COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **31** (2016) 777–793

Development of a Data-Driven Framework for Real-Time Travel Time Prediction

Sehyun Tak, Sunghoon Kim, Simon Oh & Hwasoo Yeo*

Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea

Abstract: *Travel time prediction is one of the most important components in Intelligent Transportation Systems implementation. Various related techniques have been developed, but the efforts for improving the applicability of long-term prediction in a real-time manner have been lacking. Existing methods do not fully utilize the advantages of the state-of-the-art cloud system and large amount of data due to computation issues. We propose a new prediction framework for real-time travel time services in the cloud system. A distinctive feature is that the prediction is done with the entire data of a road section to stably and accurately produce the long-term (at least 6-hour prediction horizon) predicted value. Another distinctive feature is that the framework uses a hierarchical pattern matching called Multilevel k-nearest neighbor (Mk-NN) method which is compared with the conventional k-NN method and Nearest Historical average method. The results show that the method can more accurately and robustly predict the long-term travel time with shorter computation time.*

1 INTRODUCTION

Future traffic information through travel time prediction will provide benefits to road users and mitigate congestion, as it takes a role as a prior knowledge contributing to make informed decisions on the route choice and the optimal departure time. Moreover, the reliable future information is helpful in developing appropriate control strategies in Traffic Management Systems (TMS). The existing prediction techniques are comprehensively reviewed by recent studies (van Hinsbergen and van Lint, 2007; Oh et al., 2001; Vlahogianni et al., 2004). Traffic prediction using recent data-driven approaches can be classified into parametric and nonparametric approaches. Parametric models find the model parameters from a finite dimensional space, and they are known to perform in a quick manner with robust theoretical grounds. However, site specific property on coefficients and limited performance on large prediction range have been pointed out as limitations (Ishak, 2002; Kwon et al., 2000; Williams and Hoel, 2003; Zhang and Rice, 2003). Neural network methods, which are nonparametric, are known to have high performances in various experimental settings (Dharia and Adeli, 2003; Dougherty and Mark, 1997; Ghosh-Dastidar and Adeli, 2006; Jiang and Adeli, 2005; Van Lint et al., 2005; Van Lint, 2006; Park and Rilett, 1999). However, complex training with burdensome computation is an obstacle. Another nonparametric method is nearest neighbor-based approach that finds similar historic records based on the designed feature vector. This approach is known to be feasible when the prediction is conducted with the sufficient historical records (Clark, 2003; Davis and Nihan, 1991). This method seems to be promising with the improvement in computing capability and big data availability.

With the improvements in traffic data acquisition technology, the use of pattern-searching methods have increased due to their innate ability to predict complex nonlinear relationships. One of the representative methods is k-nearest neighbor (k-NN) method that finds the k most similar historic records based on the designed feature vectors with the assumption that the knowledge about the relationship lies in the data (Bajwa et al., 2005; Nair et al. 2001; Sun et al., 2003; Tak et al., 2014a, b). One merit of this is that there is no distributional assumption on model parameters, which can avoid site-specific model structure. However, as the pattern recognition process requires various computer resources, the searching process space needs to be improved for real-time application. Moreover, extension of prediction horizon is required as the previous models predict relatively short-term future

*To whom correspondence should be addressed. E-mail: *hwasoo@gmail.com*.

DOI: 10.1111/mice.12205

(5–90 minutes) for small spatial scope (2.2–77 km) using the linear regression (Fei et al., 2011; Kwon et al., 2000; Rice and Van Zwet, 2004), ARIMA (Ishak, 2002; Oda, 1990; Saito and Watanabe, 1995), Kalman filter (Chien and Kuchipudi, 2003; Park and Rilett, 1999), ANN (Dia, 2001; Van Lint, 2006), and k-NN based prediction (Bajwa et al., 2005; Clark, 2003; Nair et al., 2001; Sun et al., 2003). Although some previous studies have predicted the long-term future with heuristic approaches (Chrobok, 2005; Röhr et al., 1996; Siegener and Schmitt, 1980), there is still a dearth of a robust real-time prediction framework for large-scale highway network and longer prediction horizon due to the incapability of dealing with the variability of daily traffic. To overcome this limitation, the differences in traffic patterns across time-of-day (peak and nonpeak hour) and type-of-day (weekdays and weekend) should be considered. For example, unlike weekdays, the congestion occurrence time and level of congestion differ by different weekends (Castillo and Nogal, 2012). So, the accuracy of heuristics-based methods would be lower when predicting the weekend traffic particularly, because it yields low likelihood to find similar posterior distribution (Bajwa et al., 2005).

For real-time application, several on-line prediction frameworks are proposed using model-based approaches (Oh et al., 2015). Some frameworks like TOPL (Chow et al., 2008) and BOSS-METANET (Papageorgiou and Papamichail, 2010) use macroscopic traffic flow models, DynaSMART-X (Mahmassani et al., 2005) is mesoscopic, and AIMSUN On-line (Casas et al., 2013) is microscopic. Also, OLSIM (Chrobok et al., 2004) uses a cellular automaton-based model. Meanwhile, there are also some efforts to increase applicability of data-driven approaches on real-time service. Some researchers provided system framework integrating several modules of data collection, preprocessing, and prediction (Van Lint, 2006; Yu et al., 2008). To deal with the data feeding in real-time, time-varying coefficients are incorporated in the linear regression (Zhang and Rice, 2003). In a more advanced way, Bayesian approaches are proposed to efficiently update the model for the unexpected events (Fei et al., 2011; van Hinsbergen and van Lint, 2008). Note that incident detection algorithms also have been developed using wavelet functions (Karim and Adeli, 2002, 2003b). Van Lint et al. (2005) developed on-line prediction system based on state space neural network, and later they improved the state space neural network-based system by incorporating extended Kalman filter for the efficient training process (van Lint, 2008).

In addition, advanced methods have been developed, such as hybrid version of genetic algorithm-simulated annealing algorithm in SVR (Hong et al., 2011; Li et al., 2013) and Beta-Gaussian Bayesian Networks (Castillo and Nogal, 2012). Their methods showed potential application in real-time service with efficient computational costs in dealing with high number of components. Missing data also has to be deeply considered in practical implementation (Chen et al., 2001; Wen et al., 2005).

To increase the practicability of real-time on-line implementation that can guarantee the accuracy and computational efficiency with 6-hour prediction horizon, we develop a long-term prediction framework that consists of several levels. The framework consists of three steps: (1) data preprocessing, (2) traffic pattern matching, and (3) building predicted data. In data preprocessing, we specify road sections into certain levels (global and local road sections) and rearrange raw data sets such as toll collection system (TCS), vehicle detection system (VDS), and dedicated short-range communication (DSRC). In traffic pattern matching, we propose a hierarchical structure of pattern matching, which we call multilevel k-nearest neighbor (Mk-NN) method. The method carries out traffic pattern matching processes for global and local sections sequentially, and such processes find the historical dates that have similar patterns. Then, in building predicted data, predictions are carried out based on the assumption that the date having the most similar traffic pattern from certain time (e.g., 4 hours ago) until the current time (the real-time) will indicate the most similar traffic pattern in the future. In most of the data-driven methods, such assumption is typically made. They assume that the traffic pattern that is similar with the current traffic pattern exists in the historical data. Here, there may be an issue on the minimum required days of historical data, because it is a major factor that affects the performance of prediction. It has not yet been thoroughly analyzed on the minimum required days of historical data, but 300 or more days of historical data derive good enough results (Tak et al., 2016). So, we use historical data sets that are more than 300 days.

The entire prediction framework uses k-NN technique as the base method for prediction. We use such technique, because the nearest neighbor-based methods have shown good performances for short-term predictions (Clark, 2003; Davis and Nihan, 1991; Smith et al., 2002), and because the heuristic mechanism is incorporated in the methods, k-NN technique seems to be appropriate for long-term prediction as well. The major demerit of the method is the long computation time for searching the historical data sets. However, by modifying the feature vector in road section basis, the computation time would be significantly reduced, and with the advanced computing like distributed computing, the computation time can be more reduced. The details of the prediction framework and test results are provided in the following sections. Abbreviations are listed in Table 1.

Table 1
List of abbreviations

Abbreviation	*Full description*	*Abbreviation*	*Full description*
$X^{(g)}_{TCS,in}(t)$	The tensor for the vehicle inbound data at all	W_{In}	The weight for TCS inbound data
	where $g=\{1,2,3,\ldots,GS\}$	W_{Out}	The weight for TCS outbound data
	and $t=\{1,2,3,\ldots,T\}$	W_{Flow}	The weight for VDS flow data
VDS	Vehicle Detection System	$D^{(g)}_{In,i}(t)$	The ith historical TCS inbound data of global section g at time t
GS	The total number of global sections	$D^{(g)}_{Out,i}(t)$	The ith historical TCS outbound data of global section g at time t
$N_t^{(G^{(g)})}$	The inbound value obtained by detector $G^{(g)}$ at time t	$D^{(g)}_{Flow,i}(t)$	The ith historical VDS flow data of global section g at time t
τ	The range of neighboring time intervals	$d_i^l(t)$	The local section l's distance metric representing the dissimilarity of ith historical data at time t
$X^{(g)}_{TCS,out}(t)$	The tensor for the vehicle outbound data at all tollgates within global section g	W_{Occ}	The weight for VDS occupancy data
	at time t	W_{Sp}	The weight for VDS speed data
$M_t^{(G^{(g)})}$	The outbound value obtained by detector $G^{(g)}$ at time t	W_{DSRC}	The weight for DSRC speed data
$X^{(g)}_{VDS,Flow}(t)$	The tensor for the flow data at all loop detectors within global section g at time t	$D^{(l)}_{Occ,i}(t)$	The ith historical VDS occupancy data of local section l at time t
$F_t^{(G^{(g)})}$	The flow value obtained by detector $G^{(g)}$ at time t	$D^{(l)}_{Sp,i}(t)$	The ith historical VDS speed data of local section l at time t
$X^{(l)}_{VDS,Occ}(t)$	The tensor for the occupancy data at all loop detectors within local section l at time t,	$D^{(l)}_{DSp,i}(t)$	The ith historical DSRC speed data of local section l at time t
	where $\text{l}=\{1,2,3,\ldots,\text{LS}\}$ and $t=\{1,2,3,\ldots,\text{T}\}$	$H^{(g)}_{Detector,\ i}(\text{t})$	The data health matrix of the detectors in global section g at time t
LS	The total number of local sections within in a global section	$h^{Detector,\ (G^{(g)})}_{subject}(t)$	The data health value of subject data provided by detector $G^{(g)}$ at time t
$O_t^{(L^{(l)})}$	The occupancy value obtained by detector $L^{(l)}$ at time t	$h^{Detector,\ (G^{(g)})}_{history,i}(t)$	The data health value of ith historical data provided by detector $G^{(g)}$ at time t
$X^{(l)}_{VDS,Sp}(t)$	The tensor for the speed data at all loop detectors within local section l at time t	$H^{(l)}_{Detector,\ i}(t)$	The data health matrix of the detectors in local section l at time t
$S_t^{(L^{(l)})}$	The speed value obtained by detector $L^{(l)}$ at time t	$h^{Detector,\ (L^{(l)})}_{subject}(t)$	The data health value of subject data provided by detector $L^{(l)}$ at time t
$X^{(l)}_{DSRC,Sp}(t)$	The tensor for the speed data at all RSUs of DSRC within local section l at time t	$h^{Detector,\ (L^{(l)})}_{history,i}(t)$	The data health value of ith historical data provided by detector $L^{(l)}$ at time t
$C_t^{(L^{(l)})}$	The speed value obtained by RSU $L^{(l)}$ at time t	K	The number of nearest neighbors in data pattern and k ∈ K
$X_t^{(s)}$	The feature vector for individual detector s at time interval t	d_k	The dissimilarity between subject data and kth historical data of local section l
S	The set of individual detector IDs, where s ∈ S	σ	The prediction horizon
$Z^{(s-j)}_{t-\tau}$	The traffic values (flow or speed) obtained by the detector with ID s – j at time t,	$V_k^{(l)}(t+\sigma)$	The speed of kth historical data of local section l at time $t+\sigma$
	where the spatial parameter j represents the range of the neighboring detectors	$e_t^{(l)}$	The error in local section l at time t
$d_i^g(t)$	The Global section g's distance metric representing the dissimilarity of ith historical	$V_t^{(l)}$	The actual data value in local section l at time t
	data at time t	n	The number of predicted values

2 FRAMEWORK

To provide travel time information to freeway users before they begin their travels, the speed prediction horizon has to be significantly long. For example, if the route's travel time increases due to traffic congestion, then it would require even longer prediction horizon. Therefore, to provide long-term future traffic information service to users, the prediction horizon should be longer than the maximum travel time of a certain highway route.

Thus, we propose a reliable data-driven framework for predicting freeway speed and travel time in real-time (Figure 1). The framework consists of three steps at the highest level: (1) data preprocessing, (2) traffic pattern matching, and (3) building predicted data (speed and travel time). One of the main features of the framework is that an entire highway network is specified into several sections. Such specification enables independent prediction processes by different sections, which can ease managing data quality by the specified sections. Another feature is that the traffic pattern matching is designed with a hierarchical matching strategy that uses the k-NN method. The hierarchical process improves the computation efficiency and prediction accuracy with long prediction horizon. Also, the framework uses distributed computing strategy when predicting speed and travel time, which is also related to improving the computation speed and it enables real-time service. The detailed description on the framework is provided in the following subsections.

2.1 Data preprocessing

In data preprocessing, data collected from various sources installed in highway are arranged and corrected to improve the computation efficiency and quality of traffic pattern matching for prediction. To predict the future traffic for main line in highway shown in Figure 2 (right figure), the data from TCS, DSRC, and loop detectors are used. There are tollgates at the 143 interchanges, where vehicles enter or exit the highway roads, and the number of vehicles entering or exiting the highway are counted and aggregated for every 1-hour time span. Such counted numbers are recorded and compose TCS data sets. Also, there are 3,408 loop detectors installed. The traffic information collected through these loop detectors builds VDS data sets. There are two different types of VDS data. One type of VDS data set has flow values at every 15 minutes near junction, where several highway lines are merged. The other type has average speed and occupancy values at every 5 minutes. There are also 3,408 roadside units (RSUs) installed over the route. The RSUs communicate with high-pass devices equipped in vehicles and collect travel time information on certain road sections at every 5 minutes. The collected travel time information becomes the source of DSRC data sets. For all kinds of data sets, 471 days of data are used as the historical data for traffic pattern matching.

The data preprocessing step is related to the modules K1, K2, and K5 in Figure 1. Module K1 (data preprocessor) rearranges raw data sets and transforms them into historical data sets. Module K5 (real-time data processor) also rearranges raw data sets and transforms them into real-time data sets with the same format as the historical data sets. The formats of the two data sets are the same for traffic pattern matching purposes. Module K2 (date list management) extracts information from the transformed data sets, such as data health, day type, and weather condition. The details of the data preprocessing step is provided as follows.

Before rearranging and transforming raw traffic data, we specify a highway network into several sections, as shown in Figure 2. As mentioned earlier, this is one of the main features of the proposed framework. Note that a junction (JC) refers to an intersecting point of two or more highway routes. Based on the JC locations, we divide the highway roads into global and local sections. The global sections are specified in the purpose of capturing changes in traffic demand of certain road segments, and the local sections are specified in the purpose of capturing more detailed traffic state such as free flow, back of queue, bottleneck front, and congestion (Song and Yeo, 2012; Yeo et al., 2012). The range of a global section is wider than the range of a local section. Note that the variables used for global sections have a low spatial resolution compared to those for local sections. So, a global section should be wider than a local section, to obtain large enough information on traffic demand changes. Also, the performance of demand prediction is significantly affected by the spatial range and temporal distribution of demand patterns (Wen, 2008; Zhou and Mahmassani, 2006, 2007). So, the range of a global section should be relatively wider.

Korea's main highway network is divided into 23 global sections and 46 local sections based on the correlation analysis on detectors near junction. The correlation between detectors and sections significantly decreases near junctions due to the in/outflow from/to other highway routes. The global and local sections use different data sources. To represent traffic demand changes in global sections, we use the data collected through TCS and VDS and averagely 6 TCS detector data and 16 VDS detector data are used for each global section. TCS collects the number of vehicles entering or leaving highways, and VDS collects data such as average speed, flow, and average occupancy from loop

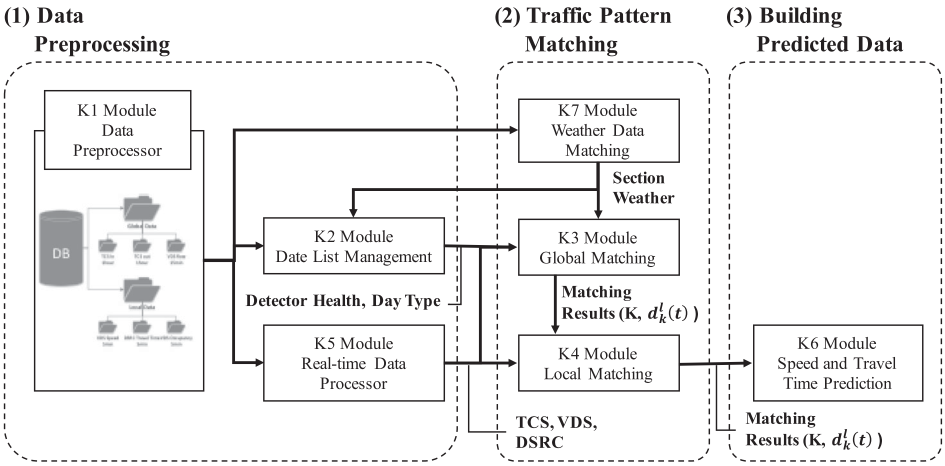
(1) Data Preprocessing
(2) Traffic Pattern Matching
(3) Building Predicted Data
K1 Module Data Preprocessor
DB
K7 Module Weather Data Matching
Section Weather
K2 Module Date List Management
K3 Module Global Matching
Detector Health, Day Type
Matching Results (K, $d_k^l(t)$)
K5 Module Real-time Data Processor
K4 Module Local Matching
K6 Module Speed and Travel Time Prediction
TCS, VDS, DSRC
Matching Results (K, $d_k^l(t)$)

Fig. 1. Data-driven framework for predicting speed and travel time in real-time.

detectors. The feature vectors of a global section for traffic pattern matching are defined as follows:

$$X_{TCS,in}^{(g)}(t) = \begin{bmatrix} N_{t-\tau}^{(1)} & \cdots & N_{t-\tau}^{\left(G^{(g)}\right)} \\ \vdots & \ddots & \vdots \\ N_t^{(1)} & \cdots & N_t^{\left(G^{(g)}\right)} \end{bmatrix} \tag{1}$$

$$X_{TCS,out}^{(g)}(t) = \begin{bmatrix} M_{t-\tau}^{(1)} & \cdots & M_{t-\tau}^{\left(G^{(g)}\right)} \\ \vdots & \ddots & \vdots \\ M_t^{(1)} & \cdots & M_t^{\left(G^{(g)}\right)} \end{bmatrix} \tag{2}$$

$$X_{VDS,Flow}^{(g)}(t) = \begin{bmatrix} F_{t-\tau}^{(1)} & \cdots & F_{t-\tau}^{\left(G^{(g)}\right)} \\ \vdots & \ddots & \vdots \\ F_t^{(1)} & \cdots & F_t^{\left(G^{(g)}\right)} \end{bmatrix} \tag{3}$$

The set of detectors within a global section is $q = \{1, 2, 3, \ldots, G^{(g)}\}$, where $G^{(g)}$ is the number of detectors located within global section g, and each value of q is the detector ID within the section. The value $G^{(g)}$ varies by global sections and type of detectors (tollgates and loop detectors).

The feature of detailed traffic pattern in a local section can be expressed with the data collected through VDS and DSRC system. DSRC system collects sampled data of travel time and speed for each DSRC link. Several RSUs are installed on road sections and approximately 65% vehicles, which use the Korean highway, install the DSRC device in the vehicle. By using these data sources, the feature vectors of a local section for traffic pattern matching are defined as follows.

$$X_{VDS,Occ}^{(l)}(t) = \begin{bmatrix} O_{t-\tau}^{(1)} & \cdots & O_{t-\tau}^{\left(L^{(l)}\right)} \\ \vdots & \ddots & \vdots \\ O_t^{(1)} & \cdots & O_t^{\left(L^{(l)}\right)} \end{bmatrix} \tag{4}$$

$$X_{VDS,Sp}^{(l)}(t) = \begin{bmatrix} S_{t-\tau}^{(1)} & \cdots & S_{t-\tau}^{\left(L^{(l)}\right)} \\ \vdots & \ddots & \vdots \\ S_t^{(1)} & \cdots & S_t^{\left(L^{(l)}\right)} \end{bmatrix} \tag{5}$$

$$X_{DSRC,Sp}^{(l)}(t) = \begin{bmatrix} C_{t-\tau}^{(1)} & \cdots & C_{t-\tau}^{\left(L^{(l)}\right)} \\ \vdots & \ddots & \vdots \\ C_t^{(1)} & \cdots & C_t^{\left(L^{(l)}\right)} \end{bmatrix} \tag{6}$$

If we define that the set of detectors within a local section is $r = \{1, 2, 3, \ldots, L^{(l)}\}$, then $L^{(l)}$ is the number of detectors located within local section l and each value of r is the detector ID within the section. The value $L^{(l)}$ varies by local sections and type of detectors (loop detectors and RSUs of DSRC). The range of neighboring time intervals is set to 4 hour by referencing the previous study with a consideration of computation time and accuracy (Tak et al., 2014b).

The feature vectors in this study are different from the feature vector used in previous studies (Davis and Nihan, 1991; Gong and Wang, 2002; Smith et al., 2002). The goal of the pattern matching in previous studies is to find similar traffic patterns for each individual detector. So, the feature vector used in previous studies is defined at the level of individual detectors, meaning that only the temporal and spatial relation are limitedly

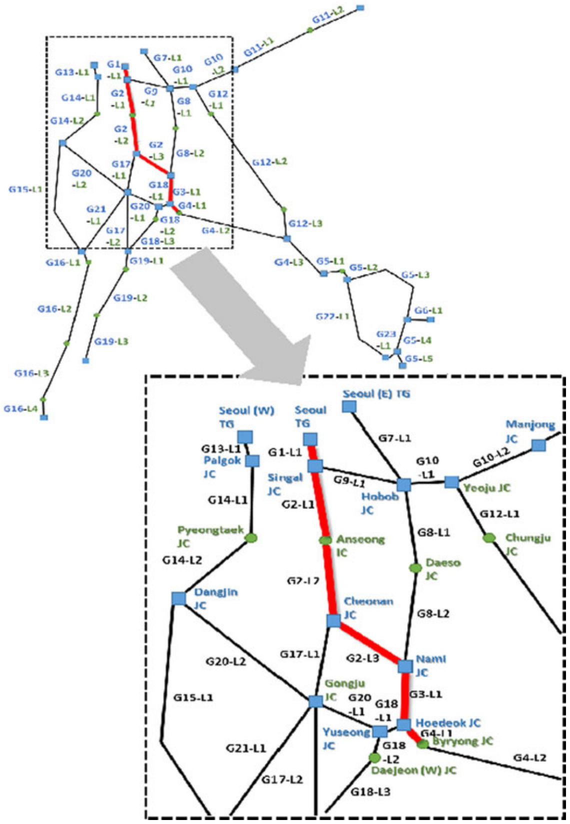

Fig. 2. Specified network of Korea's main line highways.

considered. The feature vector used in other studies can be expressed with:

$$X_t^{(s)} = \begin{bmatrix} Z_{t-\tau}^{(s-j)} & \cdots & Z_{t-\tau}^{(s+j)} \\ \vdots & \ddots & \vdots \\ Z_t^{(s-j)} & \cdots & Z_t^{(s+j)} \end{bmatrix} \tag{7}$$

Using the feature vector in Equation (7), previous pattern matching methods find similar patterns for each detector and predict future values by aggregating the results. This leads to increases in computation time, and limits the search range for historical traffic patterns (Clark, 2003; Haworth and Cheng, 2012; Nair et al., 2001). Generally, three to four neighboring detectors are used for matching the patterns.

Due to the limited searching range, the results from previous methods can get caught up in local minima. To overcome the weakness, we define new feature vectors as provided in Equations (1)–(6). By defining the feature vectors with a specified road section rather than with an individual detector, wider range of traffic information can be used. Such process can help the pattern matching escape the local minima.

To explain how the feature vectors will be used, we want to show a case of the traffic pattern matching. When finding a similar pattern in a road section with five detectors, previous studies executed the pattern matching for each detector. Such strategy may have different results by detectors, and it may cause inconsistency in the prediction results according to detectors. However, a matching strategy using the proposed feature vectors can find the similar pattern at once for the entire road section, in which the five detectors are installed. By investigating the feature vectors at the level of entire road section, the changes of traffic state within the section can be more clearly seen. The traffic patterns such as free flow, back of queue, bottleneck front, and congestion can be captured.

In modules K1 and K5, the raw data collected from the various sources are preprocessed before the pattern matching on the specified road sections. In module K1, 1-day data from the detectors are grouped by each road section. The rearranged data sets are then stored in the directories named with dates. Module K1 is implemented once a day with a batch process. As the daily process is done by module K1, module K2 generates monitoring data, which record the information on day type, weather condition, and detector health for each global and local section. In this module, the detector health for each global and local section is aggregated and continuously monitored. The state of all detectors in each section is expressed as an average value, and the dates having bad detector health are removed from the list for pattern matching. This process provides a good foundation for high speed searching for pattern matching, and maintains high quality of pattern matching. In module K5, real-time data are also grouped with the same manners of module K1. Only the difference is that module K5 is implemented at every 5 minutes to obtain real-time data sets to be used for pattern matching.

2.2 Traffic pattern matching

The central subject in traffic pattern matching is hierarchical strategy using k-NN method. The pattern matching process consists of three layers: classification, global, and local matching. With such arrangement of the layers, the search space of historical data sets can be reduced. With the effect of reduced search space, the computation time for matching process is reduced while maintaining appropriate accuracy.

In layer 1, classification, the historical days are classified by day types. The day types are Monday to Thursday, Friday, Saturday, Sunday, and holiday. Also, the historical days are classified by weather types. The weather types are dry, rainy, and snow. Traffic demand and travel time patterns are quite different depending on the day and weather types (Cools et al., 2010; Jun, 2010; Lam et al., 2008; Liu and Sharma, 2006a, b; Liu et al., 2008; Maze et al., 2006). So, for each road section, we only use the data sets of historical days that had the same condition for the pattern matching with the current data set. The list of classified days (the historical days with the same condition) is then used in the next layer, global matching.

In layer 2, global matching, we find a number of similar days in terms of traffic demand pattern for each global section by using feature vectors of Equations (1)–(3) from the list of classified days. For demand pattern matching, TCS data with 1-hour interval and VDS data with 15-minute interval are used. TCS data contains in/outflow of tollgates, and such values represent external demand. For the global matching, the flow data from only the loop detectors that are located near junction are used. In this way, we can estimate the in/outflow of the global section with the data. Such values represent the internal demand effect, which is related to other sections of the highway network. Using such knowledge and data, the dissimilarity between the subject data set and each of the historical data sets (which are classified in layer 1) is calculated with the distance equation as follows:

$$d_i^g(t) = W_{In} \cdot X_{TCS,in}^{(g)}(t) - D_{In,i}^{(g)}(t) + W_{Out} \cdot X_{TCS,out}^{(g)}(t) - D_{Out,i}^{(g)}(t) + W_{Flow} \cdot X_{VDS,Flow}^{(g)}(t) - D_{Flow,i}^{(g)}(t) \tag{8}$$

where

$$x = \sqrt{x_{1,1}^2 + \cdots + x_{m,n}^2},\ \mathrm{x} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix}$$

This demand-based approach makes the k-NN algorithm produce the stable result because the demand on a road section is highly related to the long-term traffic trend (Wen, 2008; Zhou and Mahmassani, 2007). The data used in global matching is not sensitively changed compared to the speed or flow data.

In layer 3, local matching, we find a number of similar days for each local section by feature vectors of Equations (4)–(6) among the results from global matching. The traffic properties used in this process are travel time values from DSRC, speed and occupancy values from VDS. All data sets have 5-minute intervals. A global section consists of several consecutive local sections. The local sections within the same global section find the similar traffic pattern from the same historical days that are sorted in the global matching layer. Using this matching strategy, each local section can find the similar historical data sets that have the similar traffic pattern in terms of both the traffic state and demand level. So, the accuracy of finding similar traffic pattern and consistency between local sections can be improved, and it can overcome the limitation of global matching that cannot capture the details of traffic changes such as time of speed drop and speed drop location. Also, by reducing the searching space of historical data sets step by step, the computation time for pattern matching can be reduced. The dissimilarity between the subject data set and each of the historical data sets (which are results in layer 2) is calculated with the distance equation as follows:

$$\begin{aligned} d_i^l\ (t) = {} & W_{Occ} \cdot X_{VDS,Occ}^{(l)}(t) - D_{Occ,i}^{(l)}(t) \\ & + W_{Sp} \cdot X_{VDS,Sp}^{(l)}(t) - D_{Sp,i}^{(l)}(t) \\ & + W_{DSRC} \cdot X_{DSRC,Sp}^{(l)}(t) - D_{DSp,i}^{(l)}(t) \end{aligned} \tag{9}$$

$$H_{Detector,\ i}^{(g)}(t) = \begin{bmatrix} h_{subject}^{Detector,(1)}(t-\tau)\cdot h_{history'i}^{Detector,(1)}(t-\tau) & \cdots & h_{subject}^{Detector,(G^{(g)})}(t-\tau)\cdot h_{history,i}^{Detector,(G^{(g)})}(t-\tau) \\ \vdots & \ddots & \vdots \\ h_{subject}^{Detector,(1)}(t)\cdot h_{history,i}^{Detector,(1)}(t) & \cdots & h_{subject}^{Detector,(G^{(g)})}(t)\cdot h_{history,i}^{Detector,(G^{(g)})}(t) \end{bmatrix} \tag{10}$$

$Detector \in \{TCSin,\ TCSout,\ VDSflow\}$,
$h_{subject}^{Detector,\ \left(G^{(g)}\right)}(t) \in \{0, 1\}$, $h_{history,i}^{Detector,\ \left(G^{(g)}\right)}(t) \in \{0, 1\}$

$$\begin{bmatrix} h_{subject}^{Detector,(1)}(t-\tau)\cdot h_{history'i}^{Detector,(1)}(t-\tau) & \cdots & h_{subject}^{Detector,(L^{(l)})}(t-\tau)\cdot h_{history,i}^{Detector,(L^{(l)})}(t-\tau) \\ \vdots & \ddots & \vdots \\ h_{subject}^{Detector,(1)}(t)\cdot h_{history,i}^{Detector,(1)}(t) & \cdots & h_{subject}^{Detector,(L^{(l)})}(t)\cdot h_{history,i}^{Detector,(L^{(l)})}(t) \end{bmatrix} \tag{11}$$

$$Detector \in \{VDSocc,\ VDSsp,\ DSRCsp\},$$

$$h_{subject}^{Detector,\ (L^{(l)})}(t) \in \{0,1\},\ h_{history,i}^{Detector,\ (L^{(l)})}(t) \in \{0,1\}$$

$$d_i^g(t) = W_{In}\cdot \frac{f_m\left(X_{TCS,in}^{(g)}(t),\, H_{TCSin,\ i}^{(g)}(t)\right) - f_m\left(D_{In,i}^{(g)}(t),\, H_{TCSin,\ i}^{(g)}(t)\right)}{H_{TCSin,\ i}^{(g)}(t)^2} \tag{12}$$

$$d_i^l(t) = W_{Occ}\cdot \frac{f_m\left(X_{VDS,Occ}^{(l)}(t),\, H_{VDSocc,\ i}^{(l)}(t)\right) - f_m\left(D_{Occ,i}^{(l)}(t),\, H_{VDSocc,\ i}^{(l)}(t)\right)}{H_{VDSocc,\ i}^{(l)}(t)^2} \tag{13}$$

There are several measures of dissimilarity, such as Euclidean Distance, Squared Euclidean Distance, Manhattan Distance, etc. Here, we use the Euclidean Distance, because it represents the straight-line distance between observation variable space, and it is the most commonly used in many disciplines. Particularly, the traffic data that we use is in the form of time-series. So, as we carry out the pattern recognition between subject data and historical data, the straight-line distance between the two data sets that are in time-series format would well represent the difference in the data pattern. When calculating the Euclidean Distance, most of the existing studies assume that the data is perfect (Clark, 2003; Gong and Wang, 2002; Kim and Kim, 2005). This distance metric is able to deal with heterogeneous variables for dissimilarity (Robinson, 2006). However, in real-world, traffic data contains missing data points due to many issues in hardware or data transfer (Sharma et al., 2004). In this case, data imputation process is required, and when it is carried out, the computation time for pattern matching increases. Thus, we propose a dissimilarity calculation method that can be applied when either of historical or subject data sets are imperfect. For this procedure, we generate data health matrix in modules K1 and K5 for each road section. The data dissimilarity calculation using the data health matrix is defined as Equations (10) and (11).

The individual health value ($h_{history,i}^{Detector,\ (L^{(l)})}(t)$) represents the availability of data of detector $L^{(l)}$ at time t. This value is zero if a data point has a missing value. The heath value is one if not.

Using the proposed data health matrices, the data dissimilarity calculations for $d_i^g(t)$ of Equation (8) and $d_i^l(t)$ of Equation (9) are then redefined as Equations (12) and (13).
Where

$$\mathrm{A} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \tag{14}$$

$$\mathrm{B} = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \tag{15}$$

$$f_m(A,B) = \begin{bmatrix} a_{11}\cdot b_{11} & \cdots & a_{1m}\cdot b_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1}\cdot b_{n1} & \cdots & a_{nm}\cdot b_{nm} \end{bmatrix} \tag{16}$$

Based on these equations, the dissimilarity between the subject and historical data sets is calculated by using only available data points. With this technique, the pattern matching process does not require any additional work for data imputation while keeping an effective performance of finding similar data sets. In Equations (12) and (13), a squared norm of data health matrix is used in the denominators to reduce the chance to select the data with low detector health. If we only use a norm of data health matrix in the denominators, historical dates with extremely low detector health and fortunately a few matched data points may show the low dissimilarity. Even though these dates show the low dissimilarity, the dates with large available points and slightly higher dissimilarity are more appropriate for the pattern matching, because the historical dates with low available points may lead to the large error and unstable results of matching. By using the squared value in the denominators, such possible negative effect is prevented and the possibility the dates with high available points being selected increases for finding the similar traffic data pattern.

When the size of data is not sufficient to implement the proposed algorithm, the situation that calculated minimum dissimilarity is large can occur due to a poor data quality of filtered date in layer 1. To overcome this limitation, the calculated dissimilarity is monitored after local pattern matching. So, if the minimum dissimilarity is larger than a threshold value obtained from a distribution of dissimilarity, the proposed algorithm finds the nearest neighbors again without classification of day and weather condition.

2.3 Building predicted data

In module K6 (speed and travel time prediction), we predict the future speed and then estimate the travel time from certain origin to destination. The pattern matching results of local sections (module K4) are used for the prediction based on the assumption that the date having the most similar traffic pattern from certain time until the current time will indicate the most similar traffic pattern in the future. So, we select K nearest historical data sets and K is set into three by referencing pervious research to ensure high accuracy and maintain the uniqueness of traffic phenomenon such as speed drop (Tak et al., 2014b). Based on the selected historical data, the data points in future time period is produced. Such process is carried out with speed data of each detector of each road section, and the speed prediction is done with the following Equation (17) based on the calculated distance value d_k.

$$P^{(l)}\ (\mathrm{t}+\sigma)=\sum_{k=1}^{K}\frac{V_k^{(l)}\,(\mathrm{t}+\sigma)}{d_k^l\,(\mathrm{t})}\Big/\sum_{k=1}^{K}\frac{1}{d_k^l\,(\mathrm{t})} \qquad (17)$$

This equation is commonly used in other types of k-NN methods (Haworth and Cheng, 2012; Liu et al., 2008; Robinson, 2006). The predicted value of link l after time σ is calculated by weighted average of historical values of Kth nearest neighbors of link l at time $(t+\sigma)$. In general, the accuracy of the predicted data decreases as σ increases. The dissimilarity of kth historical data is used for giving a weight on the predicted value. As the dissimilarity is lower, the weighting factor is greater. The predicted data may be unstable when $d_k^l(t)$ nearly equals to zero. To prevent such situation, we set the minimum value of $d_k^l(t)$ to 0.001, below the value 0.1% of Euclidean distance in all data sets.

3 DISTRIBUTED COMPUTING FOR REAL-TIME TRAFFIC INFORMATION SERVICE

The proposed traffic pattern matching strategy has a powerful potential to reduce the computation time of the entire prediction process. As described earlier, with the proposed strategy, the prediction can be implemented independently by each road section, and such design enables the distributed computing for the entire prediction process. With the cloud computing technology and Structured Query Language (SQL) database system, a distributed computing for the entire prediction process can be implemented in various ways. We can implement the prediction process by each global road section or by a specified region. A region can be specified as a group of global sections, and global sections can be grouped differently depending on the computation environment such as computer cluster size and CPU powers. For example, let us assume that there is a distributed computing system with 10 computers. Each computer performs the prediction process for a single region that is grouped with 100 global sections. If we use 10 additional computers for the distributed computing, then we can regroup the regions into 20 regions with 50 global sections. Then, the work load for each computer would be reduced and the computation time can be reduced with that effect. So, the proposed strategy also shows the flexibility in adjusting the computation power.

In this study, the performance of the proposed prediction method is tested by using a distributed computing technique with the program called Microsoft Azure (Armbrust et al., 2010; Pace et al., 2010; Wilder, 2012). Microsoft Azure is a cloud platform that provides integrated services of compute, storage, data, and network. For testing the proposed prediction framework, we use 10 virtual machines for running the entire prediction process and each virtual computer has four 2.20 GHz CPUs with a 3.50 GB RAM memory. The tested results show that the computation time for the entire prediction process is 45 seconds on average. The prediction covers approximately 50% of the Korean highway network. The length of the entire network is 1,800 km, and there are 143 tollgates (TCS), 3,408 loop detectors (VDS), and 843 RSUs (DSRC). Considering the data sets with 5-minute intervals, we can provide the real-time prediction service at every 5 minutes. If we use a larger size of cluster or work more deeply on the cluster design, then the computation time can be reduced even more. However, the main focus of this study is to validate the proposed method by comparing it with the existing method. Following sections provide case studies and results for the validation, and these works are done with the computation environment described above.

4 CASE STUDY

4.1 Evaluation

For testing the proposed prediction strategy, some case studies are practiced with real-world traffic data. The study site is a highway route from Seoul to Daejeon as shown in Figure 2, and this is the path where the congestion most irregularly and frequently occurs. The route is 131 km in length and the speed limit is 110 km/h. The specified route's free flow travel time is approximately 90 minutes, and the maximum travel time that has been observed so far is approximately 145 minutes. There are seven JCs within this route. The numbers of global sections and local sections within the route are three and six, respectively.

In this study, the proposed prediction strategy is tested with weekend data. The existing heuristics-based prediction methods can well predict the future traffic, particularly when predicting traffic patterns during weekdays (Kim, 1996; Chrobok, 2005). On Korean highways, during weekdays, the traffic patterns show similar patterns in commuting hours (e.g., morning and evening peak hours). This pattern makes the accuracy of heuristics-based methods become higher. On the other hand, the traffic patterns during weekends or holidays appear irregularly on Korean highways with higher demand for use of the network. Not only does the level of congestion differ, but also the time of congestion occurrence differs by different days of weekends. So, it is more probable that the accuracy of heuristics-based method would be lowered when predicting weekend traffic due to the severe and irregular congestion patterns. Therefore, we use 12 days of Saturday traffic data sets as the subject data, to show that the proposed strategy can appropriately predict the future traffic of weekend days.

The proposed prediction strategy can be evaluated in terms of computation time, accuracy, and robustness. The time for traffic pattern matching and calculating future traffic values mainly depends on the number of links in road section to be predicted. The computation time shows the applicability of the proposed method to the real-world. This study is carried out with the purpose of actually applying it to the real-world traffic information service system. The prediction accuracy is the most important factor in performance evaluation. The accuracy can be derived by comparing the predicted future traffic with actual data of the subject days (12 Saturday data sets). The values to be compared are future speed and travel time. The accuracy can be expressed with root-mean-square error (RMSE), mean absolute percentage error (MAPE), and mean error (ME) for specific prediction horizons (Rashidi and Ranjitkar, 2015). Also, it is important to see excessive errors. Even though there were some predicted data points showing 90% MAPE, if the entire data points show relatively low MAPE values, then total MAPE value would be low. So, we also check the accuracy with 10% of worst data points in terms of both MAPE and RMSE, which we call "worst 10% MAPE" and "worst 10% RMSE," respectively. The equations for MAPE and RMSE are provided as follows:

$$RMSE = \sqrt{\frac{\sum e_t^2}{n}} \tag{18}$$

$$MAPE = 100 \cdot \frac{\sum \left| \frac{e_t^{(l)}}{V_t^{(l)}} \right|}{n} \tag{19}$$

$$ME = \frac{\sum e_t^{(l)}}{n} \tag{20}$$

The prediction accuracy is highly related to prediction horizon. Generally, the prediction accuracy decreases as the prediction horizon increases. So, to analyze the effect of prediction horizon on the accuracy, we also calculate the MAPE and RMSE with different prediction horizons.

Robustness is another factor of evaluating the performance. It represents the ability of a prediction method in maintaining appropriate performance in different traffic conditions. A prediction method may be more accurate when traffic state is free flow, but less accurate when traffic is congested, or vice versa. Particularly in road traffic, it is important to predict the time when congestions occur and become released. A method should fairly predict the travel time regardless of traffic state. This robustness however cannot be told with either of total MAPE for speed prediction or total RMSE for speed prediction. So, to check the robustness of the proposed method, the predicted travel time is to be compared with actual data sets in both free flow and congested conditions.

4.2 Benchmark models

For evaluation, the proposed method is compared with nearest historical (NH) average method and the conventional k-NN method. NH predicts the future traffic speed with average historical data collected on the same detector at the same time but from a neighboring day (Conklin and Scherer, 2003; Williams and Hoel, 2003; Williams, 2001). It is one of the most common method in the traffic prediction with stable prediction output with relatively easy and intuitive implementation even when the detector health is not good. For this reason, this method is frequently used to compare the accuracy of newly proposed algorithm by many researchers (Conklin and Scherer, 2003; Williams and Hoel, 2003; Williams, 2001). In this study, the arithmetic average speed of the same time and same day of week over 7 historical days is used to produce the future speed.

Comparing with conventional k-NN method, the main differences of the proposed method lie in the strategy of traffic pattern matching and in the feature vectors for calculating the pattern dissimilarity. Conventional k-NN (Ck-NN) has been widely used in the field of nonparametric pattern matching (Chrobok, 2005; Clark, 2003; Robinson, 2006; Smith et al., 2002), and short-term future traffic prediction. Ck-NN finds the similar data pattern for each individual detector and aggregates the individually predicted results. It considers the neighboring detector's data patterns as well to deal with the spatial and temporal relations. Ck-NN is appropriate as the reference for comparison analysis, as it shows the good performance in the previous studies for short-term traffic prediction. Referencing the previous researches, we set the values of n and t as 3 (Chrobok, 2005; Clark, 2003; Robinson, 2006; Smith et al., 2002).

The global traffic pattern matching method (Global variables-based k-NN: Gk-NN) finds the similar traffic pattern only based on the global pattern for road section. In this method, only Equations (1)–(3) and (15) are used for traffic pattern matching. We use demand values (TCS in/out flow and VDS flow) as the global variables, so, it shows how well it can predict future speed based on changes in traffic demand.

We call the local traffic pattern matching method Local variables-based k-NN (Lk-NN), which is provided in Section 2. This method finds the similar traffic pattern only based on the local pattern for each road section. In this method, only Equations (4)–(6) and (16) are used for traffic pattern matching. The results of Lk-NN represent how well this method can predict the future traffic by only using the local variables. We use speed, occupancy, and travel time values for local variables, so, it means that Lk-NN shows how well it can predict future speed based on changes in the various local variables. Also, by comparing the results of Lk-NN with Ck-NN, the effect of differences in feature vector for pattern matching can be observed.

Denote the proposed hierarchical combination of Gk-NN and Lk-NN as Mk-NN. In pattern matching, the dissimilarity is calculated by using Equations (12) and (13), and weights W_{In}, W_{Out}, W_{Flow}, W_{Occ}, W_{Sp}, and W_{DSRC} are set as 0.8, 0.8, 0.2, 0.8, 0.8, and 0.2. These weights are determined based on a numerical study on the effects of weights on the performance of the proposed prediction algorithm. In the numerical study, the best combination of weights for all matching strategies such as Ck-NN, Gk-NN, Lk-NN, and Mk-NN are determined by using Genetic Algorithm, so all matching strategies were tested with different weights in case study. By comparing the results of Mk-NN with the three other methods, the effect of hierarchical pattern matching strategy on the prediction accuracy can be seen. Such comparison results are provided in the following section.

Table 2
Comparison of computation time (seconds)

	Number of links for matching				
	3	*40*	*48*	*87*	*115*
Ck-NN	1.23	16.66	19.83	36.03	46.47
Gk-NN	0.43	1.19	1.68	2.60	3.39
Lk-NN	0.48	3.56	4.47	8.21	11.70
Mk-NN	0.52	1.82	2.47	4.07	5.44

5 EXPERIMENTAL RESULTS

The computation time is defined as the required time for finding three most similar historical traffic patterns of each detector. Table 2 shows the computation time for pattern matching process with different number of road links that are applied to the traffic pattern matching methods (Ck-NN, Gk-NN, Lk-NN, and Mk-NN). As shown in the table, Ck-NN shows the worst performance in terms of computation time, because such method is designed to match the traffic pattern of each individual detector. Such result shows that the proposed method (road section-based traffic pattern matching) can effectively reduce the computation time for the pattern matching process. The computation efficiency is stable, because even if the number of road links increases, the computation time for pattern searching increases with small fluctuation. Averagely, Gk-NN is 11 times faster, Lk-NN is 4 times faster, and Mk-NN is 8 times faster than Ck-NN. Gk-NN shows the minimum computation time, which is 1.5 times faster than Mk-NN. This happens because Mk-NN executes additional pattern matching process (local matching) after finishing the global matching process. Even though this additional process leads to slight increases in the computation time of Mk-NN, it has to be done for improving the accuracy and robustness of the method.

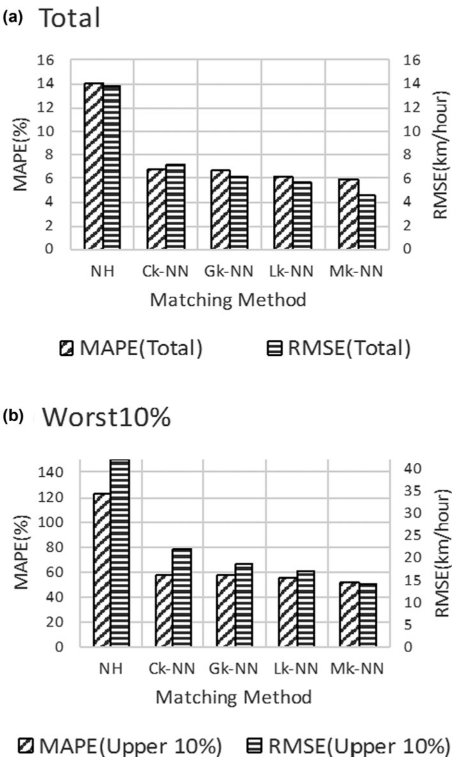

Fig. 3. MAPE and RMSE of speed prediction.

The accuracy and robustness will be discussed in the following contents.

Figures 3a and b show the MAPE and RMSE values of five prediction methods for the 12 tested Saturdays. These values are calculated with predicted speed and actual speed values within 1-hour prediction horizon. Refer to Figure 3, NH method shows 13.8 km/h of RMSE and 14.1% of MAPE value, which is slightly higher than MAPE value tested in previous study (Williams and Hoel, 2003; Williams, 2001). In the previous study, NH method shows 11–12% of MAPE value and NH method is compared to other methods such as seasonal autoregressive integrated regressive model (8–9% of MAPE value) and random walk-based prediction model (10–12% of MAPE value). By referencing these results, we compare the accuracy of proposed prediction algorithm to other prediction algorithm implemented in previous researches in a roundabout way and the accuracy of NH method is used as a base case.

As shown in Figure 3, NH method shows the highest MAPE and RMSE, and other four prediction methods (Ck-NN, Gk-NN, Lk-NN, and Mk-NN) show good enough performances. Particularly, Ck-NN shows similar performance compared to previous research that had 6.748% of MAPE value and 7.216 km/h of RMSE value (Clark, 2003). By comparing simple prediction methods, which were studied in the previous researches such as NN and Auto Regressive, the improvement of the proposed algorithm is also observed (e.g., 8.74% and 8.97% of MAPE value using seasonal autoregressive integrated regressive model (Williams and Hoel, 2003; Williams, 2001) and 12.12% of MAPE using nearest neighbor method (Sun et al., 2003)). Such results show that a k-NN based prediction method can be used for long-term prediction as well. This is due to the heuristic characteristics of k-NN that utilizes the historical data for prediction. So, even though there are some errors in prediction compared to the actual data, k-NN-based method can robustly estimate the future speed of road sections, as long as the day type and weather are considered when searching historical data.

Comparing the four methods, Mk-NN method shows the best accuracy with 5.87% of MAPE and 4.16 km/h of RMSE, which are much better than those of Ck-NN. Lk-NN shows the second best accuracy, and Gk-NN shows the third best. These results show that, rather than predicting traffic of an individual road link, dealing with an entire road section at once (pattern matching with a group of detectors at once) improves the prediction accuracy. Even though Ck-NN still considers the spatial and temporal traffic relations by using the neighboring detector's data for pattern matching, predicting future traffic with the road section-based feature vectors is more effective. The pattern matching method using the road section-based feature vectors finds the similar patterns in terms of full range of a road section (multiple road links) rather than significantly considering the differences between the subject data and historical data of a single road link. So, the method using the road section-based feature vectors can escape the local minima that can frequently trap link-based feature vectors. For example, when predicting five consecutive links in a section with link-based feature vectors, these five links show different nearest neighbor dates, such as March 09, 2013; November 30, 2013; July 13, 2013; December 07, 2013; and April 20, 2013, respectively. Even though these dates show the similar trends for each link with lowest similarity, the predicted value with these results cannot reflect the various traffic phenomena such as the propagation pattern of shockwave and propagation of congestion, because there is not clear relation between selected dates of each link. However, when predicting five consecutive links in a section with section-based feature vector, five links in the same section would produce the same nearest neighbor date, so the various traffic phenomena can be reflected in the predicted value. This leads to the higher accuracy and lower excessive error as shown in Figure 3.

As mentioned earlier, it is also important to check the excessive errors. Even if the average accuracy is high,

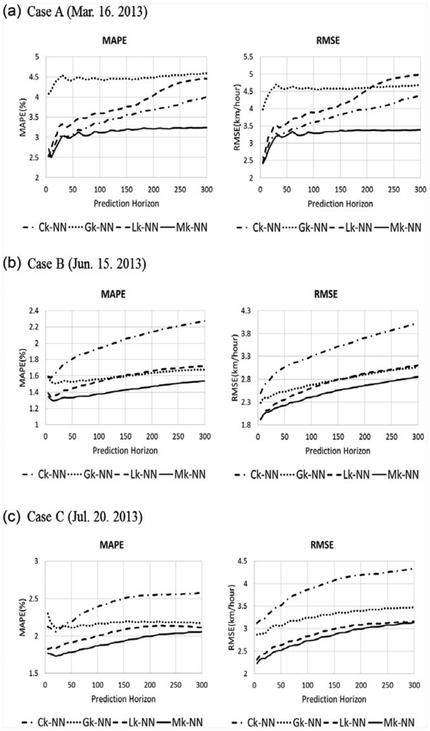

Fig. 4. MAPE and RMSE of speed prediction with various prediction horizons for three example cases.

intermittent occurrences of excessive errors may reduce the reliability of the prediction system. Figure 3b shows the worst 10% of MAPEs and the worst 10% of RMSEs of the four different methods. As in the figure, Mk-NN again shows the best performance among them in terms of excessive errors. Particularly for RMSE, Mk-NN's performance is twice better than Ck-NN. As we compare Ck-NN with Gk-NN, excessive errors show a bit different trends from the total comparison in Figure 3a. The worst 10% of MAPE for Ck-NN is slightly better than Gk-NN. This shows that Gk-NN considers demand patterns only, so it sometimes does not capture local traffic pattern like local congestions. This leads to the increase in the excessive errors of Gk-NN, even if the overall performance is better than Ck-NN.

In the comparison of the prediction accuracy, NH and k-NN based methods show significant differences in terms of both trend and accuracy. The NH method shows stable accuracy regardless of prediction horizon, so the NH produces the predicted value between 11.07% and 14.15% of MAPE value in all prediction horizons. However, these values are much higher than MAPE value of k-NN based methods (Ck-NN, Gk-NN, L-NN, and Mk-NN).

Overall, in k-NN based methods, the MAPE and RMSE values increase as the prediction horizon increases as shown in Figure 4. Among them, Gk-NN shows the lowest changes in both MAPEs and RMSEs for all prediction horizons, because traffic demand patterns can well represent the long-term trend of traffic change (Wen, 2008; Zhou and Mahmassani, 2007). Mk-NN shows the second lowest changes in both MAPEs and RMSEs for all prediction horizons. This means that both Gk-NN and Mk-NN are stable regardless of the size of prediction horizon. On the other hand, both Ck-NN and Lk-NN show significant changes in both MAPEs and RMSEs, meaning that their prediction results are greatly influenced by the size of prediction horizon. Particularly, Ck-NN shows the most dramatic changes by prediction horizons. So, even if Ck-NN shows a good prediction accuracy for relatively short prediction horizon, it results in the worst stability of accuracy as the prediction horizon increases, because such method focuses only on local traffic changes and cannot capture a larger trend of traffic changes.

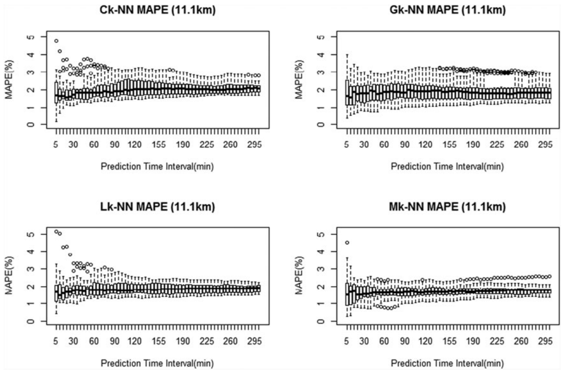

Fig. 5. MAPE for 11.1 km prediction.

The prediction accuracy results are also different by the cases. In all cases, the proposed Mk-NN shows the best performance for all prediction horizons. In case A, Ck-NN shows the second best accuracy, whereas Gk-NN is the second best in case B. Lk-NN is the second best in case C. Depending on demand pattern and local traffic, the performances of these three methods significantly changes. So, each of the three methods has weaknesses. Both Ck-NN and Lk-NN do not capture the larger trend of traffic changes. On the other hand, Gk-NN only deals with traffic demands and does not capture local traffic patterns. As only Mk-NN captures both traffic demand and local traffic changes, it shows the most stable performance. Therefore, the better performance of Mk-NN is clearly shown in terms of both prediction accuracy and robustness.

Figures 5 and 6 show how the accuracy of travel time prediction is influenced by the length of prediction route. As shown in the figures, all pattern matching methods show relatively accurate results when the length of prediction route is short. The differences

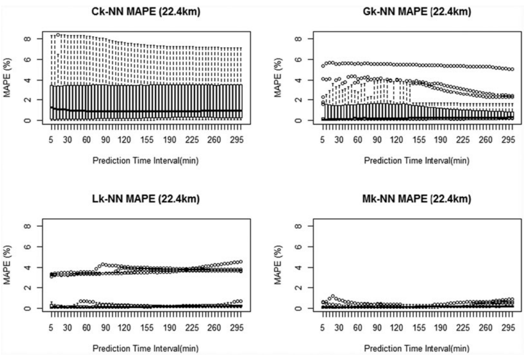

Fig. 6. MAPE for 22.4 km prediction.

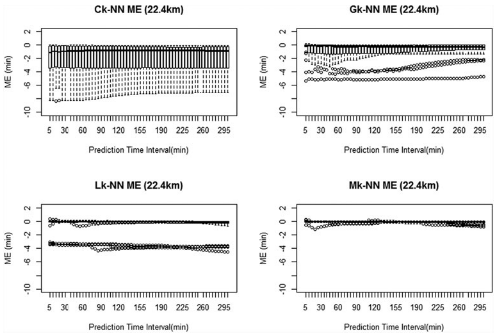

Fig. 7. ME for 22.4 km prediction.

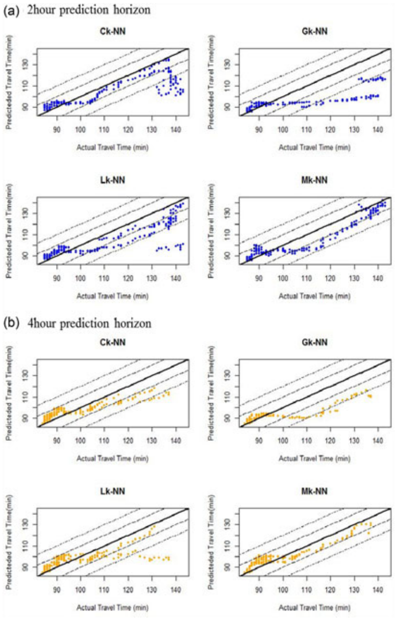

Fig. 8. Comparison of predicted and actual travel time with (a) 2-hour prediction horizon and (b) 4-hour.

among the methods are low, while Mk-NN is still outperforming the otherthree methods. As the length of prediction route increases, the differences among the methods clearly appear. When predicting travel time for 22.4 km in length, except for Mk-NN, the error ranges of all other methods significantly increase. Particularly, Ck-NN shows the maximum MAPE value of approximately 7%. Mk-NN also shows a stable performance even when the prediction route length increases. All MAPE values are under 1.5% for Mk-NN.

Figure 7 shows the direction of the error of prediction by different prediction horizons. Positive value represents the overestimation of predicted value, and negative value the underestimation of predicted value. As in the figure, Mk-NN shows the most balanced results with slightly underestimated predicted value. Next to Mk-NN, Lk-NN shows the second best performance with similar average ME to Mk-NN due to excessive error. However, Ck-NN underestimates the travel time compared to other pattern matching methods with much large error bound. Therefore, the robustness of the proposed Mk-NN is also observed in these results.

Figure 8 shows the example results showing the trend of predicted travel time in various traffic situations. It shows the reason that Mk-NN has the most balanced performance and Ck-NN has the worst performance in terms of ME shown in Figure 7. The prediction is for the study site (from Seoul to Daejeon), which is a highway route having a length of 131 km. In the figures, the *x*-axis represents the actual travel time and *y*-axis is for the predicted travel time. The solid lines in the middle of the figures represent where the predicted and actual travel time values are equal. So, as the data points are closer to this line, the prediction accuracy increases. The dashed lines represent 10-minute-error bound, and dashed-dotted lines for 20-minute-error bound. Small values of travel time represent the travel time during free flow traffic state, and large values of travel time represent congested traffic. So, Figure 8 shows the robustness of different travel time prediction methods in various traffic situations.

As shown in Figure 8, all four different methods can accurately predict the travel time during free flow state. The data points within free flow state are located within 10-minute-error range. The travel time of the specified route during free flow state is approximately 90 minutes, and all methods predict the travel time between 85 and 100 minutes. The common thing of all methods is that they provide slightly overestimated prediction values. On the other hand, as the specified route becomes congested, all methods show underestimated prediction values. This is due to the averaging effect of k-NN. In this study, three nearest neighbors from the historic pattern are chosen and then future traffic is predicted by obtaining the weighted average of the three nearest neighbors. The three neighbors have slight differences in congestion occurrence time and level of congestion. During the weighted averaging process, the low speed in congested state is smoothed by the value from other nearest neighbors, and this effect renders the predicted value to be underestimated.

The four methods show distinctive trends of predicting travel time. Mk-NN shows robust and accurate travel time prediction regardless of the prediction horizons. In all prediction horizons, the differences between predicted and actual travel time values are less than 20 minutes. Up to 4-hour prediction horizon, Mk-NN can accurately predict the travel time with 20-minute error range in any cases. Furthermore, most of the predicted values in Mk-NN is within 10-minute error range for 4-hour prediction horizon, showing the robustness of Mk-NN method in long-term prediction. Compared to Mk-NN, Ck-NN and Lk-NN show relatively low accuracy as the prediction horizon and the actual travel time increase. As shown in Figure 8a, for 1-hour prediction horizon, the differences of the actual and predicted values of Ck-NN and Lk-NN are mostly smaller than 20 minutes except some cases. For 2- and 4-hour prediction horizons, the errors of Ck-NN and Lk-NN increase as the traffic becomes congested. Considering 50 minutes of travel time difference between free flow and congested state, Ck-NN and Lk-NN lose the capability of capturing the increasing trend of travel time as the prediction horizon increases. Lastly, Gk-NN shows the largest errors in congested state for 2-hour prediction horizon. This means that Gk-NN fails to find the similar patterns for short-term traffic, but it better captures long-term traffic changes (4-hour prediction horizons) by using global variables. The entire results show that Mk-NN shows a stable prediction capability in various traffic situations even when the prediction horizon increases. Such higher robustness and accuracy are the results of the hierarchical traffic pattern matching strategy. As shown in Figure 8, the single-step pattern matching strategies (e.g., Gk-NN and Lk-NN) have their own weaknesses. Gk-NN has a weakness in predicting short-term traffic changes, while Lk-NN has a weakness in predicting long-term traffic changes. By combining the global and local matching strategies, we can predict long-term travel time more robustly and accurately.

6 CONCLUSION

In this study, we propose a data-driven framework for predicting speed and travel time of freeway network in real-time for cloud computing environment. The proposed framework consists of three steps: (1) data preprocessing step arranges the collected data sets for reducing computation time and improvement of the quality of traffic pattern matching results; (2) traffic pattern matching step finds the similar traffic pattern based on Mk-NN method. Applying Mk-NN with distributed computing, the computation time can be dramatically reduced, with higher accuracy and robustness; and (3) in building predicted data, we provide a method that can generate the predicted values based on the result of Mk-NN pattern matching process.

In long-term prediction, the Mk-NN algorithm can accurately and robustly predict the speed and travel time with relatively short computation time. Compared to the Ck-NN algorithm, Mk-NN algorithm improves the accuracy of predicting future speed by 13% of MAPE and by 36% of RMSE for 1-hour prediction horizon. Although improving the accuracy, Mk-NN algorithm can find the similar historical traffic patterns eight times faster than Ck-NN. In travel time prediction, the robustness of Mk-NN algorithm is also shown. Mk-NN shows consistent performance regardless of the prediction horizon and the traffic state, whereas other methods result in worse performances in congested condition.

Even though the proposed Mk-NN method can bring large benefits to prediction accuracy and computation time, there are still some works to be done for improving the prediction efficiency. Data-driven prediction methods hardly estimate the delays caused by rare incidents like accident and road maintenance work, even though there are some related studies that have tried to solve such problems (Adeli and Ghosh-Dastidar, 2004; Adeli and Karim, 2000; Adeli and Samant, 2000; Ghosh-Dastidar and Adeli, 2003; Jiang and Adeli, 2003; Karim and Adeli, 2003a). Therefore, further studies can include the method that can reflect the delays by unexpected events to the prediction method.

ACKNOWLEDGMENTS

This research was supported by Microsoft Research Award in Urban Informatics and Korea Minister of Ministry of Land, Infrastructure and Transport (MOLIT) as “U-City Master and Doctor Course Grant Program.”

REFERENCES

Adeli, H. & Ghosh-Dastidar, S. (2004), Mesoscopic-wavelet freeway work zone flow and congestion feature extraction model, *Journal of Transportation Engineering*, **130**(1), 94–103.

Adeli, H. & Karim, A. (2000), Fuzzy-wavelet RBFNN model for freeway incident detection, *Journal of Transportation Engineering*, **126**(6), 464–71.

Adeli, H. & Samant, A. (2000), An adaptive conjugate gradient neural network–wavelet model for traffic incident detection. *Computer-Aided Civil and Infrastructure Engineering*, **15**(4), 251–60.

Armbrust, M., Fox, A. & Griffith, R. (2010), A view of cloud computing, *Communications of the ACM*, **53**(4), 50–58.

Bajwa, S., Chung, E. & Kuwahara, M. (2005), Performance evaluation of an adaptive travel time prediction model, in *Proceedings of the 2005 IEEE Intelligent Transportation Systems*, Vienna, Austria.

Casas, J., Torday, A. & Perarnau, J. (2013), Present and future methodology for the implementation of decision support systems for traffic management, in *Proceedings of the 36th Australasian Transport Research Forum (ATRF)*, Brisbane, Queensland, Australia.

Castillo, E. & Nogal, M. (2012), Stochastic demand dynamic traffic models using generalized beta-Gaussian Bayesian networks, *IEEE Transactions on Intelligent Transportation Systems*, **13**(2), 565–81.

Chen, H., Grant-Muller, S., Mussone, L. & Montgomery, F. (2001), A study of hybrid neural network approaches and the effects of missing data on traffic forecasting, *Neural Computing & Applications*, **10**(3), 277–86.

Chien, S. & Kuchipudi, C. (2003), Dynamic travel time prediction with real-time and historic data, *Journal of Transportation Engineering*, **129**(6), 608–16.

Chow, A., Dadok, V., Dervisoglu, G., Gomes, G., Horowitz, R., Kurzhanskiy, A. A. & Sánchez, R. O. (2008), TOPL: Tools for Operational Planning of transportation networks, in *ASME 2008 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, pp. 1035–42.

Chrobok, R. (2005), *Theory and Application of Advanced Traffic Forecast Methods*. University of Duisburg-Essen, Campus Duisburg, Fachbereich Physik.

Chrobok, R., Hafstein, S. & Pottmeier, A. (2004), Olsim: a new generation of traffic information systems, *Forschung und Wissenschaftliches Rechnen*, **63**, 11–25.

Clark, S. (2003), Traffic prediction using multivariate nonparametric regression, *Journal of Transportation Engineering*, **129**(2), 161–68.

Conklin, J. H. J. & Scherer, W. W. T. (2003), *Data Imputation Strategies for Transportation Management Systems*, Report, UVACTS-13-0-80, Center for Transportation Studies, University of Virginia, 127.

Cools, M., Moons, E. & Wets, G. (2010), Assessing the impact of weather on traffic intensity, *Cools, Mario, Elke Moons, and Geert Wets*, **2**(1), 60–68.

Davis, G. A. & Nihan, N. L. (1991), Nonparametric regression and short-term freeway traffic forecasting, *Journal of Transportation Engineering*, **117**(2), 178–88.

Dharia, A. & Adeli, H. (2003), Neural network model for rapid forecasting of freeway link travel time, *Engineering Applications of Artificial Intelligence*, **16**(7), 607–13.

Dia, H. (2001), An object-oriented neural network approach to short-term traffic forecasting, *European Journal of Operational Research*, **131**(2), 253–61.

Dougherty, M. S. & Cobbett, M. R. (1997), Short-term interurban traffic forecasts using neural networks, *International Journal of Forecasting*, **13**(1), 21–31.

Elfaouzi, N. (1996), Nonparametric traffic flow prediction using kernel estimator, in *Proceedings of the International Symposium on Transportation and Traffic Theory*, Lyon, France, 41–54.

Fei, X., Lu, C.-C. & Liu, K. (2011), A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction, *Transportation Research Part C: Emerging Technologies*, **19**(6), 1306–18.

Ghosh-Dastidar, S. & Adeli, H. (2003), Wavelet-clustering-neural network model for freeway incident detection, *Computer-Aided Civil and Infrastructure Engineering*, **18**(5), 325–38.

Ghosh-Dastidar, S. & Adeli, H. (2006), Neural network-wavelet micro-simulation model for delay and queue length estimation at freeway work zones, *Journal of Transportation Engineering, ASCE*, **132**(4), 331–41.

Gong, X. G. X. & Wang, F. W. F. (2002), Three improvements on KNN-NPR for traffic flow forecasting, in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, 2002*, 736–40.

Haworth, J. & Tao, C. (2012), Non-parametric regression for space–time forecasting under missing data, *Computers, Environment and Urban Systems*, **36**(6), 538–50.

van Hinsbergen, C. & van Lint, J. (2008), Bayesian combination of travel time prediction models, *Transportation Research Record: Journal of the Transportation Research Board*, **2064**(1), 73–80.

van Hinsbergen, J.W.C. & van Lint, F. M. S. (2007), Short term traffic prediction models, in *Proceedings of the ITS World Congress*, Beijing, China, 608–16.

Hong, W., Dong, Y., Zheng, F. & Lai, C. (2011), Forecasting urban traffic flow by SVR with continuous ACO, *Applied Mathematical Modelling*, **35**(3), 1282–91.

Ishak, S. & Al-Deek, H. (2002), Performance evaluation of short-term time-series traffic prediction model, *Journal of Transportation Engineering*, **128**(6), 490–98.

Jiang, X. & Adeli, H. (2003), Freeway work zone traffic delay and cost optimization model. *Journal of Transportation Engineering*, **129**(3), 230–41.

Jiang, X. & Adeli, H. (2005), Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering*, **131**(10), 771–79.

Jun, J. (2010), Understanding the variability of speed distributions under mixed traffic conditions caused by holiday traffic, *Transportation Research Part C: Emerging Technologies*, **18**(4), 599–610.

Karim, A. & Adeli, H. (2002), Incident detection algorithm using wavelet energy representation of traffic patterns, *Journal of Transportation Engineering, ASCE*, **128**(3), 232–42.

Karim, A. & Adeli, H. (2003a), Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering*, **129**(5), 494–503.

Karim, A. & Adeli, H. (2003b), Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm, *Journal of Transportation Engineering, ASCE*, **129**(1), 57–68.

Kim, C. (1996), Development and evaluation of traffic prediction systems, *Transportation Research Part A*, **30**(1), 58.

Kim, T., Kim, H. & Lovell, D. J. (2005), Traffic flow forecasting: overcoming memoryless property in nearest neighbor non-parametric regression, in *Proceedings of the IEEE Intelligent Transportation Systems, 2005*.

Kwon, J., Coifman, B. & Bickel, P. (2000), Day-to-day travel-time trends and travel-time prediction from loop-detector data, *Transportation Research Record: Journal of the Transportation Research Board*, **1717**(1), 120–29.

Lam, W. H. K., Shao, H. & Sumalee, A. (2008), Modeling impacts of adverse weather conditions on a road network with uncertainties in demand and supply, *Transportation Research Part B: Methodological*, **42**(10), 890–910.

Li, M., Hong, W. & Kang, H. (2013), Urban traffic flow forecasting using Gauss–SVR with cat mapping, cloud model and PSO hybrid algorithm, *Neurocomputing*, **99**, 230–40.

Van Lint, J. W. (2006), Reliable real-time framework for short-term freeway travel time prediction, *Journal of Transportation Engineering*, **132**(12), 921–32.

Van Lint, J. W. C. (2008), Online learning solutions for freeway travel time prediction, *IEEE Transactions on Intelligent Transportation Systems*, **9**(1), 38–47.

Van Lint, J. W. C., Hoogendoorn, S. P., van Zuylen, H. J. & van Lint, J. (2005), Accurate freeway travel time prediction with state-space neural networks under missing data, *Transportation Research Part C: Emerging Technologies*, **13**(5), 347–69.

Liu, Z. & Sharma, S. (2006a), Predicting directional design hourly volume from statutory holiday traffic, *Transportation Research Record: Journal of the Transportation Research Board*, **1968**(1), 30–39.

Liu, Z. & Sharma, S. (2006b), Statistical investigations of statutory holiday effects on traffic volumes, *Transportation Research Record: Journal of the Transportation Research Board*, **1945**(1), 40–48.

Liu, Z., Sharma, S. & Datla, S. (2008), Imputation of missing traffic data during holiday periods, *Transportation Planning and Technology*, **31**(5), 525–44.

Mahmassani, H., Fei, X. & Eisenman, S. (2005), *DYNASMART-X Evaluation for Real-Time TMC Application: CHART Test Bed*. Maryland Transportation Initiative.

Maze, T., Agarwai, M. & Burchett, G. (2006), Whether weather matters to traffic demand, traffic safety, and traffic operations and flow, *Transportation Research Record: Journal of the Transportation Research Board*, **1948**(1), 170–76.

Nair, A., Liu, J., Rilett, L. & Gupta, S. (2001), Non-linear analysis of traffic flow, in *Proceedings of the 2001 IEEE Intelligent Transportation Systems*.

Oda, T. (1990), An algorithm for prediction of travel time using vehicle sensor data, in *Third International Conference on Road Traffic Control*, London, May 1–3, 1990, IET.

Oh, C., Oh, J., Ritchie, S. & Chang, M. (2001), Real-time estimation of freeway accident likelihood, in *Proceedings of the 80th Annual Meeting of the Transportation Research Board*, Washington DC.

Oh, S., Byon, Y. J., Jang, K. & Yeo, H. (2015), Short-term travel-time prediction on highway: a review of the data-driven approach. *Transport Reviews*, **35**(1), 1–29.

Pace, E., Betts, D., Densmore, S. & Dunn, R. (2010), *Moving Applications to the Cloud on the Microsoft Azure Platform*. Microsoft Press, ISBN: 9780735649675.

Papageorgiou, M. & Papamichail, I. (2010), Traffic simulation with Metanet, in *Fundamentals of Traffic Simulation*, Springer, New York, 399–430.

Park, D. & Rilett, L. (1999), Forecasting freeway link travel times with a multilayer feed forward neural network, *Computer-Aided Civil and Infrastructure Engineering*, **14**(5), 357–67.

Rashidi, S. & Ranjitkar, P. (2015), Bus dwell time modeling using gene expression programming, *Computer-Aided Civil and Infrastructure Engineering*, **30**(6), 478–89.

Rice, J. & Van Zwet, E. (2004), A simple and effective method for predicting travel times on freeways, *IEEE Transactions on Intelligent Transportation Systems*, **5**(3), 200–07.

Robinson, S. S. (2006), The development and application of an urban link travel time model using data derived from inductive loop detectors. Doctoral dissertation, Imperial College London.

Röhr, T., Lindenbach, A. & Balz, W. (1996), Entwicklung von verfahren zur grossraeumigen prognose der verkehrsentwicklung und folgerungen fuer den, *Forschung Straßenbau und Straßenverkehrstechnik*, Issue 727, ISSN: 0344-0788.

Saito, M. & Watanabe, T. (1995), Prediction and dissemination system for travel time utilizing vehicle detectors, *Steps Forward. Intelligent Transport Systems World Congress*, Yokohama, Japan, November 9, 1995, Vol. 1.

Sharma, S., Lingras, P. & Zhong, M. (2004), Effect of missing values estimations on traffic parameters, *Transportation Planning and Technology*, **27**(2), 119–44.

Siegener, W. & Schmitt, W. (1980), Prognose von Verkehrsmengen durch aktuelle Fortschreibung von Langzeiterwartungswerten, *Forsch Strassenbau und Strassenverkehrstech*, Bundesministerium fuer Verkehr, Bau und Wohnungswesen, Issue 316, ISSN: 0344-050X.

Smith, B., Williams, B. & Oswald, R. K. (2002), Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies*, **10**(4), 303–21.

Song, S. & Yeo, H. (2012), Method for estimating highway collision rate that considers state of traffic flow, *Transportation Research Record: Journal of the Transportation Research Board*, **2318**(1), 52–62.

Sun, H., Liu, H., Xiao, H., He, R. & Ran, B. (2003), Short term traffic forecasting using the local linear regression model, in *Proceedings of the 82nd Annual Meeting of the Transportation Research Board*, Washington DC.

Tak, S., Kim, S., Jang, K. & Yeo, H. (2014a), Real-time travel time prediction using multi-level k-nearest neighbor algorithm and data fusion method, in *Proceedings of the 15th International Conference on Computing in Civil and Building Engineering*, ASCE.

Tak, S., Kim, S. & Yeo, H. (2014b), Travel time prediction for Origin-Destination pairs without route specification in urban network, in *Proceedings of the 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, October 8–11, 2014.

Tak, S., Woo, S. & Yeo, H. (2016), Data-driven imputation method for traffic data in sectional units of road links, *IEEE Transactions on Intelligent Transportation Systems*, DOI: 10.1109/TITS.2016.2530312.

Vlahogianni, E., Golias, J. & Karlaftis, M. (2004), Short-term traffic forecasting: overview of objectives and methods, *Transport Reviews*, **24**(5), 533–57.

Wen, Y. (2008), *Scalability of Dynamic Traffic Assignment*, Massachusetts Institute of Technology, MA.

Wen, Y., Lee, T. & Cho, H. (2005), Hybrid models toward traffic detector data treatment and data fusion, in *Proceedings of the 2005 IEEE Networking, Sensing and Control*, 525–30.

Wilder, B. (2012), *Cloud Architecture Patterns: Using Microsoft Azure*. O'Reilly Media, Inc., Sebastopol, CA.

Williams, B. (2001), Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling, *Transportation Research Record: Journal of the Transportation Research Board*, **1776**, 194–200.

Williams, B. & Hoel, L. (2003), Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results, *Journal of Transportation Engineering*, **129**(6), 664–72.

Yeo, H., Jang, K., Skabardonis, A. & Kang, S. (2012), Impact of traffic states on freeway crash involvement rates, *Accident Analysis & Prevention*, **50**, 713–23.

Yu, J., Chang, G., Ho, H. & Liu, Y. (2008), Variation based online travel time prediction using clustered neural networks, in *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC*, Beijing, China, October 12–15, 2008.

Zhang, X. & Rice, J. (2003), Short-term travel time prediction, *Transportation Research Part C: Emerging Technologies*, **11**(3), 187–210.

Zhou, X. & Mahmassani, H. (2007), A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework, *Transportation Research Part B: Methodological*, **41**(8), 823–40.