Noname manuscript No.
(will be inserted by the editor)

Trajectory Data-Driven Network Representation for Traffic State Prediction using Deep Learning

Shohei Yasuda · Hiroki Katayama · Wataru Nakanishi · Takamasa Iryo

Received: date / Accepted: date

Abstract In this study, we propose a trajectory data-driven network representation method, specifically leveraging directional statistics. This approach allows us to extract major intersections and define links from observed trajectories, thereby mitigating the reliance on existing network data and map matching. We apply Graph Convolutional Networks and Long-Short Term Memory models to the trajectory data-driven network representation, suggesting the potential for fast and accurate traffic state prediction. The results imply significant reduction in computational complexity while demonstrating promising prediction accuracy. Our proposed method offers a valuable approach for analyzing and modeling transportation networks using real-world trajectory data, providing insights into traffic patterns and facilitating the exploration of more efficient traffic management strategies.

Keywords Network Representation · Traffic state prediction · Deep learning

1 Introduction

In transportation network analysis, the representation of the network is a crucial and challenging problem. Depending on the purpose of the analysis, such as national-level network design or identification of bottlenecks in local narrow roads, the required spatial resolution varies significantly. Therefore, careful consideration is needed regarding the resolution at which calculations should be performed for each analysis purpose. The computational cost associated with network-based calculations fluctuates rapidly with the number of links, making it necessary to consider the feasibility of computation when setting the resolution. However, there are no clear rules or algorithms for appropriately representing networks for various purposes. Most studies either use conceptual networks designed subjectively by analysts or meticulously adjust network data obtained from detailed networks made for navigation systems. Decades ago, the lack of discussion on how to place centroids and represent road network elements in traffic assignment problems was noted, despite the significant influence of these aspects on analytical outcomes [1]. Thereafter, techniques to simplify the network structure were proposed in the context of reducing the computational burden of the traffic assignment problem[2] [3] [4]. Even today, however, there are still few studies focusing on network representation, making it an issue that requires active engagement.

In recent years, with the rapid advancement of observational data, the trend in network data used for analysis is changing. In particular, with the proliferation of vehicle trajectory data, the demand for network data containing coordinate information is increasing. Trajectory data, also known as network-free data, is originally a time series of latitude and longitude, requiring additional processing to associate it with a network through techniques like map-matching. To associate vehicle observational data with a network, it is necessary for the network data to have positional co-

S. Yasuda
Department of Civil Engineering, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113–8656, Japan
E-mail: s.yasuda@civil.t.u-tokyo.ac.jp

H. Katayama
E-mail: katayama@trip.t.u-tokyo.ac.jp

W. Nakanishi
E-mail: nakanishi.se.kanazawa@gmail.com

T. Iryo
E-mail: iryo@tohoku.ac.jp

ordinate information. However, network data based on position coordinates, such as OpenStreetMap (OSM), which accurately represents detailed connection structures including narrow roads, often results in a massive number of nodes. For example, intersections or interchanges that provide route selection functionality are often represented by numerous nodes, which is redundant when considering wide-area network assignment problems. In such situations, the adjustment work for network data becomes increasingly burdensome in order to fully utilize valuable observational data that has become available in recent years.

Network representation methods that consider the characteristics of trajectory data have the potential to solve these problems. For example, network aggregation techniques based on trajectory data [5] [6] [7] objectively reconstruct networks based on actual observations, demonstrating their effectiveness, particularly for accelerating computational processes for large-scale networks. However, network aggregation currently relies heavily on the connection structure of the input network data and the accuracy of pre-executed map-matching processes, posing significant challenges. Map-matching is especially computationally intensive, and frequent mismatches still occur in networks with parallel roads. To address these issues, the development of new methodologies that reduce dependence on given network data and map-matching processes is required.

To decrease reliance on given network data, it is necessary to develop new methodologies for generating network data based on observations. One such method currently being developed is to estimate major intersections solely from trajectory data (Zhong et al., 2022). This method can be considered purely observation-based as it does not use any given network data. However, this approach assumes situations where there is absolutely no information about spaces where roads and intersections exist, such as in developing countries where there is no existing map data. From a practical standpoint, it is rational to adopt a strategy that does not rely on the connection structure of the given network data but leverages preliminary information such as the location and shape of intersections.

In this study, we develop a methodology to generate network data from trajectory data without using connection information from given network data, assuming situations where all intersection positions and degrees can be obtained in advance from maps or other sources. Specifically, we utilize the direction information and arrival information between points in trajectory data and develop major intersection extraction technology and network generation technology based on directional statistics. By generating networks from actually observed trajectory data, metrics for each generated link, such as average speed, can be calculated without the need for additional map-matching processes. By representing links based on actual arrival information, errors in link connection information and directions, which are common in conventional network data, can be avoided. The proposed method allows for adjusting the network resolution by adjusting threshold values, making it easy to create a network with a resolution suitable for the purpose and accuracy. To validate the utility of the proposed method, we perform network data generation using actual observational data and evaluate the accuracy of traffic state prediction based on deep learning.

2 Methodology

2.1 Trajectory data-driven network representation

We assume that information K regarding the coordinates and degrees of all intersections within a target area is available as prior knowledge. K is defined as follows:

$$K = \left\{ (x_i^K, y_i^K, d_i^K) \mid i = 1, 2, \ldots, n \right\}, \tag{1}$$

where x_i^K and y_i^K are the coordinates of intersection i, d_i^K is the degree of intersection i, and n is the number of intersections within the target area. The proposed method generates network data only from K and trajectory data:

$$T = \left\{ (x_{j,h}^T, y_{j,h}^T, c_{j,h}^T) \mid j = 1, 2, \ldots, m_h; h = 1, 2, \ldots, u \right\}, \tag{2}$$

where $x_{j,h}^T$ and $y_{j,h}^T$ are the coordinates of a dot observed at the j-th point of the trajectory for vehicle h, $c_{j,h}^T$ is the timestamp observed at the j-th point of the trajectory for vehicle h, m_h is the number of dots for vehicle h, and u is the number of vehicles whose trajectories could be observed in the target area.

In this study, intersections with a certain number of vehicles traversing in three or more directions are considered to play an important role as points for route choice, and the set of such intersections is defined as major intersections K^M. We describe how to extract major intersections $k^M \in K^M$ from the set of all intersections K using directional statistics. As a preprocessing step, for each dot $t_{j,h} \in T$, an azimuth angle $a_{j,h}$ to the next dot $t_{j+1,h}$ is calculated. This process is not performed for the last observed dot of each vehicle. We extract the set T_i of dots within a distance s for each intersection $k_i \in K$. T_i is described as follows:

$$T_i = \left\{ t_i \mid \text{distance}(k_i, t_i) \leq s \right\}, \tag{3}$$

where distance(k_i, t_i) represents the distance between point k_i and point t_i. For each k_i, let A_i be the set of azimuth angles calculated for all t_i.

$a_i \in A_i$ can be regarded as sampled from the distribution of azimuth angles in the direction of vehicles traversed within a distance s from intersection k_i. Assuming that the azimuth angles of vehicles traversing from each intersection to the links extending in each direction follow a normal distribution, a_i follows a mixed normal distribution with the degree d_i^K of the corresponding intersection as the mixture number. The mathematical expression is as follows:

$$a_i \sim \sum_{g=1}^{d_i^K} \pi_{i,g} N\left(\mu_{i,g}, \sigma^2\right), \sum_{g=1}^{d_i^K} \pi_{i,g} = 1, \tag{4}$$

where $\mu_{i,g}$ is the mean of each normal distribution included in the mixed normal distribution, and $\pi_{i,g}$ is the weight of each normal distribution. For simplicity, in this study, the variance of each normal distribution is assumed to be a fixed value σ^2. If we can estimate $\pi_{i,g}$, the number of vehicles traveling in each direction can be calculated by multiplying it by the total number of vehicles $|T_i|$ that passed near each intersection k_i. In this study, these parameters are estimated using the Expectation-Maximization (EM) algorithm. Using the threshold τ_1 for major node extraction, an intersection k_i is considered a major intersection k^M if there are at least three $\pi_{i,g}$ values that satisfy:

$$|t_i| \, \pi_{i,g} \geq \tau_1. \tag{5}$$

We will explain the process of defining links between major intersections. A threshold τ_2 is defined for link definition. We trace the trajectories of vehicles that traversed within the target area in a specific order. When a vehicle passes through a range within a distance s from a major intersection and subsequently passes through a range within a distance s from a different major intersection, we store information about these pairs of major intersections, the order of passage, and the timestamp.

This process is repeated for all vehicle trajectories, and for major intersection pairs where the number of vehicle passages exceeds τ_2, links are defined based on the direction of passage. For all vehicles used in defining the links, we calculate the average speed of the respective link by dividing the total distance between the dots for the given major intersection pair by the total travel time as defined by Edie [8].

2.2 Evaluation for traffic state prediction with deep learning approach

In this study, we utilize the proposed network representation to perform traffic state prediction using a deep learning approach and assess its accuracy. Our approach combines Graph Convolutional Network (GCN), which is well-suited for learning from data with graph structures, and Long-Short Term Memory (LSTM), which is effective for capturing time-evolving patterns. We evaluate the accuracy of traffic state prediction using this methodology.

GCN is a method that handles the spatial correlation of inputs by convolving only the features of neighboring nodes for each node. By repeating this convolution operation, the features of nodes as far apart as the number of iterations are convolved. This convolution operation is defined as an approximation of the graph Fourier transform using the graph Laplacian. The output $\mathbf{H}^{(l)}$ of the $l + 1$th layer is represented as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}\right), \tag{6}$$

where $\mathbf{H}^{(l)}$ is the output of the lth layer, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is a self-connected adjacency matrix, $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$ is a self-connected degree matrix, $\mathbf{A}$ is the adjacency matrix, $\mathbf{D}$ is the degree matrix, $\mathbf{W}^{(l)}$ is the weight matrix of the lth layer, and σ is the activation function.

LSTM is a type of recurrent neural network that can capture long-term dependencies and is well-suited for sequential data. It consists of input, forget, and output gates, as well as a memory cell. The equations for LSTM are as follows:

$$\boldsymbol{f}^t = \sigma\left(\mathbf{W}_{\mathrm{f}}[\boldsymbol{x}^t, \boldsymbol{h}^{t-1}] + \boldsymbol{b}_f\right) \tag{7}$$

$$\boldsymbol{i}^t = \sigma\left(\mathbf{W}_{\mathrm{i}}[\boldsymbol{x}^t, \boldsymbol{h}^{t-1}] + \boldsymbol{b}_i\right) \tag{8}$$

$$\boldsymbol{g}^t = \tanh\left(\mathbf{W}_{\mathrm{g}}[\boldsymbol{x}^t, \boldsymbol{h}^{t-1}] + \boldsymbol{b}_g\right) \tag{9}$$

$$\boldsymbol{c}^t = \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + i^t \odot \boldsymbol{g}^t \tag{10}$$

$$\boldsymbol{o}^t = \sigma\left(\mathbf{W}_{\mathrm{o}}[\boldsymbol{x}^t, \boldsymbol{h}^{t-1}] + \boldsymbol{b}_o\right) \tag{11}$$

$$\boldsymbol{h}^t = \boldsymbol{o}^t \odot \tanh\left(\boldsymbol{c}^t\right), \tag{12}$$

where the input at the $t - 1$th step is $\boldsymbol{h}^{t-1}$, the weight matrix is $\mathbf{W}_*$, the bias is $\boldsymbol{b}_*$, and the adamantine product is $\odot$, $\boldsymbol{f}^t, \boldsymbol{i}^t, \boldsymbol{g}^t, \boldsymbol{c}^t, \boldsymbol{o}^t$ are output of the gates and the concatenation of matrices is $[\cdot, \cdot]$.

In this study, the output goes through the GCN layer twice at each time step, followed by two LSTM layers, and finally a fully connected layer to obtain the prediction result (Figure 1). This model is based on the Temporal Graph Convolutional Network (T-GCN) [9], which replaces the Gated Recurrent Unit (GRU) in T-GCN with LSTM. A dropout layer is added before the fully connected layer to mitigate overfitting.

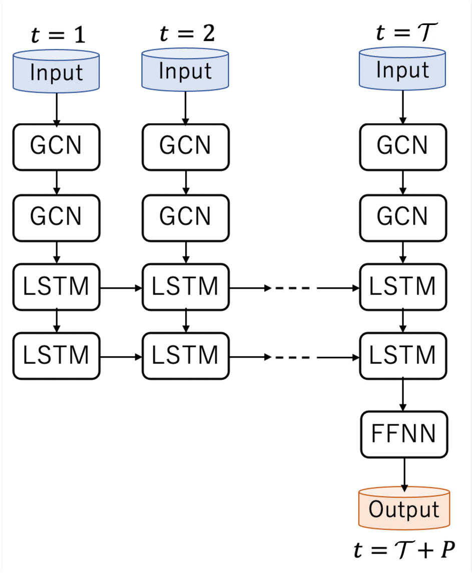

Fig. 1 The architecture of GCN-LSTM. We obtain the prediction result after P time steps based on inputs from time step 1 to $\mathcal{T}$.

3 Empirical Validation

3.1 Target and Dataset

We conducted empirical validation using actual observation data in Kobe area of Japan (mesh code 523502). The traffic observation data used in this study consisted of Electronic Toll Collection System (ETC) 2.0 data, which is vehicle trajectory data collected by the Ministry of Land, Infrastructure, Transport and Tourism of Japan, and detector data from the Hanshin Expressway Company. In addition, we compared the characteristics of the network data constructed in this validation with those of commonly used network data by using OSM data.

The ETC 2.0 data consisted of dot data observed from 00:00, November 1, 2020, to 23:59, November 30, 2020. The detector data focused on one upstream and one downstream detector between interchanges on the Hanshin Expressway. In cases where multiple lanes had detectors, the leftmost lane was selected for analysis. The OSM data within the target area was obtained using the Overpass API. We extracted the road segments that were accessible to automobiles and used them as our target network. The traffic state data were aggregated at 15-minute intervals. For the aggregation, the speed was calculated using the harmonic mean with traffic volume as weights, while traffic volume and occupancy were averaged. After the aggregation, we performed linear interpolation in the temporal direction for each detector and road link. For missing values at the edges, we replaced them with the nearest non-missing value at the edge. In this validation, the prediction target is the occupancy of each detector.

3.2 Parameters

We will explain the parameters used for validation. The parameter τ_1 for extracting major nodes was set at 50, 100, 200, and 500 (veh/day) in a stepwise manner, and the network generation results and traffic state prediction results were compared for each value. The parameter τ_2 for link definition was fixed at 50 (veh/day). The distance s used for determining the passage of trajectories near intersections was set to 30 (m). The variance σ^2 of each normal distribution within the mixture normal distribution was fixed at 15. The parameters $P = 4$ and $\mathcal{T} = 10$ mean that the forecast was made on a 4-step time scale based on the last 10 steps of observed data. The training period was from 00:00 on November 1, 2020, to 23:59 on November 23, 2020, and the validation period was from 00:00 on November 24, 2020, to 23:59 on November 30, 2020.

3.3 Result of Network representation

We show the results of the trajectory data-driven network data generated for each value of τ_1 compared with the original OSM data. Figure 2 - 6 show the visualization of OSM network data and the trajectory data-driven network data. Table 1 shows the number of nodes and links for each dataset.

The results demonstrate the ability to adjust the resolution of the network appropriately by tuning the threshold value, tau1. In comparison to the original OSM data, significant reductions in the number of links and nodes have been achieved. By carefully selecting tau1 during the major node extraction process, the network can be simplified while retaining the essential connectivity and structural characteristics. This reduction in links and nodes provides computational benefits, as it reduces the complexity and computational cost associated with network-based calculations. The trajectory data-driven network representation offers a more streamlined and efficient network representation compared to the original OSM data, without sacrificing the essential information required for transportation analysis.

3.4 Result of Traffic state prediction

We present the results of traffic state prediction using the trajectory data-driven network generated in this

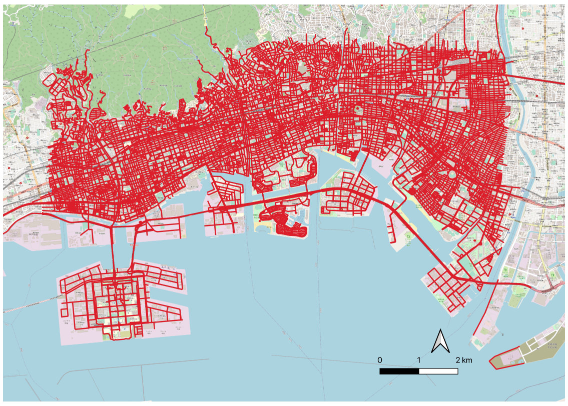

Fig. 2 Original OSM Data

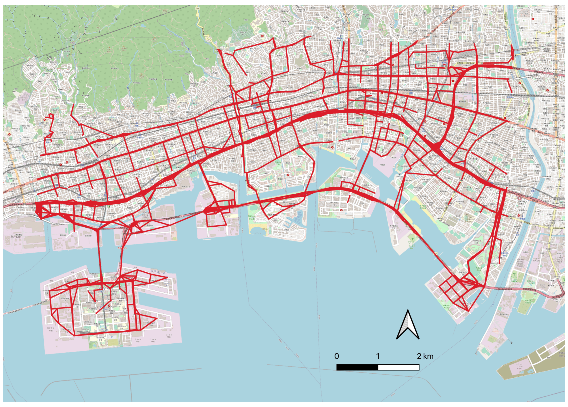

Fig. 3 Trajectory Data-Driven Network Data ($\tau_1 = 50$)

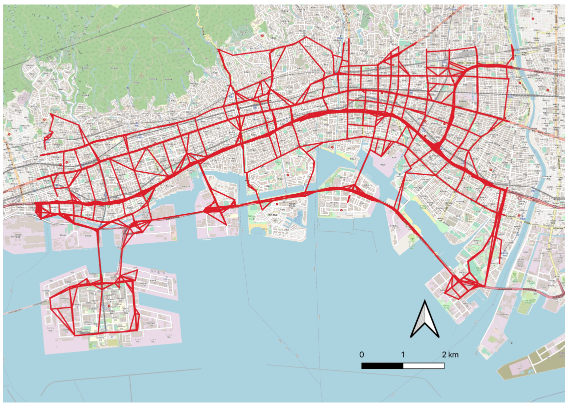

Fig. 4 Trajectory Data-Driven Network Data ($\tau_1 = 100$)

Table 1 Number of Nodes and Links for Each Result

τ_1	nodes	edges
50	1975	6949
100	1693	6874
200	999	3033
500	692	4566

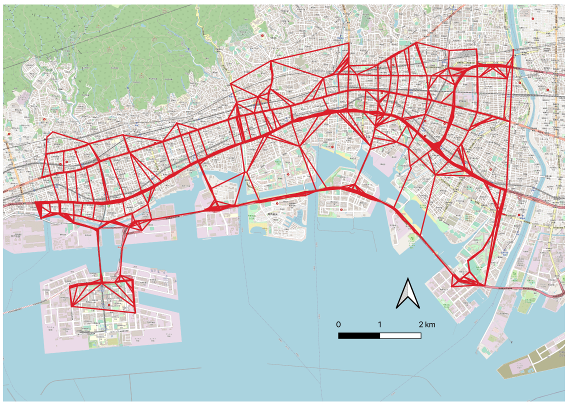

Fig. 5 Trajectory Data-Driven Network Data ($\tau_1 = 200$)

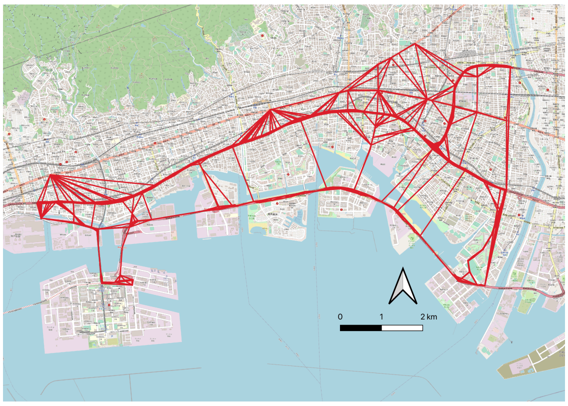

Fig. 6 Trajectory Data-Driven Network Data ($\tau_1 = 500$)

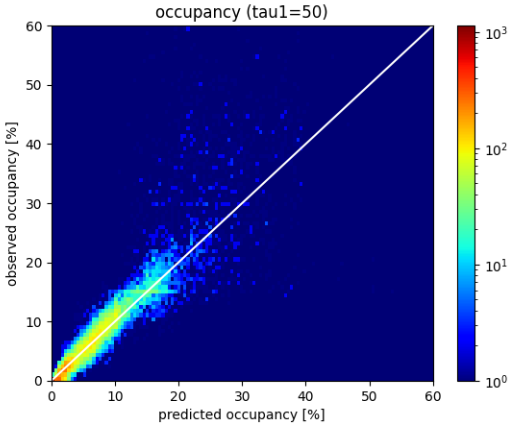

Fig. 7 Occupancy Prediction Results ($\tau_1 = 50$)

validation. Table 2 shows the accuracy of the occupancy projections. Figures 7 - 10 show the prediction accuracy of occupancy for τ_1=50,100,200,500. Figures 11 - 13 show the observed and predicted occupancy for each detector for one day on November 24, 2020, when τ_1=50, 100, 200, and 500. These results show that the proposed method is promising in terms of accuracy as well as significant reduction in computational speed in traffic condition prediction.

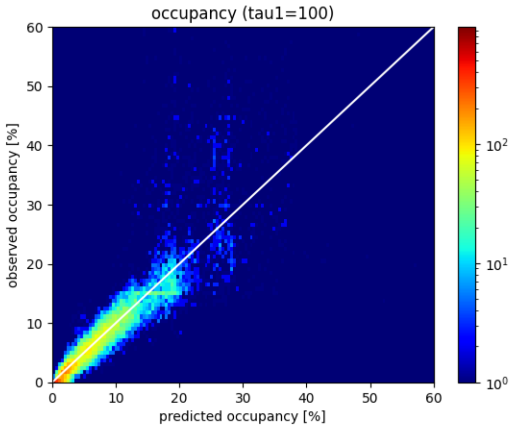

Fig. 8 Occupancy Prediction Results (τ_1 = 100)

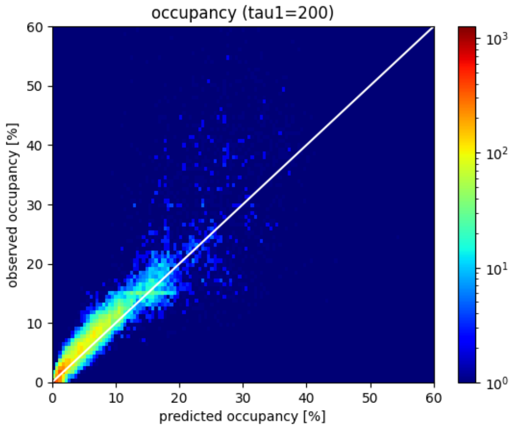

Fig. 9 Occupancy Prediction Results (τ_1 = 200)

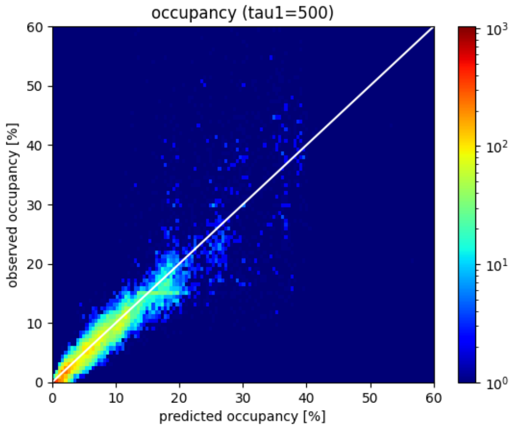

Fig. 10 Occupancy Prediction Results (τ_1 = 500)

4 Discussion and Conclusion

In this study, we proposed a methodology for generating network data from trajectory data and demonstrated its utility for traffic state prediction. By leveraging trajectory data and directional statistics, we were able to

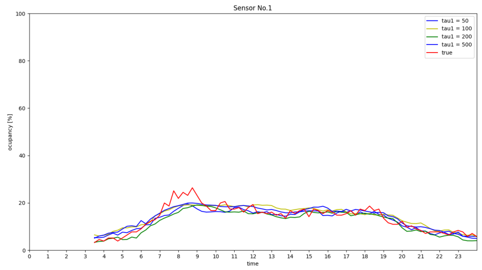

Fig. 11 Occupancy Prediction Results (Sensor No.1)

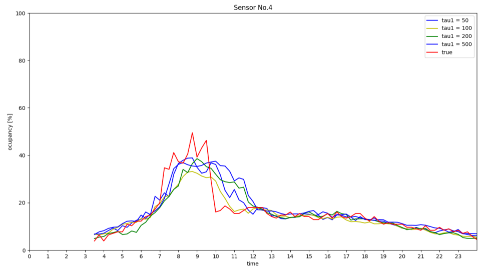

Fig. 12 Occupancy Prediction Results (Sensor No.4)

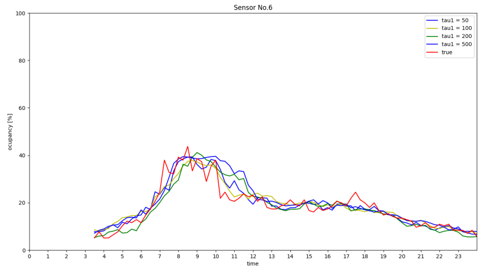

Fig. 13 Occupancy Prediction Results (Sensor No.6)

Table 2 Accuracy of occupancy prediction

tau1	50	100	200	500
MAE	2.003	2.167	2.231	2.016
RMSE	3.503	3.662	3.712	3.420
MAPE	20.598	22.184	21.690	22.101

extract major intersections and define links without relying on pre-existing network data. This approach has several advantages over traditional methods that heavily depend on given network data and map-matching processes.

One of the key contributions of this study is the ability to generate network data with a resolution suitable for specific analysis purposes based on actual observation data. By adjusting the threshold values, we were able to control the level of aggregation in the network representation. This flexibility allows researchers and practitioners to tailor the network resolution to the specific needs of their analysis, whether it is national-level network design or identifying local bottlenecks. The ability to adjust the resolution also reduces the computational burden associated with network-based calculations.

Furthermore, the trajectory data-driven network representation demonstrated promising results in traffic state prediction using deep learning. This representation eliminates the requirement for additional map-matching processes, simplifying the data processing pipeline. The empirical validation results showcased the method's potential in terms of accuracy and significant reduction in computational speed for traffic condition prediction.

List of Acronyms

EM Expectation-Maximization
GCN Graph Convolutional Network
LSTM Long-Short Term Memory
T-GCN Temporal Graph Convolutional Network
GRU Gated Recurrent Unit
ETC Electronic Toll Collection System
OSM OpenStreetMap

Acknowledgements This work was supported by Hanshin Expressway Co., Ltd. and JSPS KAKENHI Grant Number 21K14266.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. C.F. Daganzo, Transportation Research Part B: Methodological **14**(3), 221 (1980)
2. I. Wright, Y. Xiang, L. Waller, J. Cross, E. Norton, D. Van Vliet, in *European Transport Conference, 2010Association for European Transport (AET)* (Citeseer, 2010)
3. S.D. Boyles, Transportation Research Part B: Methodological **46**(1), 139 (2012)
4. K. Wada, K. Satsukawa, M. Smith, T. Akamatsu, Transportation Research Part B: Methodological **126**, 391 (2019)
5. G. Casadei, V. Bertrand, B. Gouin, C. Canudas-de Wit, Transportation Research Part C: Emerging Technologies **95**, 713 (2018)
6. S. Yasuda, T. Iryo, K. Sakai, K. Fukushima, IEEE Intelligent Transportation Systems Conference (ITSC) pp. 1677–1682 (2019)
7. H. Katayama, S. Yasuda, T. Fuse, International Journal of Intelligent Transportation Systems Research **20**(3), 830 (2022)
8. L.C. Edie, *Discussion of traffic stream measurements and definitions* (Port of New York Authority, 1963)
9. L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, IEEE Transactions on Intelligent Transportation Systems **21**(9), 3848 (2019)