediction of all learners grown previously. The following figure represents the algorithm (see Fig. 3).

4.3.3. Kalman filter method

ATT is chosen for observation, as it is the observation nearest the DTT to be predicted. After rearranging Eq. (4) as $ATT_t = DTT_t - \Delta tt_t$ and treating $-\Delta tt_t$ as the observation noise (v_t), observation equation shows a linear relationship between ATT and DTT in Eq. (6):

$$ATT_t = DTT_t + v_t \tag{6}$$

where, v denotes the observation noise. This equation is equivalent to a standard KF observation equation such as $z_t = Hx_t + v_t$ where z and x represent observation and state variables (or ATT and DTT in this case) respectively. The observation matrix, H is assumed to be 1.

The current traffic condition is more correlated with close conditions than it is with distant conditions. The proposed state–space equation is:

$$DTT_t = \Phi_{t-1} DTT_{t-1} + w_t. \tag{7}$$

And the transition function, Φ_{t-1}:

$$\Phi_{t-1} = {}^{DTT_{t-1}}/_{DTT_{t-2}} \tag{8}$$

where w denotes the state–space noise. The proposed state–space equation is similar to a standard KF state–space model, $x_t = \Phi_{t-1} x_{t-1} + w_t$ where x, the state variable, is replaced with the DTT.

The proposed KF model based on Welch and Bishop [53] is described below (see Fig. 4).

The most recent DTT ($t - 1\ and\ t - 2\ steps$) is unavailable until the vehicles have finished traveling the route; therefore, the DTT from the latest and same historical day of week at $t - 1\ and\ t - 2\ steps$ are used to estimate Φ_t. w and v are assumed to be independent of each other and follow the normal probability distributions: $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$.

A priori ($\widehat{DTT}_t^-$), according to its definition, should be equal to a corresponding DTT. Therefore, the differences between a priori (i.e. state–space projection) and DTT are considered the state–space noise. The state–space noise (w_t) is measured by:

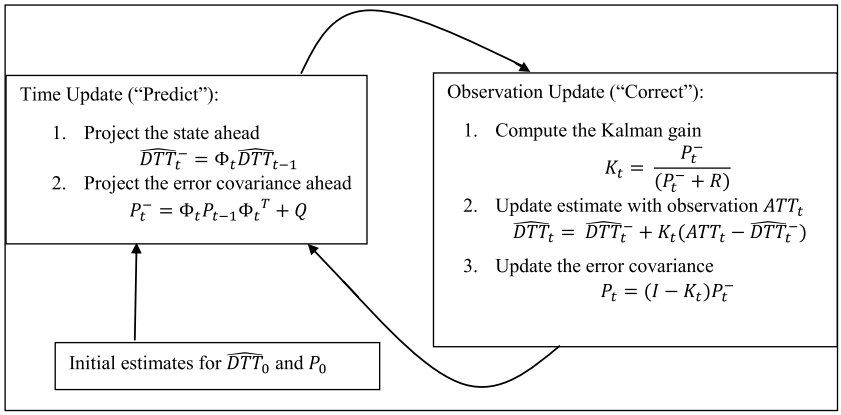

Fig. 4. KF model.

$$w_t = DTT_t - \widehat{DTT}_t^- \tag{9}$$

where $\widehat{DTT}_t^-$ = Corresponding a priori of DTT at time t.

The observation – after the noise is removed – should be equal to the predicted DTT or a posterior, according to Eq. (6). Therefore, the difference between observation and predicted time (i.e. DTT–ATT) is considered as the observation noise. The observation noise (v_t) at time t can be expressed as:

$$v_t = DTT_t - ATT_t. \tag{10}$$

Considering computational simplicity and the availability of sufficient data, noise from historic all-days (instead of same-days) was used to estimate the noise covariance. Assumptions regarding the temporal characteristics of noise can be categorized into three types:

(a) Steady Noise (SN): Regardless of time of day, noise is assumed to be the same for a day and estimated from the complete training dataset.
(b) Contextual Noise (CN): Noise is assumed to vary by traffic state (free-flow, delay, recurrent, and non-recurrent congestions). Therefore, the entire dataset is divided into four subsets – free-flow, delay, recurrent, and non-recurrent congestions – based on the travel time. Four covariance matrices (Q_{ff}, Q_{dl}, Q_{rc} and Q_{nrc}) are estimated using these sub-datasets.
(c) Time-varying Noise (TVN): Noise is assumed to vary with every time step of prediction; hence, the covariance matrix (Q_t) is estimated by the noise of a training dataset at time t. In other words, covariance matrices are estimated by splitting the entire dataset into 1,440 subsets for 1,440 intervals of a day. Therefore, $Q = \{Q_1, Q_2, \ldots\ldots Q_{1440}\}$ depending on tth interval of a day.

4.4. Prediction performance evaluation criteria

The mean absolute error (MAE), mean absolute percentage error (MAPE) and root-mean-square error (RMSE) were applied to evaluate the performance of travel time prediction methods. In general, lower value of the MAE, MAPE and RMSE indicates the superiority in prediction. However, a lower MAE, MAPE and RMSE for a corridor that experiences free-flow condition, not necessarily prove that the prediction method is superior. Since travel time of such a corridor mostly remains unchanged for a short time interval, a naïve method (e.g. using current travel time as the 5-min ahead prediction), as opposed to an advanced method, might work best. Therefore, it is important to quantify the performance of the naïve method and compare the results with the results of advance methods for better understanding. The comparison helps to quantify the overall improvements made by an advance method. In this study, the MAE, MAPE and RMSE from the prediction results of a naïve method, considering that using ATT as the prediction of DTT is a naïve method, were termed as Actual MAE, Actual MAPE and Actual RMSE (i.e. AMAE, AMAPE and ARMSE) respectively.

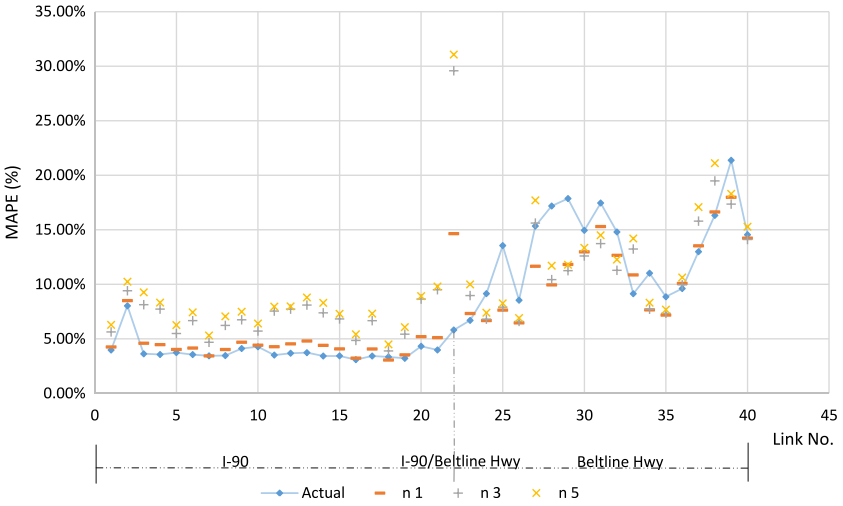

Fig. 5. MAPE of k-NN model at each link vs. actual gap.

5. Results & discussion

The complete dataset in this study was divided into two sets: Training and Validation. Twenty-eight of forty-seven days' worth of data was used for the training dataset, and the rest of the data was used as the validation dataset. It was more appropriate to use link speed data as opposed to link travel time due to the variability in link lengths. The prediction performance index was utilized to perceive global performance (i.e. the performance of the entire network). The MAPE was calculated based on the travel time dataset in order to discern the local performance at each link. Quantifying improvement is impossible without knowing the AMAPE or actual lag/gap, considering that using ATT as the prediction of DTT is a naïve method. AMAPE (i.e. MAPE of ATT) was used as the benchmark for MAPEs generated by other methods.

The k-NN method showed different prediction performance of different values of n ($n = 1, 2, \ldots, 10$). Fig. 5 represents the local (i.e. each link) performance of the k-NN model for validation dataset when $n = 1$, 3 and 5. When evaluating criteria of local performance, the MAPE of the prediction for each link should be smaller than that of the actual lag/gap (AMAPE or the benchmark).

In Fig. 5, MAPEs of travel times predicted by the k-NN method using $n = 1$, 3 and 5 are compared to the actual lag/gap. The local performance is unacceptable since the MAPEs of more than half of the links are greater than the actual lag/gap. Performance shows a decreasing trend (i.e. MAPE increases) with the increase of n. Moreover, the global performance measured by the MAPE over entire route is poor. The overall MAPEs are higher than the actual gap or AMAPE (6.70%), suggesting the lack of repeating traffic condition over the entire period. Therefore, k-NN is not an appropriate method to predict DTT from ATT when the traffic conditions in the training dataset are not similar to the conditions in the validation dataset.

Two new variables, day of the week and time of the day, were included besides ATT in the LSBoost application. In addition, the latest ATTs (2, 3, 5, 10 min etc.) were combined with the current ATT to create a new variable. Similar prediction performances were observed for different numbers of newly added ATT columns ($n = 2$, 3, 5 and 10). In other words, this algorithm shows limited sensitivity to the size of the combined ATTs. For visual clarity, only two selected results with the actual gap are shown in Fig. 6.

In Fig. 6, MAPEs of travel times predicted by LSBoost using two, three and five previous-step ATTs ($n = 2$, 3 and 5 respectively) with current ATT are compared to the actual lag/gap. The MAPEs and AMAPEs of different links on I-90 have very similar values which indicate the shortcoming of this method on a freeway that experiences free-flow. The performance of LSBoost prediction is notably well for the Beltline Highway that experiences congestion. The link (22nd) that connects I-90 and Beltline corridor performs very poorly showing MAPE much higher than that of AMAPE. Therefore, LSBoost is not recommended at this point to predict DTT from ATT for freeways.

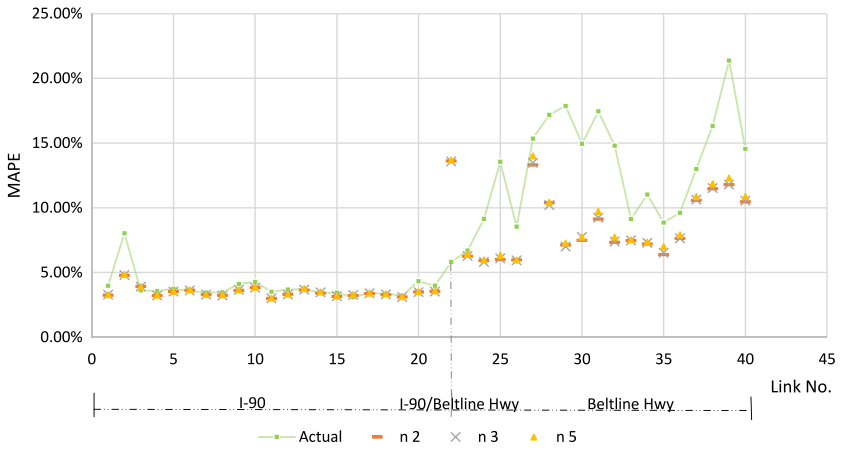

Fig. 6. MAPE of LSBoost at each link vs. actual gap.

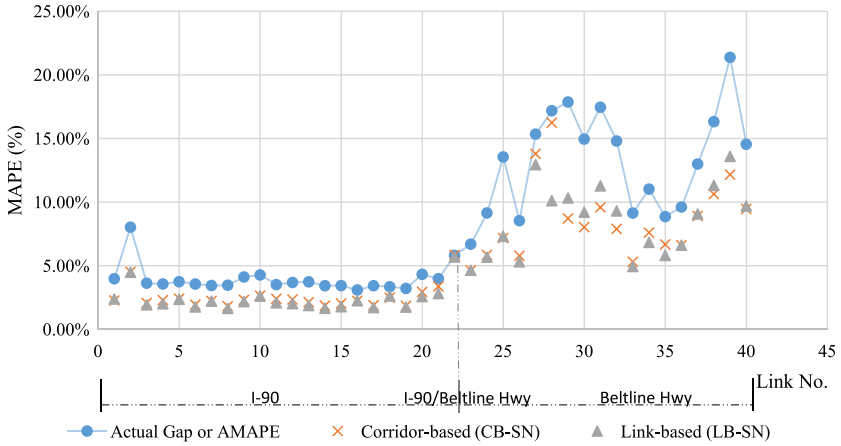

Fig. 7. MAPE of KF at each link for steady noise assumption vs. actual gap.

Spatial noise characteristics for KF models were assumed in this study by considering that (a) noise of different links in a particular corridor can have similar characteristics, and (b) noise of different links, regardless of corridor, can have different characteristics. Previously, three categorical assumptions regarding temporal characteristics of noise have been discussed; therefore, the output of the KF model would be affected by six different estimation procedures of noise covariance regarding the spatial–temporal characteristics of noise. The six methods are: corridor-based steady noise (CB-SN), contextual noise (CB-CN), time-varying noise (CB-TVN), link-based steady noise (LB-SN), contextual noise (LB-CN), and time-varying noise (LB-TVN).

Appropriate noise characteristics of KF should be determined through the evaluation of local performance. Figs. 7, 8, and 9 represent the local (i.e. each link) performance of the KF model for validation dataset with different noise assumptions. When evaluating criteria of local performance, the MAPE of the prediction for each link should be smaller than that of the actual lag/gap (AMAPE or the benchmark). In Fig. 7, MAPEs of prediction by the KF model using CB-SN and LB-SN are compared to the actual lag/gap. The prediction performances of the KF model using both CB-SN and LB-SN for each individual link are acceptable, as no link shows a MAPE greater than the actual lag/gap.

In Fig. 8, MAPEs of travel times predicted by the KF model using CB-CN and LB-CN are compared to the actual lag/gap. It is clear that the KF model with both CB-CN and LB-CN has a few links' MAPE greater than AMAPE. In general, the context-based noise assumption is supposed to perform better for the corridor that experiences congestion.

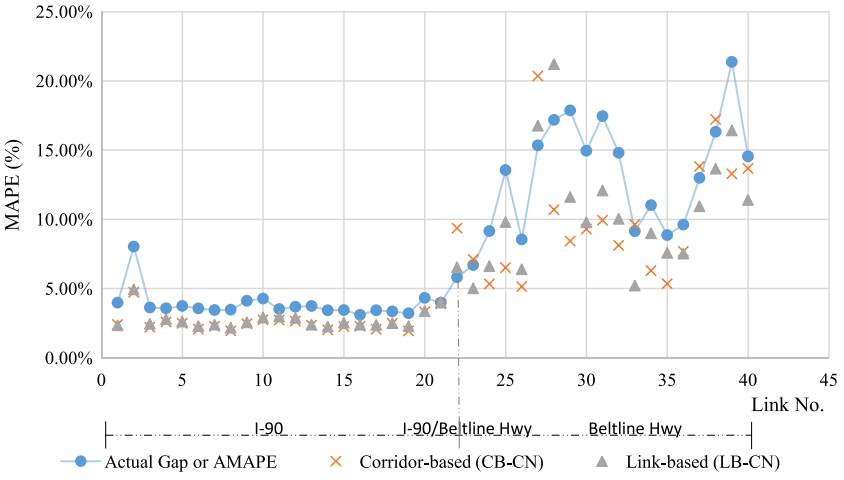

Fig. 8. MAPE of KF at each link for contextual noise assumption vs. actual gap.

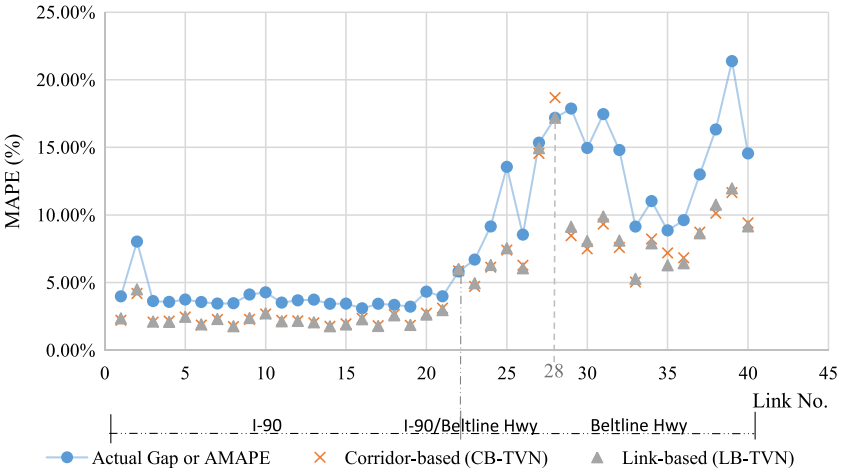

Fig. 9. MAPE of KF at each link for time varying noise assumption vs. actual gap.

The poor performance could be due to the stability of travel time resulting from the saturated traffic flow rate under congestions. At a saturated flow, the variations between ATT and DTT become similar to the variations in free flow or delay conditions.

In Fig. 9, MAPEs of travel times predicted by the KF model using CB-TVN and LB-TVN are compared to the actual lag/gap. The assumption of CB-TVN is invalid for a corridor with some links that experience traffic congestion. Despite the improvement from contextual noise assumption, Fig. 9 shows that the noise homogeneity assumption of CB-TVN is violated at Link 28, as its performance exceeds the actual lag/gap (AMAPE).

The above discussion provides a comprehensive description of local/link performance of different noise assumptions in a KF model. The unambiguous analyses reflect the suitability of CB-SN, LB-SN and LB-TVN. Table 1 shows the global performance of prediction expressed by the MAE and RMSE of speed data calculated from travel time, which reaffirms the most appropriate noise assumption for the dataset is LB-SN.

The KF model with LB-SN assumption is more accurate and computational efficient. For instance, the run time of KF with corridor-based noise assumptions is approximately 15 min, whereas link-based assumptions take only a minute or two. Corridor-based noise homogeneity assumption calls for extra processing of data since Bluetooth data is collected over each link; this way, noise covariance can be estimated over the entire corridor. This extra processing

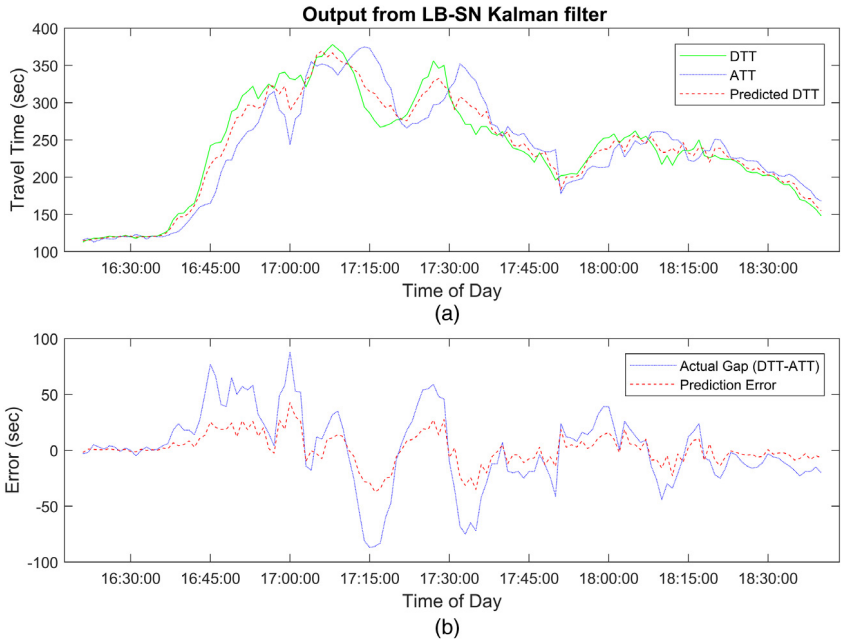

Fig. 10. Prediction performance between free flow and congested conditions.

Table 1
Overall (global) performance of KF model for selected noise assumptions.

Noise assumption	Training dataset		Validation dataset	
	MAE	RMSE	MAE	RMSE
CB-SN	2.25	6.31	2.41	6.75
LB-SN	2.17	5.96	2.31	6.33
LB-TVN	2.15	6.78	2.45	7.47

increases the run time significantly. The link-based model is the most suitable for an on-line application. KF with LB-SN was selected to predict DTT from ATT.

The MAPEs of the training and validation datasets using LB-SN KF are 4.27% and 4.53%, respectively, whereas, the actual lags/gaps (i.e. AMAPEs) are 6.43% and 6.70%, respectively, based on travel time dataset. Improvements are significant when the smaller actual lag/gap is considered, including a 50% reduction in AMAPE in some links. When the KF model with LB-SN assumption is applied, the validation dataset shows that a 40%–50% gap between ATT and DTT is minimized in different links of the I-90 corridor, and a nearly 30%–40% gap is minimized in different links of the Beltline Highway corridor.

Moreover, the KF model with the LB-SN assumption has a superior prediction performance in cases of traffic state transition (e.g. onset and end of congestion). Fig. 10(a) represents the DTT, ATT, and predicted travel time, and Fig. 10(b) demonstrates the actual lag/gap and prediction error corresponding to Fig. 10(a).

Fig. 10 clearly depicts that the ATT, DTT, and predicted travel time are almost equal at free flow conditions (before 16:30:00). Actual and prediction errors are negligible after 18:15:00 when congestion is stable. Fig. 10(a) Box A shows the onset of congestion where the actual lag/gap is more than 50 s, and Fig. 10(b) shows prediction lag is less than 20 s. Box B in Fig. 10(a) shows the end of the congestion situation where the actual and prediction lags show little difference (less than 20 s). However, at 17:15:00, the mid-point of the end of congestion, the ATT is off by 100 s from DTT while the prediction error is around 30 s. Despite having a moderate improvement (around 40%) according to MAPE (global performance), the prediction shows excellent improvement (around 70%) in cases such as that in box B where there is a state transition. Since such cases cover shorter time periods compared to the complete

study period, the overall performance (local or global) indices are unable to represent the robustness of the prediction algorithm. Therefore, the selected noise assumption based on the KF model is capable of predicting travel time from the travelers' point of interest (the onset and end of congestion rather than free-flow or stable congested conditions).

To conclude, to predict the current traffic state as well as travel time, k-NN and LSBoost utilize the full dataset whereas the KF method uses most recent dataset. Intrinsically, the heteroscedasticity of traffic condition captured by the most recent data is more resemble to the current. Perhaps, this gives KF advantages over other methods. Moreover, the noise modeling capability offers additional benefits such as the selection of a subset of data to represent the outstanding heteroscedasticity that may not be adequately represented by the most recent data used in the state projection. Hence, prediction performances of the KF method with noise assumptions are more consistent than that of the k-NN and LSBoost in context of different roadway links.

6. Conclusion

ATT (or $\boldsymbol{t_{expn}}$) is the most available form of travel time, but DTT (or $\boldsymbol{t_{expt}}$) is the most desirable. The k-NN, LSBoost and KF algorithms were used to predict DTT and thus assist motorists by providing a more accurate and reliable travel time. Overall, KF outperformed other two methods. Although ATT and DTT differ slightly when the flow of traffic is stable, the variation becomes significant when the traffic state is in transition (e.g. moving from unstable to stable or vice versa). The KF algorithm with steady noise assumption captured a state of transition property accurately and provided an excellent prediction. KF is fast for link-based applications, making it desirable for data sources that contain route travel time split into shorter links (e.g. loop detectors/Bluetooth data). Although KF is applied (by default) to each link that is isolated as a different model, it demonstrates a higher level of accuracy and faster speed due to its flexibility, simplicity and compatibility with data characteristics. The application was demonstrated during peak periods on freeways covering two corridors — one with fewer transitions in traffic state and the other with frequent transitions. Diversity in noise assumptions showed negligible impact on the former, while steady noise assumption showed better performance on the latter. Steady noise (SN) refers to a fixed covariance estimated from the complete training dataset, which indicates that the KF model performs better with the generalized noise assumption. Hence, the KF with LB-SN assumption was preferred over other assumptions. Better performance was observed during a time when transitions in traffic state occurred more frequently.

Since arterial highways are supposed to exhibit more state transitions, future research should examine the performance of predicting arterial highway travel time to test the robustness of this method. Furthermore, as firstly pointed out in Guo et al. [21], vehicular traffic condition series is heteroscedastic in nature, new research is needed to investigate the second-order moment of travel time, i.e., the uncertainty of the time series data of travel time. Future research can substantiate and expand on the pioneering research methodologies in the travel time uncertainty domain [6,22,23,28].

References

[1] H. Al-Deek, M.P. D'Angelo, M. Wang, Travel time prediction with non-linear time series, in: Fifth International Conference on Applications of Advanced Technologies in Transportation Engineering, 1998.
[2] C. Bachmann, B. Abdulhai, M.J. Roorda, B. Moshiri, A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling, Transp. Res. C 26 (2013) 33–48.
[3] Z. Barutçuoğlu, E. Alpaydın, A comparison of model aggregation methods for regression, in: Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP, 2003, Springer, 2003.
[4] A. Bhaskar, E. Chung, A.G. Dumont, Fusing loop detector and probe vehicle data to estimate travel time statistics on signalized urban networks, Comput.-Aided Civ. Infrastruct. Eng. 26 (2011) 433–450.
[5] M. Blogg, C. Semler, M. Hingorani, R. Troutbeck, Travel time and origin–destination data collection using Bluetooth MAC address readers, Australas. Transp. Res. Forum (2010).
[6] J. Cao, R. Li, W. Huang, J. Guo, Y. Wei, Traffic network equilibrium problems with demands uncertainty and capacity constraints of arcs by scalarization approaches, Sci. China Technol. Sci. (2017). http://dx.doi.org/10.1007/s11431-017-9172-4. (in press).
[7] C. Carrion, D. Levinson, Value of travel time reliability: A review of current evidence, Transp. Res. A 46 (2012) 720–741.
[8] H.B. Celikoglu, Flow-based freeway travel-time estimation: A comparative evaluation within dynamic path loading, IEEE Trans. Intell. Transp. Syst. 14 (2013) 772–781.
[9] M. Chen, S. Chien, Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based, Transp. Res. Rec.: J. Transp. Res. Board (2001) 157–161.
[10] H. Chen, H.A. Rakha, Real-time travel time prediction using particle filtering with a non-explicit state-transition model, Transp. Res. C 43 (2014) 112–126.

[11] S.I.-J. Chien, C.M. Kuchipudi, Dynamic travel time prediction with real-time and historic data, J. Transp. Eng. 129 (2003) 608–616.
[12] S. Clark, S. Grant-Muller, H. Chen, Cleaning of matched license plate data, Transp. Res. Rec.: J. Transp. Res. Board (2002) 1–7.
[13] J.J. Commandeur, S.J. Koopman, An Introduction to State Space Time Series Analysis, OUP Oxford, 2007.
[14] D. Darwish, Assessment of offline digital signature recognition classification techniques, Int. J. Comput. Netw. Commun. Secur. (2013) 1.
[15] F. Dion, H. Rakha, Estimating dynamic roadway travel times using automatic vehicle identification data for low sampling rates, Transp. Res. B 40 (2006) 745–766.
[16] J. Durbin, S.J. Koopman, Time Series Analysis by State Space Methods, Oxford University Press, 2012.
[17] M. Elhenawy, I. El-Shawarby, H. Rakha, Modeling the Perception Reaction Time and Deceleration Level for Different Surface Conditions using Machine Learning Techniques, in: Advances in Applied Digital Human Modeling and Simulation, Springer, 2017.
[18] X. Fei, C.-C. Lu, K. Liu, A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction, Transp. Res. C 19 (2011) 1306–1318.
[19] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. (2001) 1189–1232.
[20] B. Ghosh, M.T. Asif, J. Dauwels, W. Cai, H. Guo, U. Fastenrath, Predicting the duration of non-recurring road incidents by cluster-specific models, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, (ITSC), IEEE, 2016, pp. 1522–1527.
[21] J. Guo, W. Huang, B.M. Williams, Integrated heteroscedasticity test for vehicular traffic condition series, ASCE J. Transp. Eng. 138 (9) (2012) 1161–1170.
[22] J. Guo, W. Huang, B.M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transp. Res. C 43 (2014) 50–64.
[23] J. Guo, Z. Liu, W. Huang, Y. Wei, J. Cao, Short term traffic flow prediction using fuzzy information granulation approach under different time intervals, IET Intel. Transport Syst. (2017) (in press).
[24] A. Haghani, M. Hamedi, K. Sadabadi, S. Young, P. Tarnoff, Data collection of freeway travel time ground truth with bluetooth sensors, Transp. Res. Rec.: J. Transp. Res. Board (2010) 60–68.
[25] M.M. Hamed, H.R. Al-Masaeid, Z.M.B. Said, Short-term prediction of traffic volume in urban arterials, J. Transp. Eng. 121 (1995) 249–254.
[26] J.D. Hamilton, Time Series Analysis, Princeton University Press, Princeton, 1994.
[27] A.C. Harvey, Forecasting, Structural Time Series Models and the Kalman Filter, Cambridge University Press, 1990.
[28] W. Huang, W. Jia, J. Guo, B.M. Williams, G. Shi, Y. Wei, J. Cao, Real time prediction of seasonal heteroscedasticity in vehicular traffic flow series, IEEE Trans. Intell. Transp. Syst. (2017) (in press).
[29] E. Jenelius, H.N. Koutsopoulos, Urban network travel time prediction based on a probabilistic principal component analysis model of probe data, IEEE Trans. Intell. Transp. Syst. (2017).
[30] W. Jiang, Is regularization unnecessary for boosting? in: AISTATS, 2001.
[31] W. Jiang, Some theoretical aspects of boosting in the presence of noisy data, in: Proceedings of the Eighteenth International Conference on Machine Learning, Citeseer, 2001.
[32] F. Jiao, J. Xu, L. Yu, D. Schuurmans, Protein fold recognition using the gradient boost algorithm, in: Computational Systems Bioinformatics Conference, 2006, pp. 43–53.
[33] A. Khattak, A. Polydoropoulou, M. Ben-Akiva, Modeling revealed and stated pretrip travel response to advanced traveler information systems, Transp. Res. Rec.: J. Transp. Res. Board (1996) 46–54.
[34] J. Kim, J. Rho, D. Park, On-line estimation of departure time-based link travel times from spatial detection system, Int. J. Urban Sci. 13 (2009) 63–80.
[35] H. Liu, Travel Time Prediction for Urban Networks, Delft University of Technology, TU Delft, 2008.
[36] H. Liu, H. Van Zuylen, H. Van Lint, M. Salomons, Predicting urban arterial travel time with state–space neural networks and Kalman filters, Transp. Res. Rec.: J. Transp. Res. Board (2006) 99–108.
[37] P.S. Maybeck, The Kalman filter: An introduction to concepts, in: Autonomous Robot Vehicles, Springer, 1990.
[38] S. Moghaddam, B. Hellinga, Quantifying measurement error in arterial travel times measured by bluetooth detectors, Transp. Res. Rec.: J. Transp. Res. Board (2013) 111–122.
[39] S. Moghaddam, B. Hellinga, Algorithm for detecting outliers in Bluetooth data in real time, Transp. Res. Rec.: J. Transp. Res. Board (2014) 129–139.
[40] J.A. Moscoso-López, I. Turias, M.J. Jiménez-Come, J.J. Ruiz-Aguilar, M.D.M. Cerbán, A two-stage forecasting approach for short-term intermodal freight prediction, Int. Trans. Oper. Res. (2016).
[41] J. Myung, D.-K. Kim, S.-Y. Kho, C.-H. Park, Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system, Transp. Res. Rec.: J. Transp. Res. Board (2011) 51–59.
[42] C. Nanthawichit, T. Nakatsuji, H. Suzuki, Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway, Transp. Res. Rec.: J. Transp. Res. Board (2003) 49–59.
[43] T. Oda, An algorithm for prediction of travel time using vehicle sensor data, in: Third International Conference on Road Traffic Control, 1990, IET, 1990, pp. 40–44.
[44] S.D. Oduro, S. Metia, H. Duc, G. Hong, H.A. Q, Multivariate adaptive regression splines models for vehicular emission prediction, Vis. Eng. 3 (2015) 13.
[45] D.D. Puckett, M.J. Vickich, Bluetooth®-based travel time/speed measuring systems development, 2010.
[46] J. Rice, E. Van Zwet, A simple and effective method for predicting travel times on freeways, IEEE Trans. Intell. Transp. Syst. 5 (2004) 200–207.
[47] A. Skabardonis, N. Geroliminis, Real-time estimation of travel times on signalized arterials, 2005.
[48] B.L. Smith, B.M. Williams, R.K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, Transp. Res. C 10 (2002) 303–321.

[49] A. Sumalee, T. Pan, R. Zhong, N. Uno, N. Indra-Payoong, Dynamic stochastic journey time estimation and reliability analysis using stochastic cell transmission model: Algorithm and case studies, Transp. Res. C 35 (2013) 263–285.
[50] A. Toppen, K. Wunderlich, Travel time data collection for measurement of advanced traveler information systems accuracy, in: Mitretek Systems, 2003.
[51] C. Van Hinsbergen, J. Van Lint, F. Sanders, Short term traffic prediction models, in: Proceedings of the 14th World Congress on Intelligent Transport Systems, ITS, Held Beijing, October 2007, 2007.
[52] E.I. Vlahogianni, M.G. Karlaftis, J.C. Golias, Short-term traffic forecasting: Where we are and where we're going, Transp. Res. C 43 (2014) 3–19.
[53] G. Welch, G. Bishop, An Introduction to the Kalman Filter, Department of Computer Science, University of North Carolina, Chapel Hill, NC, 2006, unpublished manuscript.
[54] C.-H. Wu, J.-M. Ho, D.-T. Lee, Travel-time prediction with support vector regression, IEEE Trans. Intell. Transp. Syst. 5 (2004) 276–281.
[55] Y. Xiao, S. Qom, M. Hadi, H. Al-Deek, Use of data from point detectors and automatic vehicle identification to compare instantaneous and experienced travel times, Transp. Res. Rec.: J. Transp. Res. Board (2014) 95–104.
[56] J.-S. Yang, Travel time prediction using the GPS test vehicle and Kalman filtering techniques, in: American Control Conference, 2005. Proceedings of the 2005, IEEE, 2005, pp. 2128–2133.
[57] X. Zhan, S. Hasan, S.V. Ukkusuri, C. Kamga, Urban link travel time estimation using large-scale taxi data with partial information, Transp. Res. C 33 (2013) 37–49.
[58] Y. Zhang, A. Haghani, X. Zeng, Component GARCH models to account for seasonal patterns and uncertainties in travel-time prediction, IEEE Trans. Intell. Transp. Syst. 16 (2015) 719–729.
[59] X. Zhang, E. Onieva, A. Perallos, E. Osaba, V.C. Lee, Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction, Transp. Res. C 43 (2014) 127–142.
[60] F. Zheng, H. Van Zuylen, Urban link travel time estimation based on sparse probe vehicle data, Transp. Res. C 31 (2013) 145–157.
[61] R. Zhong, J. Luo, H. Cai, A. Sumalee, F. Yuan, A.H. Chow, Forecasting journey time distribution with consideration to abnormal traffic conditions, Transp. Res. C 85 (2017) 292–311.
[62] Y. Zou, X. Zhu, Y. Zhang, X. Zeng, A space–time diurnal method for short-term freeway travel time prediction, Transp. Res. C 43 (2014) 33–49.