Online Estimation and Prediction of Large-Scale Network Traffic From Sparse Probe Vehicle Data

Shun Taguchi and Takayoshi Yoshimura, *Member, IEEE*

***Abstract*—Network traffic prediction based on probe vehicle data is important for traffic management and route recommendation and has been intensively studied. Previous traffic prediction methods mainly focused on recurring traffic congestion. Predicting non-recurring traffic congestion, caused by events and accidents, is significantly more important; however, it has not been intensively studied. To predict non-recurring traffic congestion using probe data, we need to estimate the current traffic conditions based on *sparse* observations for *large* traffic networks to track traffic changes *online*. Conventional traffic forecasting methods have not been able to solve all of these problems. To address these problems, we propose a data assimilation method using a state space neural network (SSNN) with an incorporated topology of road networks. The SSNN model can easily model network traffic and can easily estimate its states and parameters by data assimilation using Bayesian filtering. In this study, we adopted a decoupled extended Kalman filter (DEKF) based data assimilation, which is scalable and applicable to large-scale network traffic, to estimate the states and parameters online. We evaluate the proposed method using an open dataset that includes a road network comprising over 30000 road segments. The results show that our method achieves higher prediction accuracy for predicting unknown traffic congestion and is more robust against data sparsity than conventional state estimation methods.**

***Index Terms*—Data assimilation, decoupled extended Kalman filter, state space neural network, traffic speed prediction.**

I. Introduction

TRAFFIC state prediction, which estimates the future traffic speed, density, or travel time, is required for traffic management and route recommendation systems, and it has attracted considerable research interest for over three decades [1]. Prediction based on Global Positioning System (GPS) probe vehicles has been studied intensively in recent years, as these vehicles can cover a wider area than prediction based on loop detectors. In recent years, traffic congestion prediction approaches based on deep learning techniques have been intensively studied. Most of them have focused on recurring traffic congestion which can be learned by historical data. However, non-recurring traffic congestion, caused by events and accidents, are much more important for traffic management and route recommendation. Predicting

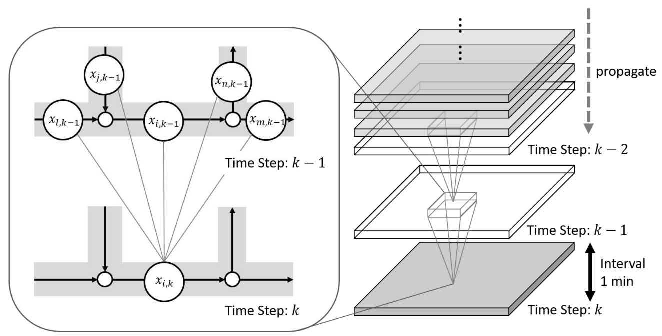


Fig. 1. **Proposed SSNN model that incorporates the topology of the road network.** The state $x_{i,k}$ depends on the previous states of all connected road segments and its own previous state. The time step interval is set short (1 min), and long prediction is performed through multiple layers of short predictions.

non-recurring traffic congestion based on probe vehicle data is extremely challenging because of the following three problems. (1) The estimation of the current edge states from sparse GPS data: If we can obtain abundant probe data, we can easily estimate each edge state. However, if the probe data is sparse, we have to interpolate the information temporally and spatially; (2) Short-term prediction of large-scale network traffic: The future state should be predicted using prediction models that can not only consider network connections but also be applied to large-scale networks; and (3) Online tracking traffic changes: To predict the state of non-recurring traffic congestions, the model needs to be flexible enough to accommodate changing traffic conditions caused by events and accidents using online learning.

We attempted to achieve accurate prediction and online learning using sparse data by employing data assimilation using a state-space neural network (SSNN) model that incorporates the topology of the road network. Machine learning models require large amounts of data and are difficult to learn with sparse data online. State-space models such as macro traffic models can identify the system online using data assimilation; however, the prediction performance of the traffic flow model often used is low. The SSNN model has properties of both a machine learning model and a state-space model. Therefore, the SSNN has high expression potential of the neural network model, and its parameters can be estimated by data assimilation methods online.

The proposed prediction model is outlined in Fig. 1. We enhance the SSNN model to incorporate the topology of the road network; each node corresponding to a road

Manuscript received 3 June 2019; revised 12 January 2020, 3 August 2020, and 10 February 2021; accepted 11 March 2021. Date of publication 30 March 2021; date of current version 8 July 2022. The Associate Editor for this article was W. Jin. *(Corresponding author: Shun Taguchi.)*

The authors are with the Toyota Central R&D Labs., Inc., Nagakute 480-1192, Japan (e-mail: s-taguchi@mosk.tytlabs.co.jp; yoshimura@mosk.tytlabs.co.jp).

Digital Object Identifier 10.1109/TITS.2021.3067603

segment depends on both its own previous state and the previous states of all connected road segments. The proposed method achieves long predictions through multiple layers of short predictions using adjacent edges. Conventional macro traffic models suffer from errors caused by the microscopic phenomena of urban traffic; however the proposed method can reduce such errors by learning the parameters sequentially. The number of parameters and states becomes very large when the proposed SSNN model is applied to large-scale road networks. They cannot be tracked by simple data assimilation methods such as extended Kalman filter (EKF) because the dimension of the variance matrix becomes very large. Therefore we use a Bayesian filtering method applicable to high-dimensional data, namely the decoupled extended Kalman filter (DEKF) [2], for data assimilation. The DEKF is an approximation method of the EKF that is applied to high-dimensional models, and it assumes that the interactions between certain weight estimates can be ignored. The DEKF is fast enough to estimate the traffic states of large-scale road networks in real time, and it can be processed in parallel.

The main contributions of our work are summarized as follows:

- We proposed a novel SSNN model that incorporates the topology of the road network. The proposed model has high expression potential of the neural network model, and its parameters can be estimated by data assimilation methods online. The introduction of the road network topology also allows us to model very large road networks.
- We employed DEKF-based data assimilation for estimating the states and parameters of the proposed model from sparse probe vehicle data in real time. By learning the model parameters online using data assimilation, the model can be adapted to non-recurring traffic congestion.

We evaluated the proposed method by employing open datasets used in the ICDM 2010 contest [3]. To evaluate the prediction performance of non-recurring traffic congestion, evaluation was performed in a much more difficult task than that of the ICDM 2010 contest, which was performed without using historical data. The results showed that the proposed method achieves higher accuracy for predicting non-recurring traffic congestion and robustness against sparseness compared with that achieved by conventional state estimation methods.

The remainder of this paper is organized as follows. Section II reviews related studies and discusses the similarities and differences compared to the present study. Section III introduces an SSNN-based network traffic model. Section IV describes the data assimilation method for estimating the states and parameters of the proposed traffic model. Section V presents the experimental results. Finally, Section VII concludes the paper.

II. Related Work

In the introduction, we identified three problems associated with predicting non-recurring traffic congestion from sparse probe vehicle data. Conventional methods have addressed only some of these problems. Estimation methods focus on only the first problem. Machine learning methods, such as deep-learning-based methods, focus on the second problem. However, most of these methods require dense probe data or external ground-truth data for training, and therefore, they do not work well without sufficient training data. Some researchers have proposed a learning method based on sparse probe vehicle data [36]–[38]; however, this method cannot track traffic changes. Data assimilation-based approaches, which employ macro traffic models, such as CTM, focus on the third problem. These approaches do not consider the sparseness of the data and they cannot be applied to large-scale network traffic. Table I summarizes the various approaches and the problems solved by these approaches. Each method is reviewed in more detail in the following sections.

TABLE I

Problems Solved by Various Approaches. in Order to Predict Non-Recurring Traffic Congestion Using Probe Data, We Have to Estimate Current Traffic Conditions Based on *Sparse* Observations for *Large* Traffic Networks to Track Traffic Changes *Online*

	Est.[a]	MLs[b]	SLs[c]	DA(CTM)[d]	Proposed
Sparse Estimation	✓	(✗)[e]	✓	(✗)[f]	✓
Network Prediction	✗	✓	✓	✗	✓
Online Learning	✗	(✗)[g]	✗	✓	✓

[a] Estimation methods
[b] Machine learning methods
[c] Sparse learning methods [36]–[38]
[d] Data assimilation methods based on a CTM
[e] Requires dense historical data for training
[f] Applied to only freeway traffic
[g] Requires sparse estimation

A. Estimation Methods

Estimation methods focus on the estimating edge states from sparse GPS data [4]–[10]. However, estimation without the use of dynamics requires the acquisition of additional information on the states of the road segments from sparse GPS coordinates. Hence, numerous estimation algorithms based on probe vehicle data that rely on the decomposition of path travel times into individual road segments have been proposed [6]–[9]. When a vehicle travels across more than one link, the location of its delay is unknown, and decomposition in such cases can lead to inaccuracies. This issue has also been discussed in [36]. In this study, we use only the instantaneous velocity of the probe vehicle data. The information on the instantaneous speed is spatially sparse because it can be obtained only for the road segment on which the probe is currently traveling. The proposed method solves this problem by modeling the dynamics between neighboring road segments. Wang *et al.* [10] proposed an approach that integrates a travel time inference with context-aware tensor decomposition and optimal concatenation using dynamic programming. Further, they suggested that the path travel time cannot be estimated by summing the travel times of individual road segments. In their approach, it is possible to estimate the route travel time by correcting the decomposition by estimating the subpath according to not only the travel time of each segment but also the combination of orbital patterns. Although estimation methods can estimate the traffic speed or travel time from sparse probe data, they cannot predict future states.

B. Machine Learning Methods

Machine learning methods focus on the short-term prediction of traffic states. Many related studies have not addressed the influence of the sparsity of GPS data; they targeted a motorway or a small network on which a large number of vehicles travel [11]–[13]. Short-term prediction has been studied for over three decades [1] using loop detectors. In the early stage, classical time-series approaches include the autoregressive integrated moving average (ARIMA) model [14]–[17] and its extended versions [18]–[20] are studied. After classical time-series approaches, several machine learning methods have been applied to traffic prediction. Various approaches have been proposed, such as k-NN based [21], local linear regression based [22], and Bayesian network based [23] approaches. The two most commonly used approaches are neural networks [24] and support vector regression (SVR) [25], which are reviewed in [26]. SSNN is a variant of neural network models that employ a neural network as a system model of the state-space model. Lint *et al.* proposed the SSNN-based approach [27] and its online learning solution [28] for freeway travel-time prediction.

In the current decade, deep-learning-based methods have been studied intensively. Deep belief network [29], [30] and stacked auto encoder [31], [32] have been employed in several studies. Some studies introduced spatial and temporal information to deep learning methods. Ma *et al.* proposed a method that employs a convolutional neural network (CNN) by treating the traffic state as an image [33]. Wang *et al.* introduced a recurrent-neural-network-based approach to CNN-based traffic prediction [34]. Yu *et al.* proposed a spatio-temporal graph convolutional networks [35]. These approaches considered the use of probe data; however, their assumptions are much denser than that in our study because they require precise traffic states for their learning procedures.

Machine learning methods mostly focuses on recurring traffc congestion that can be captured from historical data; therefore, it is difficult to predict non-recurring traffic congestion using these methods.

C. Sparse Learning Methods

Some researchers have proposed learning methods for sparse probe vehicle data. Hofleitner *et al.* [36] proposed a model based on a dynamic Bayesian network model. The state of each segment is assumed to be either congested or uncongested, and then, parameter estimation is performed using a simulation-based expectation-maximization (EM) approach. Jenelius and Koutsopoulos [37] proposed a spatial moving average (SMA)-based model and maximum-likelihood-based estimation. This method estimates the relations between neighboring road segments using a model based on the SMA. Further, they proposed a probabilistic principal component analysis (PCA) model to address the sparsity of GPS probe data [38], in which all parameters are trained using an EM algorithm. Although these data-driven approaches based on historical data improve prediction accuracy and robustness against sparsity, they face difficulties in tracking changes that do not appear in the historical data in the traffic flow. Therefore it is difficult to predict non-recurring traffic congestion using these methods. Our method can change the model parameters online to accommodate changing traffic conditions using the data assimilation approach.

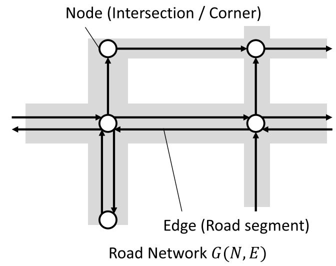


Fig. 2. **Road network defined as a directed graph.** Each node represents an intersection or a corner; each edge represents a road segment between nodes.

D. Data Assimilation Methods

Data assimilation, which is a useful approach for estimating the states and model parameters simultaneously, can track changes in the traffic flow. Data assimilation approaches for traffic flows have been studied previously [39]–[47]. For example, Work *et al.* [39] introduced a modified CTM and ensemble-Kalman-filter-based data assimilation. However, most of these approaches employ macro traffic flow models and they have been applied only to motorways. Traffic estimation methods for highways have been reviewed in [48]. Traffic on urban and rural roads is affected by factors such as route selection, traffic signals, and other microscopic environmental factors, which are not considered by macro traffic flow models. The SSNN-based proposed model can be applied easily to large-scale network traffic, and its parameters can be trained by streaming data that include the effects of various environmental factors.

III. SSNN-Based Traffic Modeling

In this study, we aim to predict the average velocity of vehicles on each road segment in a large-scale road network using sparse probe vehicle data. The target road network is a directed graph $G(N, E)$, where N is a set of nodes and E is a set of edges representing road segments (see Fig. 2). Each road segment $\varepsilon \in E$ has two properties: speed limit $\varepsilon.v_{\max}$ and a set of connected road segments including itself ε.connect.

The average velocities of the vehicles on the i-th road segment ε_i at time step k is denoted by $x_{i,k}$. The state $x_k = [x_{1,k}, \cdots, x_{|E|,k}]^T$ is a vector of average velocities of all road segments. The observation y_k represents the velocities of all probe vehicles obtained during a sampling interval. The traffic speed prediction problem is to predict state x_{k+h} using observation $y_{1:k} = \{y_1, \cdots, y_k\}$, which represents the velocities of the probe vehicles from time step 1 to time step k, where h is the prediction horizon.

We model this traffic model as the following state-space model:

$$x_k = f(x_{k-1}) + u_k,$$
$$y_k = H_k x_k + \upsilon_k, \tag{1}$$

where the elements of x_k and y_k are normalized by the speed limit of each road segment, f is the system model, u_k is the system error, H_k is the observation model, and υ_k is the observation error. Here, H_k can be modeled as a matrix whose elements are 1 if a probe vehicle is being driven on the segment and 0 otherwise. It is assumed that an observation is already mapped to a road segment by map-matching.

In this study, the system model f is modeled as a two-layer neural network model (i.e., it does not have any hidden layers):

$$f(x_{k-1}) = \varphi\left(W x_{k-1} + b\right), \tag{2}$$

where W is a matrix of neural network weight parameters, b is a vector of neural network bias parameters, and φ is the activation function. However, the dimension of W will be extremely high, as it is the square of the number of road segments. Therefore, we introduce the topology of the road network to the connections of the neural network, as shown in Fig. 1. Road segments that are unconnected in a road network are not connected in this neural network. This assumption means that the influence is transmitted along road connections, which is considered reasonable when the time step interval is set to a small value (e.g., 1 min). A longer prediction is performed by the repetition of this short-time prediction model. The model parameter W represents the magnitude of the effect from each connected road segment. The model parameter b is a bias factor. This model has the potential to fit a real traffic state by training the model parameters. By introducing this assumption, we can rewrite the state-space model as

$$x_{i,k} = \varphi\left(\sum_{j \in J_i} w_{ij} x_{j,k-1} + b_i\right) + u_{i,k},$$
$$y_k = \sum_i H_{i,k} x_{i,k} + \upsilon_k,$$
$$J_i = \left\{ j | \varepsilon_j \in \varepsilon_i.\text{connect} \right\}, \tag{3}$$

where $x_{i,k}$ is the state of the road segment ε_i at time step k, w_{ij} is an element of W, $u_{i,k}$ is the system error related to x_i, and H_i is the observation model for the road segment ε_i. Further, J_i is the set of road segments connected to ε_i. We introduce the following notation: $x_{J_i,k-1}$ denotes a vector that consists of a set $\left\{ x_{j,k-1} | j \in J_i \right\}$, and w_{iJ_i} denotes a vector that consists of a set $\left\{ w_{ij} | j \in J_i \right\}$. Therefore, (3) is rewritten as

$$x_{i,k} = \varphi\left(w_{iJ_i}^T x_{J_i,k-1} + b_i\right) + u_{i,k},$$
$$y_k = \sum_i H_{i,k} x_{i,k} + \upsilon_k. \tag{4}$$

Fig. 3 shows the dynamic network of this state-space model. This model can be regarded as a recurrent neural network (Elman network) that does not have an input layer, and for whom weights between the hidden layer and output layer are known and time variant.

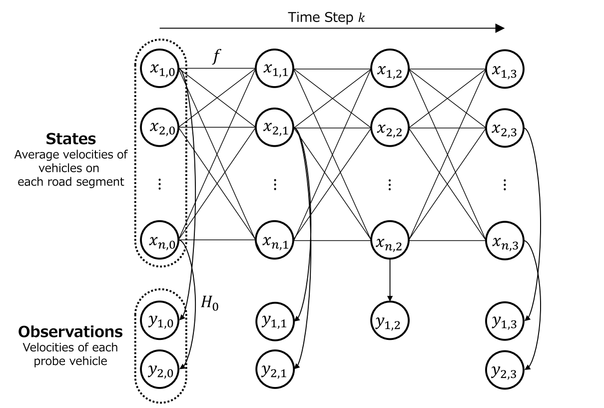


Fig. 3. **Dynamic network of the proposed state-space model.** The observations are obtained only from the edge on which the probe vehicle is driving at each step.

We use the following sigmoid function as the activation function.

$$\varphi(a) = \frac{1}{1 + \exp(-a)}, \tag{5}$$

where a represents any variable. As the sigmoid function can only take values from 0 to 1, the states and observation values are normalized by the speed limit of each road segment.

IV. Data Assimilation

In this study, the state x_k and parameters W and b are estimated by data assimilation. Data assimilation is an online method for simultaneously estimating states and parameters using Bayesian filtering, which is a recursive Bayesian estimation method that consists of the repetition of *prediction* and *update* steps. In the *prediction* step, the state x_k is predicted by the previous state x_{k-1} and the system model $p(x_k|x_{k-1})$:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1}, \tag{6}$$

where $y_{1:k} = y_1, \cdots, y_k$. In the *update* step, the posterior of the state x_k is updated by the observation y_k and observation model $p(y_k|x_k)$:

$$p(x_k|y_{1:k}) \propto p(y_k|x_k) p(x_k|y_{1:k-1}). \tag{7}$$

Estimation is performed by the repetition of these *prediction* and *update* steps. The state at step $k + h$ can be predicted by repeating the *prediction* step h times after estimation.

Data assimilation can be performed by general Bayesian filtering using an extended state-space model in which the parameters are contained in the state vector. We transform the state-space model in (4) into an extended state-space model as

$$x'_{i,k} = \begin{bmatrix} x_{i,k} \\ w_{iJ_i,k} \\ b_{i,k} \end{bmatrix} = \begin{bmatrix} \varphi\left(w_{iJ_i,k-1}^T x_{J_i,k-1} + b_{i,k-1}\right) \\ w_{iJ_i,k-1} \\ b_{i,k-1} \end{bmatrix} + u'_{i,k},$$
$$y_k = \sum_i H'_{i,k} x'_{i,k} + \upsilon_k, \tag{8}$$

where $x'_{i,k} = \left[x_{i,k} \ w_{iJ_i,k}^T \ b_{i,k} \right]^T$ is the extended state vector, $w_{iJ_i,k}$ and $b_{i,k}$ denote w_{iJ_i} and b_i at time step k, respectively, $u'_{i,k}$ is the extended system error, and $H'_{i,k}$ is the extended observation model.

Then, we apply Bayesian filtering to this model. We introduce a DEKF [2] to estimate $x'_{i,k}$. The DEKF is an approximation method of the EKF that is applied to high-dimensional models, and it assumes that the interactions between certain weight estimates can be ignored. We use the DEFK to separately estimate the posterior distribution of the expanded state-space vector $x'_{i,k}$ of each road segment ε_i. The DEKF assumes that the posterior distribution $p(x'_{i,k}|y_{1:k})$ is a Gaussian distribution $\mathcal{N}(\hat{x}'_{i,k|k}, P'_{i,k|k})$, where $\hat{x}'_{i,k|k}$ is the mean and $P'_{i,k|k}$ is the covariance matrix. In the DEKF, the *prediction* step is performed as

$$\begin{aligned}
\hat{x}'_{i,k|k-1} &= f'_i(\hat{x}'_{J_i,k-1|k-1}) \\
&= \begin{bmatrix} \varphi\left(\hat{w}_{iJ_i,k-1|k-1}^T \hat{x}_{J_i,k-1|k-1} + \hat{b}_{i,k-1|k-1}\right) \\ \hat{w}_{iJ_i,k-1|k-1} \\ \hat{b}_{i,k-1|k-1} \end{bmatrix}, \\
P'_{i,k|k-1} &= \sum_{j \in J_i} F'_{ij,k} P'_{j,k-1|k-1} F'^T_{ij,k} + Q'_i, \\
F'_{ij,k} &= \left. \frac{\partial f'_i}{\partial x'_{j,k-1|k-1}} \right|_{x'_{J_i,k-1|k-1} = \hat{x}'_{J_i,k-1|k-1}}, \qquad (9)
\end{aligned}$$

where $F'_{ij,k}$ is the Jacobian of f'_i for $x'_{j,k-1|k-1}$, Q'_i is the covariance matrix of the system error $u'_{i,k}$, and the mean of $u'_{i,k}$ is assumed to be 0. In the *update* step, the distribution is updated using the observation y_k.

$$\begin{aligned}
e_k &= y_k - \sum_i H'_{i,k} \hat{x}'_{i,k|k-1}, \\
S_k &= \sum_i H'_{i,k} P'_{i,k|k-1} H'^T_{i,k} + R_k, \\
K_{i,k} &= P'_{i,k|k-1} H'^T_{i,k} S_k^{-1}, \\
\hat{x}'_{i,k|k} &= \hat{x}'_{i,k|k-1} + K_{i,k} e_k, \\
P'_{i,k|k} &= (I - K_{i,k} H'_{i,k}) P'_{i,k|k-1}, \qquad (10)
\end{aligned}$$

where R_k is the covariance matrix of the observation error v_k and the mean of v_k is assumed to be 0. Further, $K_{i,k}$ is often called the Kalman gain, and I denotes the unit matrix. Thus, the posterior distribution $p(x'_{i,k}|y_{1:k})$ is estimated through the iteration of the *prediction* and *update* steps.

However, the dimension of the observation vector y_k in (10) is equal to the number of probe vehicles, and it is too large to calculate S_k^{-1}. Therefore, we split y_k into observations $y_{i,k}$ for the probe vehicles on each segment ε_i. The *update* step is performed for each segment ε_i sequentially.

$$\begin{aligned}
e_{i,k} &= y_{i,k} - H'_{ii,k} \hat{x}'_{i,k|k-1}, \\
S_{i,k} &= H'_{ii,k} P'_{i,k|k-1} H'^T_{ii,k} + R_{i,k}, \\
K_{ii,k} &= P'_{i,k|k-1} H'^T_{ii,k} S_{i,k}^{-1}, \\
\hat{x}'_{i,k|k} &= \hat{x}'_{i,k|k-1} + K_{ii,k} e_{i,k}, \\
P'_{i,k|k} &= (I - K_{i,k} H'_{ii,k}) P'_{i,k|k-1}, \qquad (11)
\end{aligned}$$

Fig. 4. **Traffic simulation framework: a micro traffic simulator used to generate datasets in the IEEE ICDM contest [3].** This simulator simulates the traffic in Warsaw, Poland. The road network contains 18716 nodes and 35170 edges.

where $H'_{ii,k}$ is the split observation model and $R_{i,k}$ is the split observation error. The calculation of the *prediction* and *update* steps can be performed using parallel processing. This method can thus be applied easily to a large-scale network.

V. Experimental Results

A. Datasets

For the evaluation, we used the *GPS task* dataset from the open datasets used in the IEEE ICDM 2010 contest [3], which comprises micro traffic simulation (see Fig. 4). In the *GPS task*, the algorithm receives a GPS stream, which is a stream consisting of latitude and longitude coordinates of the location with a time stamp and instantaneous velocity; the algorithm predicts the traffic speed on the selected road segments for the next 30 min. The input data comprise a stream of notifications from 1% of the vehicles sent every 10 s. The *GPS task* predicts the average speeds of the selected road segments using machine learning methods trained using historical probe data and ground-truth data. However, in this study, we perform the prediction online without using historical or ground-truth data. We used *street graph* and *training data* in this dataset.

Street graph consists of two types of information in one file.

- *Nodes*:
  - *Node ID*: unique for each node in a street graph
  - *Latitude*
  - *Longitude*
- *Edges*:
  - *Node 1 ID*: node ID of the source of the segment
  - *Node 2 ID*: node ID of the target of the segment
  - *Distance* [km]: length of the segment
  - *Number of lanes*: number of lanes in the segment
  - *Average max velocity* [km/h]: simulation parameter that indicates the velocity of free driving, which is nearly equal to the speed limit of the segment

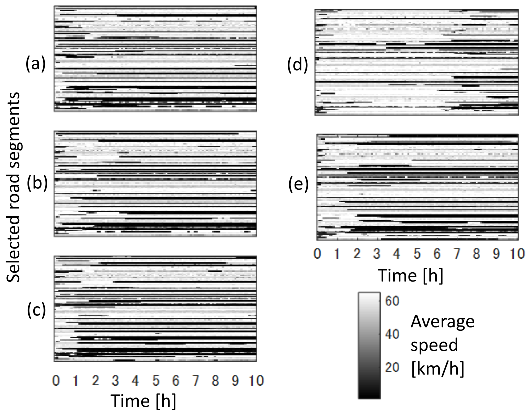


Fig. 5. **Actual speed on each selected road segment of each dataset (a)–(e).** Traffic congestion on the road segments is shown using dark color.

The road network $G(N, E)$ is constructed from the *street graph* data; it contains 18716 nodes and 35170 edges.

Training data consist of two files: the stream of data obtained from the vehicles, and the data about the actual average velocities of the selected road segments. Stream data have the following attributes:

- *Timestamp* [s]: time from the beginning of a simulation cycle
- *Car ID*: unique for each car in a given simulation
- *Velocity* [km/h]: instantaneous velocity of the car at a given second
- *Latitude*
- *Longitude*

Stream data do not contain the corresponding road segments; therefore, we map the stream data to road segments using a previously proposed online map-matching method [49] as a preprocessing step. This method can distinguish between upstream and downstream traffic flow. An overview of the map matching method has been provided in the next subsection.

Another file comprises data about the actual average velocities on 100 selected road segments. The average velocity data have the following attributes:

- *Timestamp* [s]: time from the beginning of a simulation cycle
- *Node 2 ID*: ID of the second node (terminal point) of the road segment
- *Node 1 ID*: ID of the first node (initial point) of the road segment
- *Velocity* [km/h]: harmonic average of the velocities of cars that passed or entered a given segment in the 6 min preceding the timestamp

The average velocity of each selected road segment of each dataset is shown in Fig. 5. Although the characteristics of each dataset are different, we can see that traffic congestion occurs on some road segments in all data.

The aim of the *GPS task* in the ICDM contest was to predict the traffic speed through some machine learning methods trained using the street data and average velocity data. The winner of the ICDM contest employed a random forest, which is a machine learning method [3].

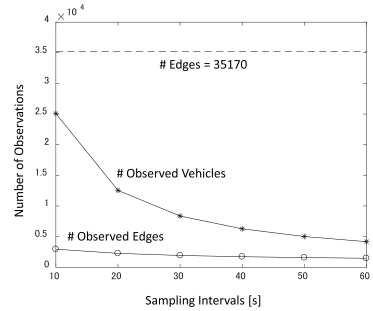


Fig. 6. **Sparsity of the observations for various sampling intervals.**

By contrast, in this study, traffic prediction is performed using only stream data without any offline training. The evaluation was performed using the average velocity data as the ground-truth data. The task undertaken in this study is more difficult than the *GPS task* in the ICDM contest in the following ways: (1) The ground-truth data of the average velocity data of the road segments are not available; therefore, we have to estimate them using only the stream of GPS data. (2) The task requires prediction of not only the velocities on the selected road segments but also the velocities on all road segments as the method does not have any specific information about the selected road segments. This assumption is more reasonable than that of the ICDM contest, as it is often not possible to obtain accurate data about the average velocity of the vehicles on a road segment if the road does not have any installed sensors.

Moreover, in this study, we evaluate the performance of the proposed method against sparseness for various sampling intervals by downsampling the streaming data. Robustness against sparseness of the sampling interval is important in terms of the cost of communicating GPS data. The sparsity of the observations for various sampling intervals is illustrated in Fig. 6, which shows the number of observed edges and vehicles in 6 min. Here, "# Observed edges" denotes the number of edges that have at least one observed vehicle on it. The observed edges are less than 10% of all the edges; thus, over 90% of the edges are not observed in 6 min. The proposed method estimates and predicts future traffic states only from these sparse probe vehicle data.

B. Map Matching

The proposed method requires mapping of GPS probe data to road segments. For this, we use our previously proposed method of map matching [49]. In this method, the matching probability for each road segment is calculated as follows:

$$
\begin{aligned}
p(r_k|g_{1:k}) &\propto p(g_k|r_k)p(r_k|g_{1:k-1}),\\
&\propto p(g_k|r_k)\sum_{r_{k-1}} p(r_k|r_{k-1})p(r_{k-1}|g_{1:k-1}), \qquad (12)
\end{aligned}
$$

where r_k represents the road segment on which the vehicle is traveling at time step k., and $g_{1:k}$ is a sequence of the k GPS data points collected by the vehicle. $p(r_k|r_{k-1})$ is the route prediction probability, which can be trained by the historical data. In this study, we used an untrained model, wherein the vehicle choose a connected road as the next road at random with equal probability. This algorithm predicts the probability of matching at step k based on the previous matching results estimated at step $k-1$, and then updates it from the actual GPS observation at step k. This method can map the GPS probe data, including the direction of travel, to the road segments by tracking the obtained GPS probes in time-series using multiple hypothesis tracking.

C. Setting

First, the step interval is set to 1 min, i.e., the traffic speed 1 min later is predicted in one layer of the neural network. The GPS probe data from the simulator can be obtained every 10–60 s; however, we assume that data accumulated for 1 min are observed at every step.

The covariance matrix of the system error Q'_i is time-invariant, the variance of the system error of the normalized average speed is set to 0.01^2, and the variance of the system error of the model parameter in the learning procedure is set to 0.1^2. The reason for including the error of the model parameter is to track the change of traffic. The variance of the observation error (i.e., the diagonal elements of the covariance matrix of the observation error R) is $(30/\varepsilon.v_{\max})^2$ [km^2/h^2]. The variance of the observation error is set to a large value, as the estimated state does not include the state of the traffic signal. This large variance absorbs the large speed difference occurring between green- and red-light signals.

In addition to setting these parameters, it is necessary to set the initial values of the states and parameters. The initial value of the states $x_{i,0}$, i.e., the average speed on the road segments, was set to 1.0 (i.e., the speed limit), and the variance was set to 10.0^2. If the road segment is not congested, vehicles on the segment can be driven at the speed limit; therefore, this initial value is a natural setting. Further, we need to set the initial value (i.e., prior distribution) of the parameters $w_{ij,0}$ and $b_{i,0}$. The parameter w_{ij} represents the effect of the previous state of the connected road segment ε_j on the current state of the road segment ε_i. Empirically, there is a positive correlation between them. The magnitude of the effects of the connected road segments is unknown; we defined the same prior distributions for all connected road segments of each road segment. Therefore, the initial values of the parameters $w_{ij,0}$ and $b_{i,0}$ were set as

$$\begin{aligned} w_{ij,0} &= 4.0/N_{J_i}, \\ b_{i,0} &= -2.0, \end{aligned} \tag{13}$$

where N_{J_i} represents the number of road segments connected to the road segment ε_i (i.e., $N_{J_i} = |\varepsilon_{i.\text{connect}}|$). The constant value is set such that the sigmoid function approaches the linear function $g(a) = a$. The initial variance of the parameters is set to 1.0^2. This initial parameter setting improves the speed of parameter training, which has been demonstrated in the last evaluation.

D. Results

Our simulator simulated traffic in Warsaw, Poland, and GPS data were obtained from approximately 1% of the vehicles. Five datasets published as sample data for the *GPS task* of the IEEE ICDM contest [3] comprised the simulation data for 10 h each. We compared the performance of the proposed method with that of the following four baseline methods.

- *Speed Limit*: Estimated using the speed limit of the road segments.
- *Simple Average*: Prediction based on the average speed of the GPS probe vehicles observed in 6 min. If a segment had no vehicle data, this was estimated using its speed limit.
- *Moving Average*: Prediction based on the average speed of the GPS probe vehicles observed in 1 h. If a segment had no vehicle data, this was estimated using its speed limit.
- *Kalman Filter*: Prediction with a constant velocity model (i.e., $x_{i,k} = x_{i,k-1} + u_{i,k}$) and Kalman filter. The initial states and system error were the same as those for the proposed method.

We compared our method with these simple estimation methods because there is no better conventional method that can estimate and predict such large-scale network traffic conditions from sparse GPS data in real time without any dividing technique.

The solutions were evaluated using the root-mean-square error (RMSE) of the inverted predictions:

$$RMSE = \sqrt{\frac{1}{n}\sum\left(\frac{60}{\varepsilon.v_{\text{gt}}} - \frac{60}{\max\{\varepsilon.v_{\text{est}}, 0.6\}}\right)^2}, \tag{14}$$

where n is the number of data points, $\varepsilon.v_{\text{gt}}$ is the velocity of the ground-truth data, and $\varepsilon.v_{\text{est}}$ is the estimated velocity. This equation denotes the RMSE of the predicted travel time over 1 km of the road segment, expressed in minutes. In this evaluation, the lower bound of the estimated velocity was set to 0.6 km/h, which is equal to the minimum velocity in the ground-truth data.

Fig. 7 and Table II present the prediction accuracy for various prediction horizons and sampling intervals, where the RMSEs were calculated using the latter half (5 h) of the data because the parameters were not sufficiently estimated in the first half of the data. These results show that the proposed method achieves the best accuracy under all conditions, in particular, it achieves significantly high performance for sparse data. Even at the shortest sampling interval (10 s), the proposed method is able to reduce the RMSE by 15% compared with the Kalman filter and 30% compared with the moving average. The RMSE for a 30 min prediction horizon with 10 s sampling interval of the proposed method is 9.18 min/km, and this accuracy approaches that of the winner of the ICDM contest (approximately 7 min/km), which uses offline training with ground-truth data. Fig. 7 also shows that, when the sampling interval is increased, the accuracy of the existing method, which considers the road segment as independent, is reduced considerably, whereas the proposed method maintains high accuracy. The RMSE of the proposed method with 1 min sampling interval was 40% lower than that of the other methods. This is because the proposed method uses a prediction model between neighboring road segments,

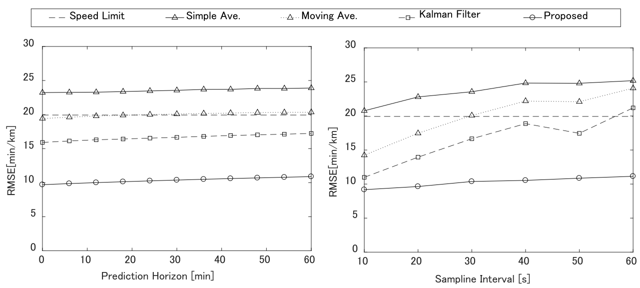


Fig. 7. **Prediction accuracy of the proposed methods.** The left graph shows prediction accuracy for each prediction horizon with 30 s sampling interval, and the right graph shows prediction accuracy for each sampling interval with 30 min prediction horizon.

TABLE II

PREDICTION ACCURACIES FOR VARIOUS PREDICTION HORIZONS AND SAMPLING INTERVALS

	RMSE [min/km]								
Horizion [min]	0			30			60		
Interval [s]	10	30	60	10	30	60	10	30	60
Speed Limit	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9	19.9
Simple Ave.	20.3	23.2	25.0	20.8	23.6	25.2	21.1	23.9	25.5
Moving Ave.	13.3	19.4	23.6	14.2	20.1	24.1	14.7	20.3	24.3
Kalman Filter	9.9	15.9	20.8	11.0	16.7	21.2	11.5	17.2	21.6
Proposed	**8.5**	**9.7**	**10.6**	**9.2**	**10.4**	**11.1**	**9.8**	**10.9**	**11.6**

TABLE III

CALCULATION TIME [ms/STEP] FOR EACH STEP (1 min) WITH MOST DENSE SETTING (I.E., 10 s SAMPLING INTERVAL)

	prediction	*update(estimate)*	Total
Simple Average	-	3.21	3.21
Moving Average	-	2.69	2.69
Kalman Filter	19.80	1.51	21.32
Proposed	257.68	20.42	278.10

and an observation contributes to the estimation of the states of the neighboring road segments. This result shows that the proposed method is particularly effective for sparse data.

The time-series results of the prediction accuracy with 30 s sampling interval are shown in Fig. 8. This figure shows that the accuracy of the proposed method is generally higher than that of the other methods, both for estimation and prediction. It is shown that the accuracy for 1 h predictions tends to be gradually increase with time. This can be attributed to the improvement of the prediction accuracy due to online training of the model. However, in case (d), the accuracy becomes lower, probably due to the appearance of a large traffic change along the way. Such a change does not cause a significant drop in performance compared to conventional methods.

Next, we demonstrate the efficiency of the proposed method. Table III summarizes the calculation time for each step of the proposed method, which is evaluated for the most dense setting (i.e., 10 s sampling interval). The algorithm was implemented in C# on the Visual Studio platform, and the results of the experiment were calculated using a single core of a computer with an Intel Core i7-4770K CPU running at 3.50 GHz with 32 GB of RAM. These results show that the proposed method takes about 10 times the computation time as the Kalman filter state estimation. Yet, the proposed method can be performed in real time because it can implement one step of the algorithm

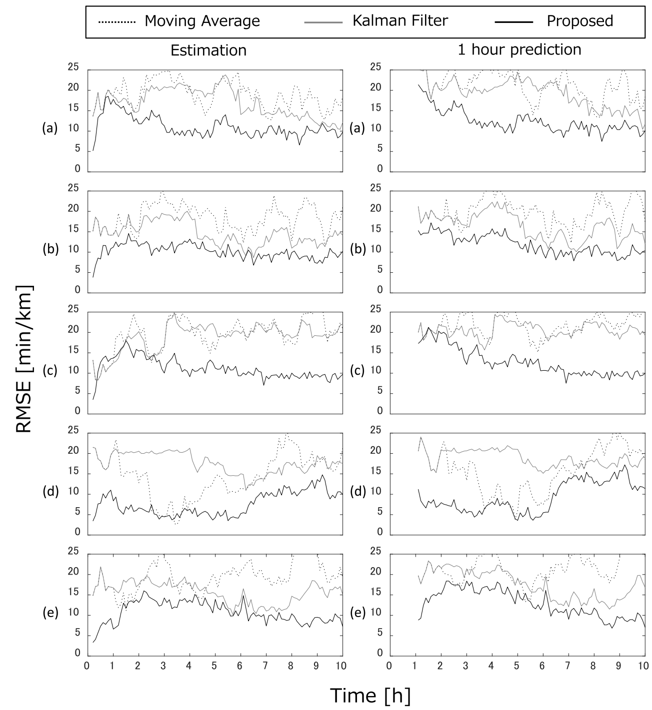


Fig. 8. **Time-series results of the prediction accuracy with 30 s sampling interval.** The left shows the estimation results, and the right shows the 1 h prediction results. Subfigures (a)–(e) show the prediction accuracy as the RMSE for five traffic simulation datasets.

(i.e., process 1 min of data) in about 300 ms. These results are associated only with the calculation times required for estimating the states and parameters. To predict the traffic states h min later (i.e., when the prediction horizon is h), h times additional *prediction* steps are required.

Next, we performed an ablation study to show the effectiveness of the proposed method by evaluating the differences between the linear and nonlinear models as well as those between the methods with and without data assimilation. For this purpose, we evaluated the following two methods.

- *LR+DKF (w/ DA)*: This method uses a linear regression model instead of a neural network. It is the same as the proposed model without the sigmoid-based activation function.
- *SSNN+DEKF (w/o DA)*: This method does not involve data assimilation. The DEKF estimates only the states and not the parameters, i.e., the parameters are invariant.

In LR+DKF (w/ DA), the initial values of the model parameters were set as $w_{ij,0} = 1.0/N_{J_i}$, $b_{i,0} = 0.0$. These initial values represent the average speeds between the connected road segments. We introduced these settings so that the initial model is close to the proposed method.

The results shown in Fig. 9 indicate that both the model and data assimilation are important for improving accuracy. The accuracy of the LR+DKF (w/ DA) is significantly poor, which is even less than that of the speed limit. The linear regression model, which is different from the sigmoid function of the neural network, has no bound; therefore, the prediction can become considerably larger than 1.0. This causes divergence

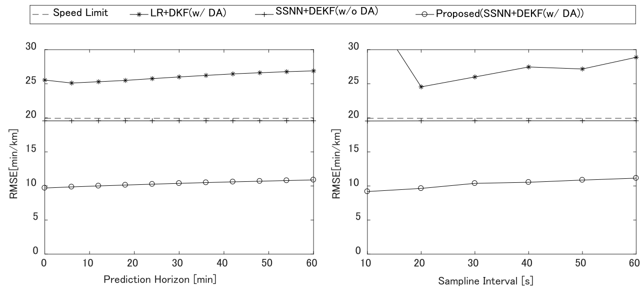


Fig. 9. **Effectiveness of the model and data assimilation.** Differences of accuracies between the proposed method (SSNN+DEKF (w/ DA)), the linear model (LR+DKF(w/ DA)), and the nonlinear model without data assimilation (SSNN+DEKF(w/o DA)) are shown.

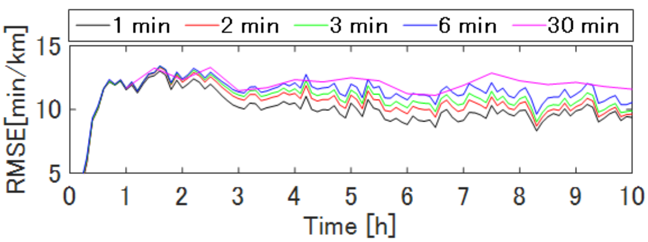


Fig. 10. **Effectiveness of multi-layered SSNNs.** Comparison of average RMSEs for the five datasets for various time step intervals (1, 2, 3, 6, and 30 min) evaluated for 30 min prediction with 30 s sampling intervals.

in the estimate. Further, this was observed when a rectified linear unit (ReLU) was used. An activation function with upper and lower bounds (e.g., a sigmoid function) is important for preventing divergence.

The accuracy of the SSNN+DEKF (w/o DA) is also poor, which is nearly the same as that of the speed limit. The initial parameters are insufficient to predict real traffic; parameter training via data assimilation is necessary to fit the model to real traffic. DEKF-based data assimilation can estimate appropriate parameters at that time online.

Next, we evaluate the effectiveness of multi-layered SSNNs (Fig. 10). In this figure, the average RMSEs for the five datasets for various time step intervals (1, 2, 3, 6, and 30 min) are shown, where the RMSE is evaluated for 30 min prediction with 30 s sampling intervals. This result indicates that the SSNN with shorter intervals (i.e., more layers) is relatively more accurate. We believe that the SSNNs with short intervals can learn from rapid traffic changes; and despite the simplicity of the model, which can only consider directly connected roads, it can consider distant roads via multi-layered prediction. The results in Fig. 10 support this reasoning.

Finally, we evaluate the initial parameter settings in (13) for the proposed method by comparison with uniform random initialization as $w_{ij,0} \sim \mathcal{U}(0,1)$, $b_{i,0} \sim -\mathcal{U}(0,1)$, where $\mathcal{U}(0,1)$ is the uniform distribution between 0 and 1. Further, we evaluate sensitivity for the initial value by varying it by $\pm 20\%$. The evaluation results for estimation using both initializations are shown in Fig. 11. In this figure, the average RMSEs for five datasets for various initial parameters are shown, where the RMSE is evaluated for the estimation with 10 s sampling intervals. The speed of parameter training

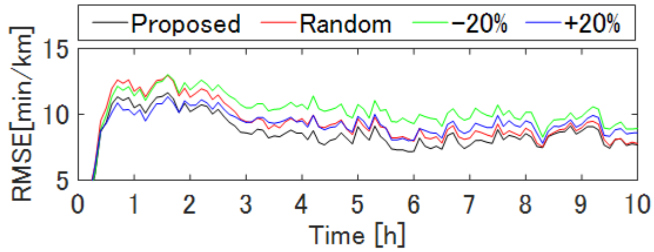


Fig. 11. **Sensitivity analysis of the initial parameter setting.** The initial parameter settings in (13) are compared with the random and the $\pm 20\%$ settings, where the RMSE is evaluated for the estimation with 10 s sampling intervals.

was improved using the proposed initial parameter settings. This is because the settings increase the correlation between the connected road segments and spread the effects of the observations. It is considered that, closer the initial value is to 0, the more this diffusion is suppressed, and the slower is the learning. This effect can be confirmed from the poor result when the initial value is set as -20% even in Fig. 11.

VI. Discussion

A. Limitation

The proposed method can be applied to any network where the connection structure is known. It only requires a network structure and an online map matching method (such as [49]) for mapping GPS data to upstream and downstream roads separately. Further, there is no need for offline training using the proposed method. At the beginning, traffic speed can be predicted with the same estimation accuracy as that of the Kalman filter. This initial estimation accuracy may be low due to the inability of introducing prior knowledge regarding traffic environments except the topology of road networks. However, the method learns sequentially and improves its accuracy. This also means that the method can be used effectively when there is insufficient data when starting the service. However, if the road structure changes, we have to reset the estimation, because the parameters trained from different road structures cannot be used. In addition, the proposed method focuses on non-recurring traffic congestion. It cannot predict in advance the long-term trends of regular events such as a morning rush because the current method only considers the accumulation of changes from the previous step.

B. Application to Real-Time Route Choice

By using this method, the future traffic speed of each road can be estimated. This can be applied to a real-time routing system. In particular, routing between the current position estimated by map matching and the destination input by users is performed using a route search technique such as the A* algorithm with time, and the predicted traffic speed is used as the cost of the route. Our method can provide travelers with more accurate arrival times and better routes than with the conventional method.

VII. Conclusion

This paper proposes a novel non-recurring traffic speed prediction method based on SSNN-based traffic model and data assimilation using DEKF for a large-scale road network based on sparse probe vehicle data. The proposed SSNN-based traffic model incorporated the topology of the road network as connections of neurons. The DEKF-based data assimilation can estimate states and model parameters of large-scale traffic from sparse probe vehicle data in real time, because of which the model can be adapted to non-recurring traffic congestion.

We evaluated our method using open datasets from the ICDM 2010 contest [3] for non-recurring traffic prediction without offline training. The results showed that our method could reduce the RMSE of the road segment travel time by 15% compared to the Kalman filter method with 10 s sampling interval. For a 30 min prediction horizon, the performance of our method approached that of the winner of the ICDM contest, even though our method does not perform any offline training using ground-truth data. Moreover, an evaluation of low-sampling-rate data showed that the proposed method is significantly more robust against sparseness than conventional estimation methods. The RMSE of the proposed method with 1 min sampling interval was lower than that of the other methods by 40%; this is probably because the proposed method has a prediction model between neighboring road segments. An evaluation of the efficiency showed that the proposed method can process over 30000 road segments in real time. In addition, the proposed method can be scaled by parallel processing. As our method can be used to estimate and predict traffic states in real time only with sparse probe vehicle data, it is expected to improve applications such as real time route recommendation systems.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[2] S. S. Haykin, *Kalman Filtering and Neural Networks*. New York, NY, USA: Wiley, 2001.

[3] M. Wojnarski, P. Gora, M. Szczuka, H. S. Nguyen, J. Swietlicka, and D. Zeinalipour, "IEEE ICDM 2010 contest: Tomtom traffic prediction for intelligent GPS navigation," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2010, pp. 1372–1376.

[4] T. Hunter, R. Herring, P. Abeel, and A. Bayen, "Path and travel time inference from GPS probe vehicle data," in *Proc. NIPS Analyzing Netw. Learn. With Graphs*, 2009, vol. 12, no. 1.

[5] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Travel time estimation for ambulances using Bayesian data augmentation," *Ann. Appl. Statist.*, vol. 7, no. 2, pp. 1139–1161, Jun. 2013.

[6] B. Hellinga, P. Izadpanah, H. Takada, and L. Fu, "Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments," *Transp. Res. C, Emerg. Technol.*, vol. 16, no. 6, pp. 768–782, Dec. 2008.

[7] J. Miller, S.-I. Kim, M. Ali, and T. Menard, "Determining time to traverse road sections based on mapping discrete GPS vehicle data to continuous flows," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 615–620.

[8] A. Hofleitner and A. Bayen, "Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 815–821.

[9] F. Zheng and H. Van Zuylen, "Urban link travel time estimation based on sparse probe vehicle data," *Transp. Res. C, Emerg. Technol.*, vol. 31, pp. 145–157, Jun. 2013.

[10] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 25–34.

[11] C. de Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 197–203.

[12] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, Aug. 2011.

[13] T. Cheng, J. Haworth, and J. Wang, "Spatio-temporal autocorrelation of road network data," *J. Geographical Syst.*, vol. 14, no. 4, pp. 389–413, Oct. 2012.

[14] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1994.

[15] C. K. Moorthy and B. G. Ratcliffe, "Short term traffic forecasting using time series methods," *Transp. Planning Technol.*, vol. 12, no. 1, pp. 45–56, Jul. 1988.

[16] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1678, no. 1, pp. 179–188, Jan. 1999.

[17] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic vol. in, urban Arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.

[18] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[19] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1857, no. 1, pp. 74–84, Jan. 2003.

[20] Y. Kamarianakis and P. Prastacos, "Space-time modeling of traffic flow," in *Proc. ERSA*, 2002, pp. 1–21.

[21] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *J. Transp. Eng.*, vol. 117, no. 2, pp. 178–188, Mar. 1991.

[22] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Use of local linear regression model for short-term traffic forecasting," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1836, no. 1, pp. 143–150, Jan. 2003.

[23] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[24] M. Jun and M. Ying, "Research of traffic flow forecasting based on neural network," in *Proc. 2nd Int. Symp. Intell. Inf. Technol. Appl. (IITA)*, vol. 2, Dec. 2008, pp. 104–108.

[25] H. Su, L. Zhang, and S. Yu, "Short-term traffic flow prediction based on incremental support vector regression," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, vol. 1, 2007, pp. 640–645.

[26] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[27] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. C, Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, Oct. 2005.

[28] J. W. C. van Lint, "Online learning solutions for freeway travel time prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 38–47, Mar. 2008.

[29] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1217–1222.

[30] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[31] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[32] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016.

[33] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

[34] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 499–508.

[35] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[36] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1679–1693, Dec. 2012.

[37] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transp. Res. B, Methodol.*, vol. 53, pp. 64–81, Jul. 2013.

[38] E. Jenelius and H. N. Koutsopoulos, "Urban network travel time prediction based on a probabilistic principal component analysis model of probe data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 436–445, Feb. 2018.

[39] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 5062–5068.

[40] X. Sun, L. Munoz, and R. Horowitz, "Highway traffic state estimation using improved mixture Kalman filters for effective ramp metering control," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, vol. 6, Dec. 2003, pp. 6333–6338.

[41] X. Sun, L. Munoz, and R. Horowitz, "Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application," in *Proc. Amer. Control Conf.*, 2004, pp. 2098–2103.

[42] L. Mihaylova and R. Boel, "A particle filter for freeway traffic estimation," in *Proc. 43rd IEEE Conf. Decis. Control*, vol. 2, Dec. 2004, pp. 2106–2111.

[43] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, Feb. 2005.

[44] D. Jacquet, M. Krstic, and C. C. de Wit, "Optimal control of scalar one-dimensional conservation laws," in *Proc. Amer. Control Conf.*, 2006, pp. 5213–5218.

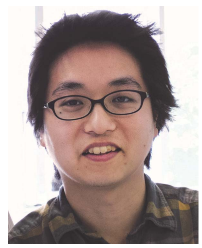
[45] R. Boel, L. Mihaylova, and A. Hegyi, "Freeway traffic estimation within recursive Bayesian framework," *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.

[46] J. Herrera and A. Bayen, "Traffic flow reconstruction using mobile sensors and loop detector data," in *Proc. 87th TRB Annu. Meeting*, 2008.

[47] Y. Yuan, J. W. C. van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.

[48] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annu. Rev. Control*, vol. 43, pp. 128–151, 2017.

[49] S. Taguchi, S. Koide, and T. Yoshimura, "Online map matching with route prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 338–347, Jan. 2019.



Shun Taguchi received the B.E. and M.E. degrees from Nagoya University, Japan, in 2007 and 2009, respectively. In 2009, he joined the Toyota Central R&D Labs., Inc., Japan. His current research interests include Bayesian filtering, data assimilation, and their applications to intelligent transport systems.

Takayoshi Yoshimura (Member, IEEE) received the B.S. degree from the Department of Computer Science, Nagoya Institute of Technology (NIT), Japan, in 1997, and the M.Eng. and Dr.Eng. degrees from the Department of Electrical and Computer Engineering, NIT in 1999 and 2002, respectively. He is currently a Senior Researcher with the Toyota Central R&D Labs., Inc., Japan. His research interests include in-vehicle signal processing, data-driven stochastic modeling, and their application to in-vehicle information systems.