Sparse Mobile Crowdsensing for Cost-Effective Traffic State Estimation With Spatio–Temporal Transformer Graph Neural Network

Jianzhe Xue, *Graduate Student Member, IEEE*, Yunting Xu, *Graduate Student Member, IEEE*, Wen Wu, *Senior Member, IEEE*, Tianqi Zhang, *Graduate Student Member, IEEE*, Qinghong Shen, Haibo Zhou, *Senior Member, IEEE*, and Weihua Zhuang, *Fellow, IEEE*

***Abstract*—Recently, mobile crowdsensing (MCS) has emerged as a promising solution for traffic state estimation (TSE), which provides real-time and accurate traffic information for supporting diversified intelligent transportation systems (ITSs) applications. However, the prohibitive overhead of collecting massive data in vehicular networks limits the available data amount, while the sparsification of MCS data incurs instability and degrades TSE accuracy. To this end, this article proposes a novel sparse MCS framework to facilitate cost-effective TSE, which utilizes a small number of vehicular MCS participants distributed across all regions as data sources. By utilizing spatial and temporal correlations of traffic flow, an innovative spatiotemporal deep learning model, namely, transformer graph attentional sample and aggregate (TGASA) neural network, is proposed to improve the TSE accuracy with sparse MCS data. Specifically, we design an incorporated graph neural network (GNN) to aggregate the spatial correlation by taking both node features and edge properties into account. And, the transformer neural network architecture is applied to capture the temporal correlation. Extensive simulation results based on real-world data sets demonstrate that the proposed framework can significantly address the instability incurred by the sparsification of MCS data and effectively achieve a more accurate TSE.**

***Index Terms*—Graph neural network (GNN), mobile crowdsensing (MCS), traffic state estimation (TSE), transformer, vehicular network.**

I. Introduction

TRAFFIC state estimation (TSE) is crucial for intelligent transportation systems (ITSs), as it enables various applications that rely on real-time and accurate traffic information, such as congestion monitoring, traffic management, and route planning [1], [2], [3]. Traditionally, the TSE is achieved by using inductive loop detectors and roadside cameras to record vehicle speeds or flow passing through a certain area. Suffering from limited coverage and expensive installation and maintenance costs, these road infrastructures are unable to meet the fast technology development of ITS [4], [5]. To overcome these challenges, mobile crowdsensing (MCS) has emerged as a promising sensing paradigm that leverages the data contributed or generated by ubiquitous mobile users or sensors [6], [7]. In reality, the advent of vehicular networks has enabled numerous vehicles to be connected and provide massive ITS data covering the whole road network in real-time [8], [9], [10], [11], [12]. By exploiting the collected data, including vehicle speeds and GPS coordinates, the MCS-based TSE can provide real-time and comprehensive traffic state information for the whole road network, which overcomes the limitations of previous approaches in terms of coverage and costs. For instance, the Land Transport Authority of Singapore uses GPS records from more than 12000 taxis to generate citywide traffic state information on its TrafficScan portal, which provides real-time traffic states to all Singaporeans [13].

The performance of MCS relies on a large amount of high-quality data, whereas it is commonly difficult to obtain adequate data due to the collecting overhead [14]. Collecting a tremendous amount of ITS data in real-time is a demanding task for both the collection mechanism and the communication network [15]. Instead, collecting a portion of vehicular data, namely, the sparse data, is more feasible and economical in real applications. Accurately estimating the traffic state from sparse MCS data is significantly necessary for two perspectives. First, from a passive perspective, it is not feasible to collect all the vehicle-driving information due to transmission loss or user selfishness. The data center may inevitably miss some information, as the participation of users depends on their willingness and personal constraints, such as privacy risks and available network resources [16]. Second, from an active perspective, collecting sparse data is more economical.

Manuscript received 23 October 2023; revised 19 December 2023; accepted 4 January 2024. Date of publication 22 January 2024; date of current version 25 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62271244 and Grant 62201311; in part by the Natural Science Fund for Distinguished Young Scholars of Jiangsu Province under Grant BK20220067; in part by the High-Level Innovation and Entrepreneurship Talent Introduction Program Team of Jiangsu Province under Grant JSSCTD202202; and in part by the Peng Cheng Laboratory Major Key Project under Grant PCL2023AS1-5 and Grant PCL2021A09-B2. *(Corresponding author: Haibo Zhou.)*

Jianzhe Xue, Yunting Xu, Tianqi Zhang, Qinghong Shen, and Haibo Zhou are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: jianzhexue@smail.nju.edu.cn; yuntingxu@smail.nju.edu.cn; tianqizhang@smail.nju.edu.cn; qhshen@nju.edu.cn; haibozhou@nju.edu.cn).

Wen Wu is with the Frontier Research Center, Peng Cheng Laboratory, Shenzhen 518066, Guangdong, China (e-mail: wuw02@pcl.ac.cn).

Weihua Zhuang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: wzhuang@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2024.3356554

A smaller size of data can reduce the network transmission load and ease the storage and processing tasks of the data center.

The sparse MCS for TSE collects information from a small group of participants distributed across all regions as data sources [17], [18]. However, reducing the number of MCS participants poses a challenge, as it results in unstable and less accurate TSE outcomes than the outcomes obtained from the data with more participants [19]. Nevertheless, in the MCS vehicular data, the characteristics of a real transportation system introduce information redundancy, which enables the use of sparse data for TSE. At the micro-level, the motion of each vehicle follows a common car-following model, where the speed of one vehicle is similar to that of the vehicles in its vicinity. To some extent, the speed of one vehicle can represent the speeds of a group of vehicles nearby. At the macro-level, the urban traffic conditions exhibit high spatial and temporal correlations [20]. For instance, roads adjacent to congested roads are more likely to be congested, and the appearance and dissipation of traffic jams are a time-consuming process. Therefore, it is feasible to accurately estimate the traffic state from the sparse MCS vehicular data by exploiting the spatial and temporal correlations in the traffic flow.

Capturing the spatial and temporal correlations of traffic data is a challenging task, especially for achieving an accurate estimation with sparse MCS data [21]. Temporally, the original sparse estimated result is unstable and noisy, making it hard to capture its tendency [22]. Spatially, the topology of the road network is irregularly connected in a graph structure, where each node has a different number of neighbors. Moreover, the spatial correlation capturing method needs to be transferable, as the structure of the road network changes dynamically with the construction and maintenance of roads. However, most graph neural networks (GNNs) can only generate embeddings for a single fixed graph and cannot learn to generalize across different graphs [23]. Furthermore, these approaches often simplify graph edges to binary values and ignore edge properties.

In this article, we propose a novel sparse MCS framework to facilitate low-cost TSE, and the corresponding spatio-temporal deep learning model for obtaining accurate TSE from sparse MCS data. Specifically, we randomly select a small portion of vehicular MCS participants that are evenly distributed citywide to be the sources of sparse MCS data. We observe that the sparsification of MCS data incurs instability and degrades TSE estimation accuracy. To this end, we propose a transformer GNN, named transformer-graph attentional sample and aggregate neural network (TGASA). TGASA can mitigate the influence of data sparsification and achieve stable and accurate TSE in real-time by capturing the spatial and temporal correlations of the traffic data. For the spatial correlation, we design a new variant of GNN, named graph attentional sample and aggregate neural network (GASA), based on the sampling mechanism and attention mechanism. Specifically, we use the sampling mechanism to uniformly sample a fixed-size set of neighbors as the node feature inputs. The edge feature of the graph is considered as the attention to enable the model to learn the complicated topological relationships of a road network, thereby incorporating the road structures into consideration. For the temporal correlation, the transformer is applied to learn the temporal tendency of the traffic data. Extensive simulations on the real-world data set demonstrate that the TGASA-aided sparse MCS is feasible for TSE and achieves an accurate estimation with low-cost sparse data. Our algorithm outperforms others and has the virtue of adaptability and robustness.

The main contributions of this article are summarized as follows.

1) We propose a novel sparse MCS framework to facilitate cost-effective TSE, which can achieve accurate estimation even with a small portion of MCS vehicular network data. Specifically, we reduce the number of vehicular MCS participants uniformly across all regions of the city to collect sparse MCS data.
2) We discover that the sparsification of MCS participants incurs imprecision and instability to the average vehicle speeds on road segments, where the estimation errors are similar to the Gaussian noise. We model the sparse MCS-based TSE problem as a sequential graph data denoise problem.
3) We design a novel spatio-temporal GNN TGASA to achieve an accurate estimation of traffic state from the sparse MCS data. The TGASA leverages the synergy of capturing both spatial and temporal correlation of the traffic data to achieve robust performance. Moreover, it is transferable to adopt a dynamic graph structure.

The remainder of this article is organized as follows. We first review the related works on TSE and deep learning approaches in Section II. In Section III, we formulate the estimation problem by analyzing the sparse MCS data. The model structure of TGASA is elaborated in Section IV. We present a comprehensive experiment to show the feasibility of sparse MCS and the superiority of TGASA in Section V. This work is concluded in Section VI.

II. Relate Work

A. Traffic State Estimation With MCS

TSE is a classic and fundamental problem for the vehicle transportation system. Traditional methods using loop detectors and microwave sensors are not suitable for a large-scale entire road network, due to the installation and maintenance costs. Moreover, historical and empirical data are insufficient for providing precise TSE that can satisfy the requirements of ITS applications. With the proliferation of Internet-connected vehicles, vehicular MCS data provide speed information with GPS coordinates for the entire road network and can be used to estimate the traffic state [24]. Unfortunately, these vehicular data are commonly coarse-grained due to the uneven temporal and spatial distribution of vehicles and the insufficient data collection, leading to an incomplete and inaccurate estimation. The data for some road segments can be missed. There have been numerous studies focusing on imputing these missing traffic speed data. A data recovery algorithm using an improved low-rank minimization problem with nonlinear spatial and temporal correlations is proposed to impute

missing speed data for the road segment network, under the assumption that several road segments share similar physical characteristics [25]. A compressive sensing-based algorithm is proposed to solve the missing data problem by exploiting the hidden structures within the traffic conditions of a road network, and it does not need the costly and complicated traffic models [22]. An approach is proposed to estimate travel speeds on road sections, which exploits the spatial-temporal causality among the travel speeds by a time-lagged correlation coefficient function quantifying the time consumption [26]. In addition, some deep learning methods have been developed for imputation. The graph aggregate generative adversarial network (GA-GAN) uses a generative adversarial network to generate complete TSE based on the extracted spatial and temporal features to achieve traffic estimation imputation [27]. The preceding methods mainly focus on achieving the estimations for the roads without data. As sparsification of MCS data may bring the accuracy decline to the estimation for the roads with data, further studies are needed for their practical applications.

B. Machine Learning for Traffic Data

The characteristics of traffic data are mainly represented as the temporal correlation and spatial correlation. With the rapid development of deep learning, deep neural network models can be exploited to learn and utilize spatiotemporal correlations.

The temporal feature of traffic data is hidden in a sequential structure. The recurrent neural network (RNN) is a type of neural network model for sequential data, and it is distinguished by taking information from prior input to influence the current input and output [28]. The long short-term memory (LSTM) networks and gated recurrent unit (GRU) networks are two classic variants of RNN. The LSTM is augmented by recurrent gates called forget gates and performs better than a traditional RNN on tasks involving long time lags [29]. The GRU has fewer parameters than LSTM and consists of a reset gate and an update gate [30]. Recently, a new architecture named transformer, which is only built on attention mechanisms to model sequential data, has been introduced to replace the RNN for machine translation [31]. Different from the RNN that needs data feeding recursively, the transformer is able to compute the attentions of all input elements in parallel, making the architecture computationally more efficient. To preserve the order information, the transformer applies a positional encoding mechanism to encode the positions of input elements with a series of sinusoidal functions. Transformer architecture and its variants have achieved great success in traffic data processing tasks [32].

The spatial feature of traffic data is hidden in a graph structure. A convolution neural network (CNN) specializes in tasks of modeling the spatial dependence for Euclidean space, such as regular grids and images, and is applied to traffic data processing with grid map [33]. However, the road network is in non-Euclidean space, which is not suitable for applying a CNN. To fit the graph-structured data, a series of GNN has been developed [23], [34], [35], [36]. Graph convolutional network (GCN) follows a layerwise propagation rule to learn the spatial correlation of traffic data [21], [35]. Graph attention network (GAT) leverages masked self-attentional layers to give different weights to different nodes in a neighborhood [36], [37]. graph sample and aggregate (GraphSAGE) is an inductive framework that samples and aggregates information from local neighborhoods [23], [25]. The existence of the sampling mechanism enables GraphSAGE to be transferable to previously unseen graph data.

In recent years, attention mechanisms have been widely applied to various machine learning tasks due to their flexibility and efficiency in modeling dependencies. It was first implemented as a part of the decoder for the machine translation task [38]. The aim of attention mechanisms is to enable the model to focus on the most relevant parts according to the features regardless of their distance in the input or output sequences. Graph multiattention network uses an attention layer to convert the encoded traffic features to generate the sequence representations of future time steps as the input of the decoder [39]. A spatio-temporal GNN framework adopts a learnable positional attention mechanism to aggregate the information from neighboring roads [40]. Gated Attention Networks use a small convolutional network to compute a soft gate at each attention head, which depends on the speed information, to control its importance [41].

Although the existing approaches can properly capture the information of traffic data, new solutions are needed to utilize the geometry properties of the road network, such as the angle between two roads and the length of a road. In other words, we should go beyond node features and account for edge features. In this work, we comprehensively consider both node and edge features of the traffic graph, in which the edge features are learned as attention.

III. Framework and Problem Analysis

This section consists of four parts. First, we present the sparse MCS framework for TSE. Second, we introduce the road-level macroscopic traffic graph of Beijing and the corresponding road matching algorithm. Third, we analyze the estimation errors of sparse MCS data, and model the errors as the Gaussian noise. Fourth, we formulate the traffic speed recovery problem based on sparse MCS data. For convenience, Table I summarizes the major notations of this article.

A. Sparse MCS Framework

The framework of sparse MCS for cost-effective TSE is shown in Fig. 1. First, the data collector collects a sparse MCS data set of vehicle-driving information through vehicular networks. Different from removing data of some regions of road segments, the data sparsification in our work refers to randomly selecting a small portion of vehicular MCS participants that are evenly distributed citywide as data sources, with the same recruitment probability for each individual vehicle. The traffic state of a road segment is represented by the average speed of the vehicles within a past period. Once the data collection is completed, these data are organized by a road segment matching process. Then, an original estimation is

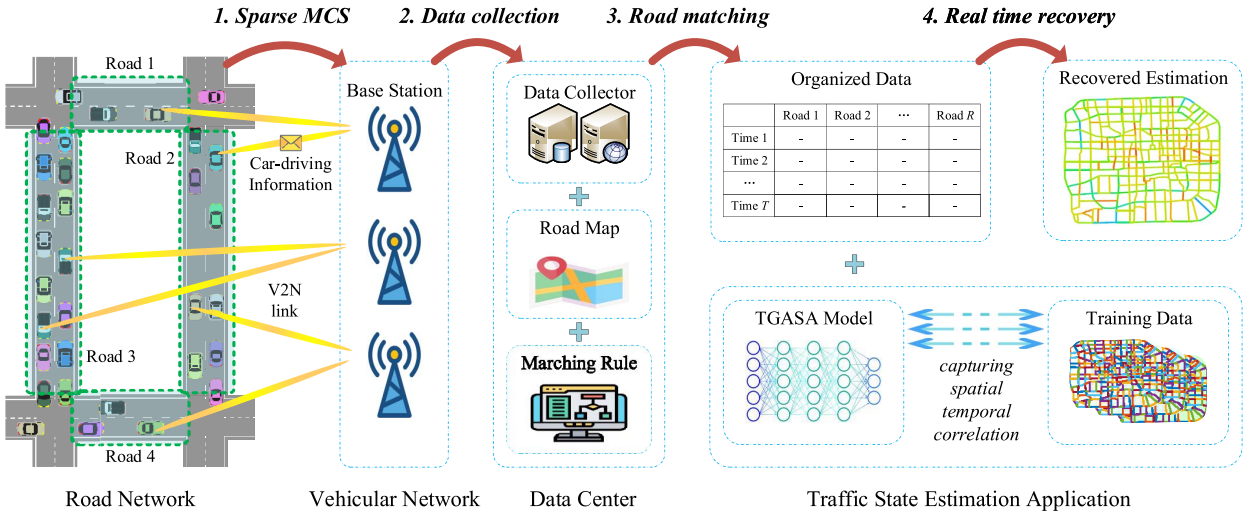


Fig. 1. Illustration of MCS for TSE.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS

$\mathcal{G}$	Road-oriented traffic graph
$e_{i,j}$	Edge feature of node i and node j
y_t^m	Estimated speed of road m at time slot t with all data
$\boldsymbol{Y}_t$	Ideal estimated speeds of road network at time slot t
x_t^m	Estimated speed of road m at time slot t with sparse data
$\boldsymbol{X}_t$	Original estimated speeds of road network at time slot t
G_t^{n}	Aggregated node features for a target node at time slot t
G^{e}	Aggregated edge features for a target node at time slot t
G_t	Target node representation at time slot t
$\mathcal{P}$	Positional encoding matrix
G^{p}	Sequential target node representations
Q	Query in self attention computation
K	Key in self attention computation
V	Value in self attention computation
h_i	One of the heads in multi-head attention block
Z	Outputs of the transformer
w, b	Neural network parameters
$\mathcal{F}(\cdot)$	Traffic state estimation mapping function
$\mathcal{A}(\cdot)$	Self attention function
$\mathcal{M}(\cdot)$	Multi-head attention function
$\mathcal{N}(\cdot)$	Layer normalization operation function

obtained by calculating the average speed of the vehicles for each road segment. Finally, we apply a recovery algorithm to achieve an accurate estimation of the traffic state based on the sparse MCS data.

B. Road-Oriented Traffic Graph Network

In this study, the city map is transformed to a road-oriented traffic graph network based on the graph theory. With a road matching rule, the vehicles with known GPS coordinates can be matched to road segments according to the geometric restrictions. Then, the average speed of each road segment can be obtained to achieve the road segment level TSE.

Typically, a graph consists of nodes and edges. To make the GNN model focus on the traffic information of road segments, we construct the road-oriented traffic graph, where nodes correspond to road segments and edges indicate their topological connections. The reason for considering road segments as nodes is that GNN is generally more powerful and efficient in capturing the node features, especially for the GraphSAGE which is designed for aggregating the features of neighboring nodes. Fig. 2 shows the construction process of the road-oriented traffic graph in three steps. First, based on the data from the Open Street Map, 500 main road segments within the Fourth Ring Road of Beijing are selected to construct a macroscopic road network as shown in Fig. 2(b). In Fig. 2(c), a small portion of the real physical road network is taken as a typical example. It naturally forms a graph, where nodes represent crosses and edges represent road segments. However, the original graph structure is not suitable for the road level speed estimation problem since the road segment features are more relevant, and it varies rapidly over time [42]. Therefore, we exchange the representation of nodes and edges to construct the road-oriented traffic graph as shown in Fig. 2(d).

Mathematically, we use undirected graph $\mathcal{G} = (\mathscr{R}, \mathscr{E}, \mathbf{A})$ to describe the road-oriented traffic graph, where $\mathscr{R} = \{r_1, r_2, \ldots, r_M\}$ is a set of nodes representing M road segments, $\mathscr{E} = \{e_{i,\ j}\}$ is a set of edges representing the topological connection relationship between two neighboring nodes, $\mathbf{A}$ is the adjacency matrix of graph $\mathcal{G}$. Specifically, each edge is represented as $e_{ij} = [\theta_{ij}, d_{ij}, l_j, p_j]$, where θ_{ij} is the angle between road segments i and j, d_{ij} is the distance between the middle points of two road segments, l_j is the length of road segment j, p_j is the property of road segment j. Note that $p_j = 1$ if road segment j is a main ring road, otherwise $p_j = 0$. Note that the overhead of building the road-oriented traffic graph network is small.

The cars are matched to the road segments with their GPS coordinates and the road matching rule in the following steps. First, the vehicle-driving data is collected through the vehicular network. To estimate the traffic state at a given time slot t, we use all the information collected over the past period of length T at time t and denote them as information set I_t for time slot t. To enhance estimation accuracy, we eliminate information from special vehicles, such as emergency or parked vehicles. Every item of vehicle-driving information in I_t is individually matched to the nearest road segment in real-time according to

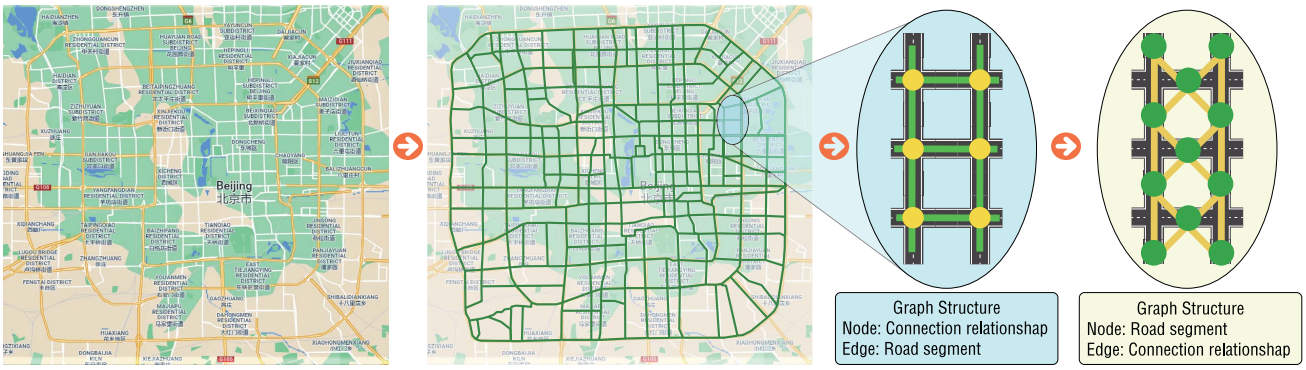


Fig. 2. Process of building the traffic graph of Beijing. (a) Map of Beijing. (b) Traffic network of Beijing. (c) Natural formed graph. (d) Road-oriented graph.

the following geometric restrictions:

$$d_c < d_h \tag{1}$$

$$\theta_c < \theta_h \tag{2}$$

where d_c is the distance between the vehicle's GPS coordinates and the road segment, θ_c is the angle between the vehicle's velocity direction and the road segment, d_h and θ_h are thresholds for distance and angle, respectively. Then, based on the matching results of I_t, we build a new set, $R_t = \{R_t^1, R_t^2, \ldots, R_t^m, \ldots, R_t^M\}$, where $R_t^m = \{v_1^m, v_2^m, \ldots, v_c^m\}$ is a set consisting of all speeds of vehicles that have matched to the road segment m in the time slot t and its size is denoted by C. Then, the traffic state is estimated with R_t at the data center. The ideal estimation of traffic state for each road segment is calculated as

$$y_t^m = \frac{\sum_{v^m \in R_t^m} v^m}{C} \tag{3}$$

where y_t^m is the average speed of vehicles on road segment m at time slot t obtained by dense data. For the entire road network, we denote its traffic state at time slot t as $\boldsymbol{Y}_t = [y_t^1, y_t^2, \ldots, y_t^m, \ldots, y_t^M]$. Here, $\boldsymbol{Y}_t$ can be regarded as a set of node features of graph $\mathcal{G}$, indicating the traffic state of the road network at time slot t.

C. Estimation Accuracy Analysis

The traffic state can also be obtained by the sparse data with a similar process. However, we have discovered that, although the average speed of sparse data is similar to the average speed of dense data, it is noisy and unstable. By theoretical analysis, we found that the noise is Gaussian in every time slot.

The process of obtaining average vehicle speeds on road segments from sparse data is as follows. The sparse data of vehicle-driving information collected in time slot t is denoted as I_t', and it can be regarded as a subset constructed by simple random sampling from dense data I_t. We give the same probability of occurrence for each item of vehicle-driving information for two reasons. On one hand, the passive data missing caused by network instability is random to all vehicles. On the other hand, reducing the collection frequency of vehicle-driving information can be seen as an expected result of random collection with a smaller probability in the time domain. Then, with a similar matching process, we can obtain the set, $S_t = \{S_t^1, S_t^2, \ldots, S_t^m, \ldots, S_t^M\}$, where $S_t^m = \{v_1^m, v_2^m, \ldots, v_{c'}^m\}$ is a set consisting of all vehicle speeds in I_t' that are matched to road segment m and its size is denoted by C'. That is, the sparse data is a subset of the dense data, i.e., $I_t' \subseteq I_t$, $S_t^m \subseteq R_t^m$. Similarly, in the sparse data case, the original estimation of traffic state at time slot t is denoted as $\boldsymbol{X}_t = [x_t^1, x_t^2, \ldots, x_t^m, \ldots, x_t^M]$ and

$$x_t^m = \frac{\sum_{v^m \in S_t^m} v^m}{C'} \tag{4}$$

where x_t^m denotes the average speed on road segment m at time slot t obtained by sparse data.

The theoretical analysis shows that x_t^m equals y_t^m plus Gaussian noise $w(t, m)$. For a population with mean μ, if we take sufficiently large random samples from the population, according to the central limited theory (CLT), the mean of all sampled variables is a random variable and is approximately normally distributed with mean μ [43]. Applying the CLT to the sparse MCS data traffic estimation problem, the ideal average speed from dense data is the population mean and the original average speed from sparse data is the sample mean. Thus, x_t^m can be considered as a random variable that is approximately normally distributed with mean y_t^m and their relationship is given by

$$E\left(x_t^m\right) = y_t^m \tag{5}$$

$$x_t^m = y_t^m + w(t, m) \tag{6}$$

where $w(t, m) \sim N(0, \sigma_{t,m}^2)$. The standard deviation, $\sigma_{t,m}$, changes slowly with time due to the temporal coherence of traffic data and is highly related to the population size of information matched to road segment m. To validate the above relationship, we present the probability distribution functions of the difference between y_t^m and x_t^m at 4 different sparsities, including 5%, 10%, 20%, and 40% in Fig. 3. They all are bell-shaped curves, indicating that the differences between y_t^m and x_t^m follow the Gaussian distribution.

D. Traffic State Estimation Problem

Based on the preceding definitions and analyses, the real-time TSE with sparse crowdsensing data becomes a sequential

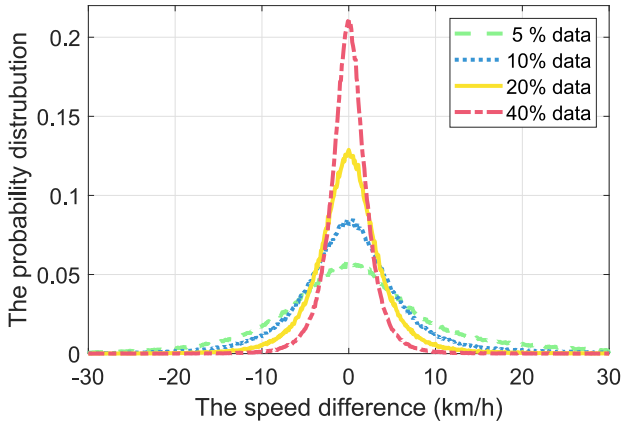


Fig. 3. Error distribution of sparse data.

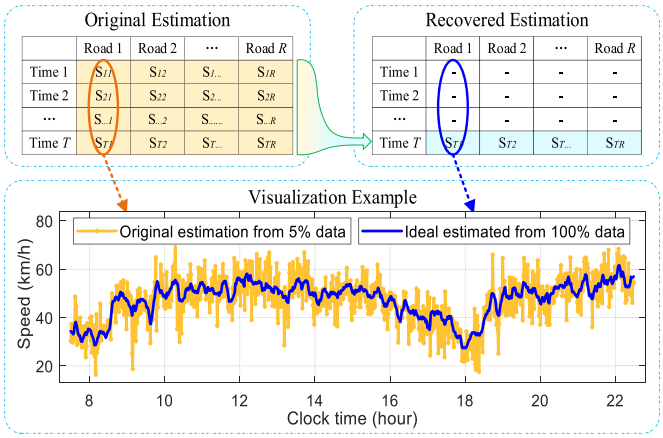


Fig. 4. Recovery problem for TSE.

graph structure data recovery problem. The original average speeds calculated from spare data form a traffic state graph, which cannot be used as the final estimation due to the existence of inaccuracy and instability as compared with the ideal estimation.

Fig. 4 shows the recovery problem for TSE from sparse MCS data with a visualization example of the estimated speeds on one road in a day. It is expected to recover an accurate estimation based on original average speeds in real-time. Specifically, the inputs of the recovery algorithm include the historical and current original estimation of traffic state graph from sparse data, $[\boldsymbol{X}_{t-(T-1)}, \ldots, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$. The goal is to reconstruct a high-accuracy current traffic state graph, $\boldsymbol{Y}_t$, similar to the ideal estimation from dense data. To obtain an accurate estimation, it is essential to leverage the spatio-temporal characteristics of traffic data over a relatively short period in the past to reduce the Gaussian noise. Thus, the recovery problem for TSE is to learn the mapping function, $\mathcal{F}(\cdot)$, defined as

$$\boldsymbol{Y}_t = \mathcal{F}\left(\left[\boldsymbol{X}_{t-(T-1)}, \ldots, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t\right]; \mathcal{G}\right) \tag{7}$$

from original average speeds of sparse data and road graph topology to an accurate estimation of traffic state.

IV. Methodology

In this section, we introduce the TGASA model used to obtain an accuracy estimation from sparse MCS data by capturing correlations of traffic flow. First, we elaborate its network structures for capturing spatial correlation and temporal correlations by GASA and transformer, respectively. Then, we present an overview of the TGASA neural network is given.

A. Spatial Correlation Modeling

The spatial correlation of traffic data is hidden in a graph structure. We design a novel variant of GNN, named GASA, to capture the spatial correlation by considering both node features and edge properties of 1-order neighboring nodes. Note that the GASA is applied to each node individually and the model parameters for all the nodes are shared as the same.

It is crucial to aggregate the information of other roads into estimation since the traffic condition of a road can be highly influenced by other roads in its neighborhood. However, in a road network system, the traffic influence is limited with spatial distance, i.e., the traffic condition of a road has little impact on those of other roads far away. Therefore, accounting for the existence of influence limitation, we restrict the traffic features used in the GNN aggregation to only 1-order neighboring road segments.

Besides, the traffic flow characteristic is highly related to the road topological structure. For a road segment, we use the attention approach to characterize the differences in traffic condition impacts from different neighboring road segments. For example, adjacent roads in the same direction usually have a bigger impact than adjacent roads that are perpendicular. Thus, the edge information representing road topological structure is integrated into the GASA model simultaneously with node information.

Furthermore, the salient feature of the graph structure is that each node has a varied number of 1-order neighboring nodes. However, the structure of the neural network is generally fixed, resulting in a fixed length of inputs. To adapt the graph structure and empower the model's transferable ability, we utilize a sampling approach with a selection rule that selects a fixed number of 1-order neighboring nodes as inputs of the GASA model for each node. In this way, the conflict between the changing number of 1-order neighboring nodes and the fixed length of inputs is resolved. Specifically, for a node in the road-oriented traffic graph, its 1-order neighboring nodes are sorted according to the angle between two roads, and a fixed number of neighboring nodes with the smallest angle are selected. The reason for choosing the angle as a measure is that branches often have a smaller traffic influence. If there are no sufficient neighboring nodes, some of them can be selected multiple times randomly.

The input of GASA consists of node information and edge information of the selected nodes. The node information input vector is denoted by $N_t = [x_t^A, x_t^B, x_t^C, x_t^D]$, which is concatenated by node features of the node itself and the sampled nodes. These sampled nodes are sorted by the angle between two roads. Then, the node information input vector

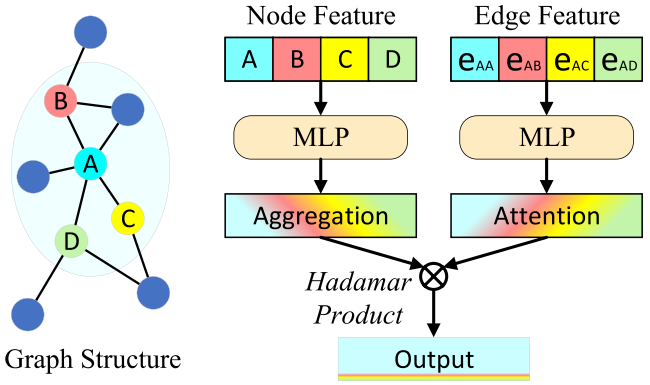


Fig. 5. GASA model.

passes a multi layer perceptron (MLP) to get an aggregation vector, G_t^n, given by

$$G_t^n = (\tanh(N_t w_1 + b_1))w_2 + b_2 \tag{8}$$

where w_1, b_1, w_2, b_2 are neural network parameters and $\tanh(\cdot)$ is the hyperbolic function defined as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{9}$$

The edge information input vector is denoted as $E = [e_{AA}, e_{AB}, e_{AC}, e_{AD}]$, which is concatenated by edge features of the corresponding edges in the same order as nodes. The attention vector, G^e, is obtained with an MLP, given by

$$G^e = (\tanh(Ew_3 + b_3))w_4 + b_4 \tag{10}$$

where w_3, b_3, w_4, b_4 are neural network parameters. Note that G_t^n and G^e have the same size. Finally, with node feature aggregation vector and edge feature attention vector, the output of the GASA model, G_t, can be obtained by

$$G_t = G_t^n * G^e \tag{11}$$

where $*$ denotes the Hadamard product.

The GASA model processes the graph node by node. Fig. 5 shows the structure of GASA with an example of node A. We first take the features of only node A and its 3 neighboring (B, C, D) nodes into consideration, and then the node features and edge features are aggregated with the GASA model. Since all the nodes share the same network structure and there is no dependence between the outputs of different nodes, the calculations of different nodes can be done parallelly. The output vectors of all nodes are organized in a graph with the same graph structure as the road-oriented graph and the information of 1-order neighboring nodes is aggregated in the new graph.

B. Temporal Correlation Modeling

The temporal correlation of traffic data is hidden in a sequential structure. The graph features of the past $(T-1)$ time slots and the current time slot are used to build the sequential data to estimate the traffic state of the current time slot t. We employ the transformer encoder with positional encoding to learn the sequential dependence of the traffic data.

The positional encoding is essential before feeding the data into the transformer encoder structure since it has no recurrent operation but only feed forward structures, which lack the ability to recognize the order of time-sequential data. To extract temporal correlation from the input sequential data, first, we need to encode the relative position of the different times. Following the positioning encoding widely used in the transformer, we use the sine and cosine functions of different frequencies as the encoding function to provide positional information on different timestamps. The positional encoding matrix, $\mathscr{P}$, is a fixed $\mathbb{R}^{T\times d_m}$ matrix that encodes the absolute positional information, given by

$$\mathscr{P}(p,i) = \begin{cases} \sin\left(\frac{p}{50^{\frac{i}{d_m}}}\right), & \text{if } i \text{ is even} \\ \cos\left(\frac{p}{50^{\frac{i-1}{d_m}}}\right), & \text{if } i \text{ is odd} \end{cases} \tag{12}$$

where p is the position, d_m is the dimension length and i is the dimension index. Adding the positional encoding matrix to the sequential graph features, the input of the transformer encoder layer, G^p, can be obtained as

$$G^p = \left[G_{t-(T-1)}, \ldots, G_{t-1}, G_t\right]^T + \mathscr{P} \tag{13}$$

where $[\,\cdot\,]^T$ is the transpose operation, G_t is the output of the GASA with graph feature at time slot t.

After the positional encoding, the transformer encoder layer is used to capture its temporal correlation. The multihead attention block based on the self-attention mechanism is the key to the transformer. The attention mechanism is powerful in measuring the importance of each part of the input vector and guiding the neural network to focus on the important part when generating the output vector. It is an effective method for "nonlocal" feature extraction. Such a mechanism is suitable for sequential data processing since it can find the correlation between each combination of any two different time slots. The inputs of self attention block consists of three $\mathbb{R}^{T\times d_k}$ matrices, the query Q, the key K, the value V as follows:

$$\begin{cases} Q = G'W_q \\ K = G'W_k \\ V = G'W_v \end{cases} \tag{14}$$

where W_q, W_k, W_v are neural network parameters. Then, the self attention function, $\mathcal{A}(\cdot)$, is computed as

$$\mathcal{A}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt[2]{d_k}}\right)V \tag{15}$$

where $softmax(\cdot)$ is an activation function that turns a vector into another same-size vector with the values of vector elements summing to 1 and is defined as

$$\text{softmax}(a_i) = \frac{e^{a_i}}{\sum_j e^{a_j}}. \tag{16}$$

Instead of only using a single attention function, it is beneficial to use the multiple head attention schema since it can capture the correlations from multiple aspects. Specifically, the queries, keys and values are linearly projected h times with different, learned linear projections. These heads can jointly aggregate information of different representation subspaces

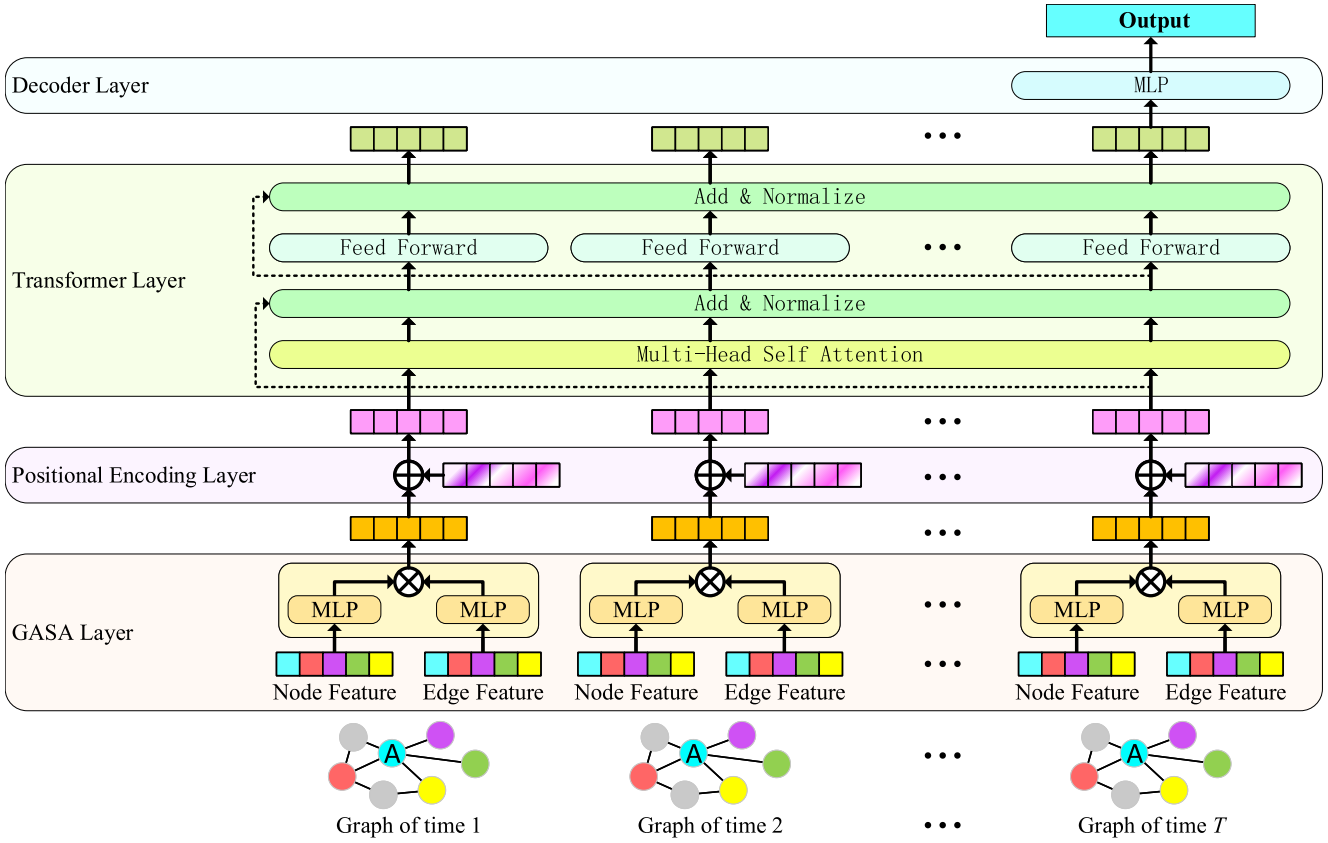


Fig. 6. TGASA model.

from different aspects thus enhancing the learning capability of the model. The multihead attention block function, $\mathcal{M}(\cdot)$, is computed as

$$\mathcal{M}(Q, K, V) = [h_1, h_2, \ldots, h_i]W_o \tag{17}$$

where W_o is a neural network parameter. In (17), h_i is the output of one head and computed as

$$h_i = \mathcal{A}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{18}$$

where W_i^Q, W_i^K and W_i^V are head-specific weights matrices of Q, K and V. The output of the multihead attention block, Z^m, is obtained by these heads passing through a linear layer. Until now, each vector in the matrix Z^m has aggregated the graph information at other times.

Then, Z^m passes through a three layer hybrid network, consisting of two *Add&Normalize* layers with a feed-forward layer in between. The layer normalization operation function, $\mathcal{N}(\cdot)$, is a general operation in a neural network that normalizes input across the features [31]. The *Add&Normalize* layer has a residual connection and is followed by a layer normalization. The feed-forward layer is a fully connected network with a rectified linear unit (ReLU) activation function. Respectively, the computation of the three layers are

$$Z^n = \mathcal{N}\left(Z^m + G^p\right) \tag{19}$$

$$Z^f = \max\left(0, Z^n w_5 + b_5\right)w_6 + b_6 \tag{20}$$

$$Z = \mathcal{N}\left(Z^n + Z^f\right) \tag{21}$$

where w_5, b_5, w_6, b_6 are neural network parameters and $Z \in \mathbb{R}^{T\times d_m}$ is the output of the transformer layer. Note that the output of the transformer layer has the same structure as its input

$$Z = \left[Z_{t-(T-1)}, \ldots, Z_{t-1}, Z_t\right]^T \tag{22}$$

where Z_t is the hidden representation at time t. The output of the transformer is a sequence of hidden representations that contain the spatial and temporal correlations of the traffic data.

Finally, a decoder layer is needed to decode the recovery traffic state from the hidden representations with an MLP. The input of the decoder is Z_t, the hidden representation from the transformer at time t. The output of the decoder is a recovered speed on a road segment for time slot t.

C. TGASA Model

The architecture of the proposed TGASA model is illustrated in Fig. 6. Our TGASA model comprises four parts: 1) multiple GASA blocks for capturing spatial correlation; 2) a positional encoding layer for adding relative position information; 3) a transformer layer for capturing temporal correlation; and 4) a decoder layer for decoding an output speed from hidden states.

Algorithm 1 describes the forward propagation of our TGASA model, under the assumption that the model has already been trained and that the parameters are given. The TGASA receives a sequence of traffic speed graphs as input. First, we use the sampling mechanism to uniformly sample a

Algorithm 1 TGASA Algorithm

Input: Road network graph $(\mathcal{G} = \mathscr{R}, \mathscr{E}, \mathbf{A})$, target node n, sequential graph node features $[\boldsymbol{X}_{t-(T-1)}, \ldots, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$
Output: Estimated speed of node n at time t

1: Node feature sampling for node n:$N_t \leftarrow \boldsymbol{X}_t$, $\forall t \in [t-(T-1), t]$
2: Aggregate node features: $G_t^n \leftarrow AGGREGATE(N_t)$
3: Edge feature sampling for node n:$E \leftarrow \mathscr{E}$
4: Aggregate edge features: $G^e \leftarrow AGGREGATE(E)$
5: Aggregate node and edge features of node n: $G_t \leftarrow G_t^e * G^n$
6: Make up sequential data of node n: $G \leftarrow CONCAT(G_{t-(T-1)}, \ldots, G_{t-1}, G_t)$
7: Positional encoding: $G^p \leftarrow G + \mathscr{P}$
8: Transformer with multi-head self-attention: $Z \leftarrow G$
9: Decode the speed for node n: $y_t^n \leftarrow Z$

fixed-size set of neighbors, and their node features and edge features are formulated as the input of the GASA layer. By multiple GASA blocks with the same neural network settings, these graph features containing the spatial correlation are aggregated in parallel to compose sequential graph embedding vectors. Then, the positional encoding layer adds a piece of information to each graph embedding vector about its sequential position. After that, the transformer is applied to capture the temporal correlation in the sequential graph embedding vectors. Finally, the decoder layer gives the recovered speed with the last output vector of the transformer layer. It is worth noting that the TGASA model returns the recovered speed on each road individually, but with the same sharing neural network parameters. The recovered speeds on all the road segments are combined to form $\boldsymbol{Y}_t$, which is the final TSE for time slot t.

The TGASA model is trained to find the optimal network parameters that minimize the loss function as the mean square error (MSE). The sparse data is obtained by randomly sampling a given percentage of data from the complete vehicular data set. Then, by road matching and average vehicle speed calculation, we obtain the ideal speed and the original speed as a part of training data. The inputs of TGASA are the past and the current original speeds with the road network. The expected output of TGASA is the ideal current speed. For each training batch, the training loss and the gradient are calculated to update the model parameters. Finally, the TGASA model is ready for implementation until the training process convergences.

V. Experiments

In this section, we use the real data to evaluate the feasibility of sparse MCS for TSE and the effectiveness of TGASA for estimation recovery. We first discuss the experimental settings for evaluation. Then, the baseline methods and the evaluation metrics are introduced. Finally, the experimental results are presented in detail.

A. Experimental Settings

The experiment is carried out based on the real MCS vehicular data of Beijing. The experimental settings are given as follows.

1) Road Network: We study the TSE for the area within the Fourth Ring Road of Beijing. As illustrated in Fig. 2, our road network includes 500 main road segments. These road segments are divided into two levels adding to whether it is a ring road. The geometric restrictions for the road matching are that the maximum distance is 80 m and the maximum angle is 20 degrees.

2) Traffic Data: We use a set of real MCS vehicular driving information in Beijing as the data for evaluation. The data set includes six days in November 2012: Nov1, Nov5, Nov7, Nov9, Nov10, and Nov11. They are Thursday, Monday, Wednesday, Friday, Saturday, and Sunday, respectively. The traffic state information during the daytime is more important than that in the night since people usually travel during the day and traffic congestion is less likely to occur at night. Thus, we focus on the TSE of the daytime, the time period from 7:30 to 22:30. The total amounts of vehicle-driving information items are 7808510, 6778639, 7855002, 7707503, 5918537, and 5983471, respectively, for each day within the period from 7:00 to 22:30. Note that the additional half-hour data is needed for estimating the traffic state at 7:30. For the road matching rule, d_h is 0.0008 degrees of longitude or latitude (around 80 m), θ_h is 20 degrees. We use 1 min as the time step size for TSE, which means that we estimate the city-wide traffic state 900 times each day. The average window length of speed estimation is 10 min. The sparse data is obtained by randomly sampling the vehicle-driving information from the complete data set. The sparsity of sparse data means the percentage of the sampled vehicle-driving information from the complete data set. To have a full evaluation, we have tested the performance of TGASA at 6 different sparsities, including 5%, 10%, 20%, 30%, 40%, and 50%.

3) TGASA Hyperparameters: The GASA block aggregates the information of its 4 neighboring roads. The structures of the two MLPs are the same, where the hidden units of the two layers are 5 and 64, respectively. The dimension of its output is 32. The dimensions of the positional encoder outputs are 32. For the transformer block, we stack two layers of the transformer encoders. The input window of the transformer is 30 min. Each transformer layer has 8 heads and the dimension of the FeedForward layer is 64. Both the input and output dimensions of the transformer block are 32. For the decoder block, the hidden units of two layers are 32 and its output is a number.

4) Training and Testing: We use the data of Nov1 as the train set and the data of the other 5 days as the test set. For all the cases, the train and test data are both of the same sparsity. To show that the TGASA has high adaptability and robustness, the test set has covered a whole week, including both weekdays and weekends. The TGASA is trained less than 20 epochs with the initial learning rate at 0.005 and decays 25% after each epoch. The hardware for experiments is Intel Xeon E5-2698 as CPU.

B. Baselines and Evaluation Metrics

To evaluate the performance of the TGASA, we choose some classic time-series regression models and state-of-the-art traffic flow prediction spatiotemporal networks as follows.

1) *Original:* The average vehicle speed of each road segment from sparse data.

TABLE II
ESTIMATION RESULTS OF THE TGASA AND OTHER BASELINE METHODS

Sparsity	Metric	Model							
		Original	GraphSAGE	DNN	LSTM	DCRNN	TGCN	STGNCDE	TGASA
5%	RMSE	11.092	6.967	5.377	3.782	5.199	4.942	4.623	**3.643**
	IPV	*	37.080%	51.689%	65.848%	53.074%	55.465%	58.014%	**67.097%**
	MAE	7.642	5.123	3.627	2.768	3.567	3.691	3.276	**2.668**
	R^2	0.238	0.700	0.820	0.912	0.832	0.849	0.867	**0.918**
10%	RMSE	7.387	5.460	3.707	3.124	3.699	4.225	3.762	**3.037**
	IPV	*	25.761%	49.895%	57.626%	49.833%	42.734%	48.915%	**58.796%**
	MAE	4.970	3.964	2.509	2.251	2.529	3.158	2.642	**2.199**
	R^2	0.662	0.816	0.915	0.940	0.915	0.890	0.913	**0.943**
20%	RMSE	4.695	4.022	2.671	2.533	2.687	3.645	2.839	**2.449**
	IPV	*	14.082%	43.134%	45.992%	42.672%	22.206%	39.386%	**47.757%**
	MAE	3.163	2.865	1.821	1.773	1.846	2.715	1.998	**1.743**
	R^2	0.863	0.900	0.956	0.960	0.955	0.918	0.950	**0.963**
30%	RMSE	3.525	3.213	2.204	2.157	2.230	3.406	2.459	**2.069**
	IPV	*	8.768%	37.487%	38.768%	36.554%	3.149%	30.006%	**41.235%**
	MAE	2.377	2.263	1.501	1.497	1.536	2.567	1.731	**1.463**
	R^2	0.923	0.936	0.970	0.971	0.969	0.928	0.963	**0.973**
40%	RMSE	2.799	2.633	1.886	1.865	1.936	3.229	2.140	**1.793**
	IPV	*	5.829%	32.591%	33.350%	30.749%	-15.665%	23.417%	**35.864%**
	MAE	1.886	1.842	1.279	1.284	1.333	2.421	1.494	**1.257**
	R^2	0.951	0.957	0.978	0.978	0.977	0.936	0.972	**0.980**
50%	RMSE	2.289	2.201	1.652	1.641	1.734	3.148	1.923	**1.595**
	IPV	*	3.784%	27.834%	28.282%	24.162%	-37.937%	15.874%	**30.223%**
	MAE	1.533	1.525	1.116	1.115	1.190	2.374	1.339	**1.111**
	R^2	0.967	0.970	0.983	0.983	0.981	0.939	0.977	**0.984**

2) *GraphSAGE:* A GraphSAGE neural network where 4 neighboring roads with the smallest angles are sampled and aggregated with a three-layer linear neural network [23].
3) *DNN:* A three-layer linear deep neural network to capture the temporal correlation.
4) *LSTM:* A LSTM network, a special RNN model [29].
5) *DCRNN:* A spatiotemporal GNN using bidirectional random walks and encoder–decoder architecture with scheduled sampling [44].
6) *TGCN:* A spatiotemporal GNN that is combined with GCN and GRU [21].
7) *STGNCDE:* A spatiotemporal GNN that is based on neural controlled differential equations [45].

Specifically, the Original is the input for other machine learning methods. GraphSAGE recovers the estimation by capturing the spatial correlation, and its inputs are the current Original speeds of self and neighboring speeds. DNN and LSTM only capture the temporal correlation, and their inputs are historical and current Original speeds of an individual road segment. DCRNN, TGCN, and STGNCDE can capture both spatial and temporal correlation, and their inputs are graph structure and overall historical and current Original speeds.

Four metrics are used to quantitatively evaluate the real-time TSE performance of the TGASA and other baseline methods. Let y_t^m and $\hat{y}_t^m$ represent the ideal traffic speed and the estimated traffic speed of the road m at time t, M represents the number of road segments and $\bar{Y}$ represents the average ideal traffic speed in a day.

1) Root Mean Squared Error:

$$\text{RMSE} = \sqrt{\frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} |y_t^m - \hat{y}_t^m|^2} \tag{23}$$

which is the rooted average squared difference between the recovered values and the ground truth. Its minimum value is 0, the smaller the better.

2) Improved Percentage:

$$\text{IPV} = \left(1 - \frac{\sqrt{\frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} |y_t^m - \hat{y}_t^m|^2}}{\sqrt{\frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} |y_t^m - x_t^m|^2}}\right) \times 100\% \tag{24}$$

which is the reduced percentage of root mean squared error (RMSE) between the RMSE of the Original data and the RMSE of the recovered data. Its maximum value is 100%, the larger the better.

3) Mean Absolute Error:

$$\text{MAE} = \frac{1}{T \cdot M} \sum_{t=1}^{T} \sum_{m=1}^{M} |y_t^m - \hat{y}_t^m| \tag{25}$$

which is the average of the absolute errors. Its minimum value is 0, the smaller the better.

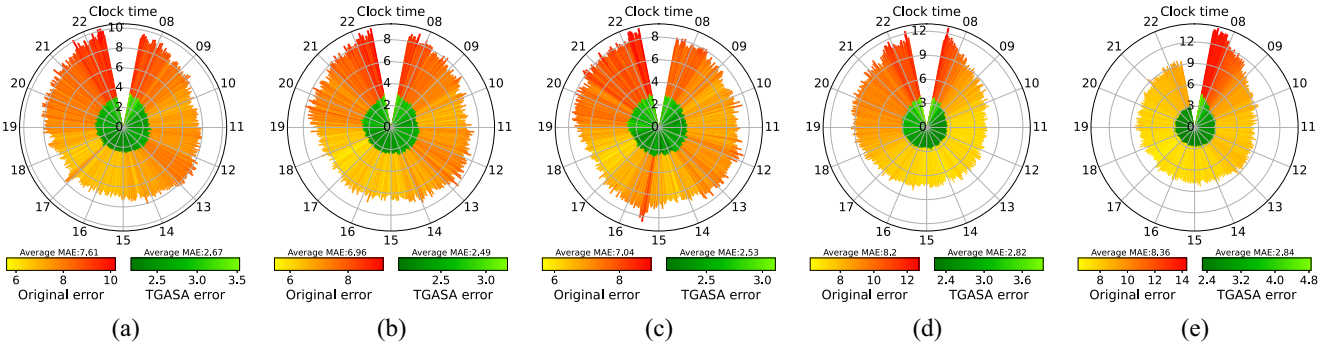
4) Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{t=1}^{T} \sum_{m=1}^{M} \left(y_t^m - \hat{y}_t^m\right)^2}{\sum_{t=1}^{T} \sum_{m=1}^{M} \left(y_t^m - \bar{Y}\right)^2} \tag{26}$$

which is a statistical measurement to examine the proportion of variation in one variable that can be explained by the variation in another variable. Its maximum value is 1, and the larger the better.

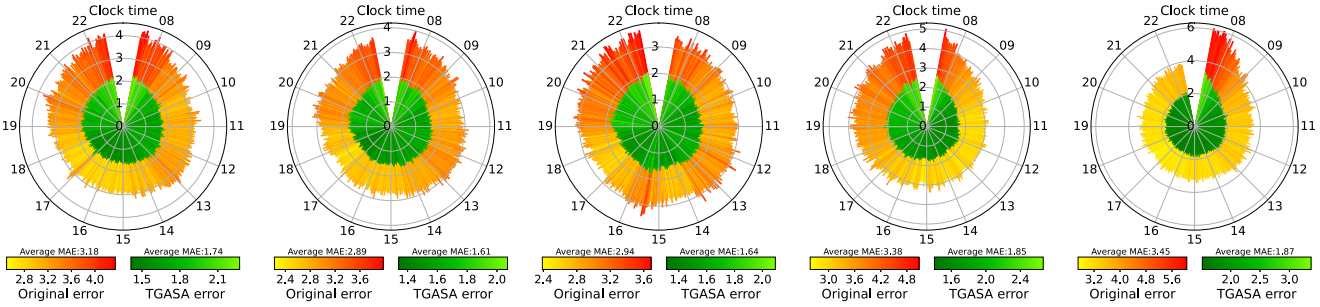
C. Performance Evaluation

To illustrate the effectiveness of the sparse MCS and the TGASA model, a comparative experiment is conducted between the Original estimation and the recovered estimation.



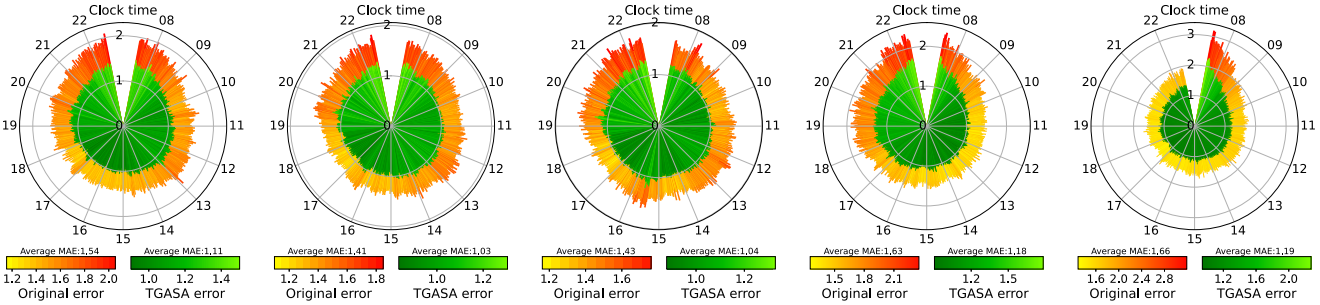
(a) (b) (c) (d) (e)

Fig. 7. MAE (km/h) comparison between the TGASA and the original at 5% sparsity. (a) Monday. (b) Wednesday. (c) Friday. (d) Saturday. (e) Sunday.



(a) (b) (c) (d) (e)

Fig. 8. MAE (km/h) comparison between the TGASA and the original at 20% sparsity. (a) Monday. (b) Wednesday. (c) Friday. (d) Saturday. (e) Sunday.



(a) (b) (c) (d) (e)

Fig. 9. MAE (km/h) comparison between the TGASA and the original at 50% sparsity. (a) Monday. (b) Wednesday. (c) Friday. (d) Saturday. (e) Sunday.

1) Overall Performance: Table II shows the estimation performance comparison of different methods at different sparsities, ranging from 5% to 50%. The sparsity is the same in each row and the method is the same in each column. We measure the five different metrics for all the cases to comprehensively show their performance. These metrics are the average of five days, including Monday, Wednesday, Friday, Saturday, and Sunday. From the experimental results, we can observe that the following.

The sparse MCS is feasible for TSE and it is cost-effective. For the Original estimation, when the sparsity decreases from 50% to 5%, the R^2 drops from 0.967 to 0.238, while the RMSE increases from 2.289 to 11.092 km/h. The performance degradation of the Original caused by the sparse data is unacceptable for the real application. To this end, the estimation performance needs to be improved by utilizing temporal and spatial correlations hidden in traffic data. The most persuasive result is the case of TGASA at 5% sparsity, where we have achieved 0.918 for R^2 and 3.643 km/h for RMSE. The RMSE has been mitigated 67.097%. In the real application, errors around 3 km/h are acceptable. Therefore, it is feasible and cost-effective to apply the sparse MCS to the TSE system, where the system can realize an acceptable estimation only using a small amount of data, such as one-twentieth.

The TGASA is superior to other methods. From Table II, we can see that the TGASA outperforms other machine learning methods and always has the smallest RMSE and mean absolute error (MAE). The GraphSAGE performs lower than others in most cases. The performance of DNN is significantly enhanced as compared with GraphSAGE but is slightly worse than LSTM. It is the same as expected that

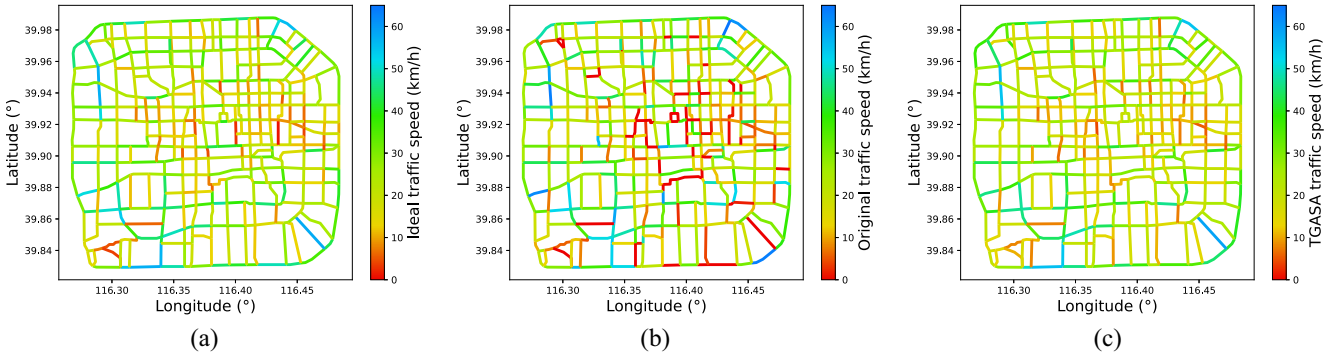


Fig. 10. Estimation of traffic speeds on Monday 6:30 P.M. at 5% sparsity. (a) Ideal estimation. (b) Original estimation. (c) TGASA estimation.

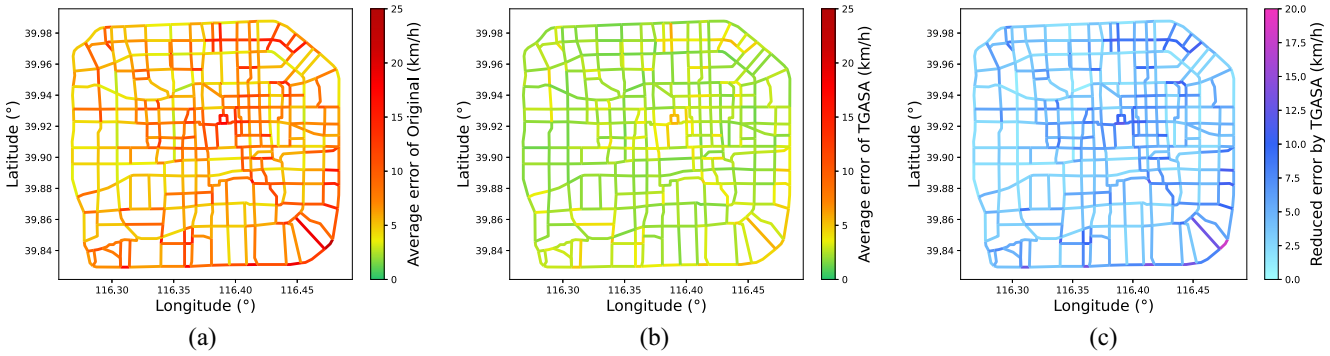


Fig. 11. Improvement of the average error on Monday at 5% sparsity. (a) Estimation error of Original. (b) Estimation error of TGASA. (c) Reduced error by TGASA.

LSTM is better than DNN since they have the same sequential input data and LSTM is specialized in handling sequential data. Commonly, the TGASA outperforms LSTM by 2% for improved percentage (IPV). The DCRNN, TGCN, and STGNCDE achieve accurate estimates for high-sparsity cases, but have varying degrees of recession for low-sparsity cases. They can reduce large errors but are short for high-precision estimation. In addition, they lack the transfer ability since their network parameters are based on a fixed graph structure.

The temporal correlation is more important than the spatial correlation in traffic data. The GraphSAGE leverages only spatial correlation for estimation, while others include the use of temporal correlation. We can see that GraphSAGE has lower performance than others, and its IPVs are always 30% lower than TGASA's. In particular, both GraphSAGE and DNN use simple linear deep neural networks to capture correlations, and the estimation performance of DNN is always much better than GraphSAGE.

2) Temporal Performance: In this section, we present and discuss the estimation results in the time domain. Figs. 7–9 show the average MAEs of 500 road segments for every minute in the daytime from 7:30 A.M. to 10:30 P.M. at the sparsity 5%, 20%, and 50%, respectively. They are plotted in the polar coordinates, where the x-axis is the clock time in minutes and the y-axis is the average error in km/h. The height of the bar represents the average error and the color of the bar is related to its height. There are two items in each polar graph: 1) the outer orange bars represent the average errors of Original and 2) the inner green bars represent the average errors of TGASA. Their daily average MAEs are given in each

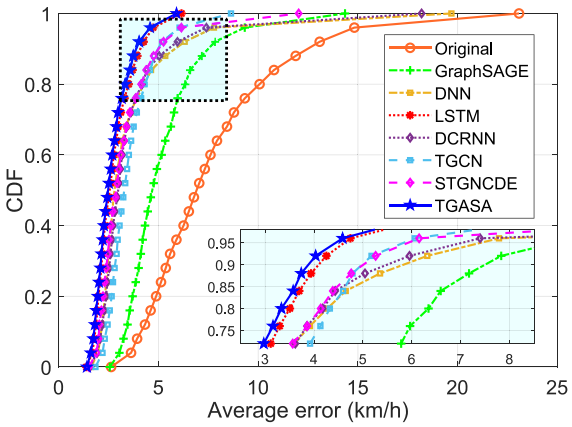


Fig. 12. CDF of average error on Monday.

figure. From these figures, we can observe that the following.

The TGASA has strong robustness and high adaptability. In general, traffic flow characteristics are highly related to the day of the week due to the different travel demands on weekdays and weekends. In the case of 5% sparsity, the IPV of TGASA of these five days are 66.865%, 66.589%, 66.335%, 67.442%, and 68.021%, respectively. With the same sparsity, the promotions are at similar levels for different days, which means that TGASA can adapt to different days of the week. This is because TGASA improves the estimation accuracy mainly by capturing spatio-temporal correlations over relatively short periods in the past, which are fundamental properties of traffic and are similar across days.

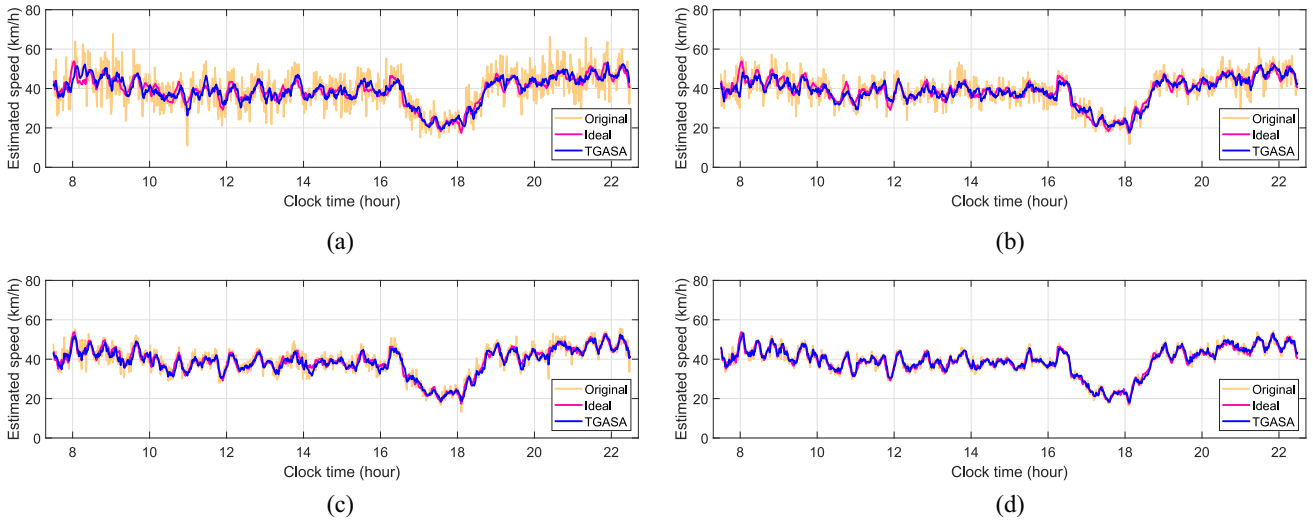


Fig. 13. Recovery results visualization for a road with abundant participants. (a) 5% sparsity. (b) 10% sparsity. (c) 20% sparsity. (d) 40% sparsity.

The TGASA performs stable for different clock times. We can see that the TGASA errors are at the same level throughout the day, with no large errors, except for the special case that the errors on Sunday morning are larger than the errors at other times of the day. This is because the system has only collected less vehicle-driving information on Sunday morning. Overall, the profile of the inner green bars is smooth, meaning that the performance of TGASA is stable and does not change over time.

As the data sparsity grows, the estimation precision increases, but the promotion brought by TGASA decreases. The area formed by the heights of orange bars minus the height of green bars, i.e., the orange part can be seen, is the promotion brought by TGASA. We can see this area has narrowed with the growth of data sparsity. This shows that the amount of information contained in the data set does not increase linearly with the size of the data, which is one of our motivations for using sparse data for TSE.

3) Spatial Performance: In this section, we present and analyze the estimation results in the space domain. From the given three groups of figures about spatial performance, we can observe that the following.

The sparsification of data incurs a high-missing rate and inaccuracy to the Original estimation. Fig. 10 shows three different traffic speed estimation results at 18:30 P.M. on Monday in the case of 5% sparsity, including the ideal estimation, the original estimation, and the estimation obtained by TGASA. Note that the speed of a road segment is set to 0 if there is no vehicle-driving information collected within its range over the past 10 min, and it is plotted in the color pure red. Fig. 10(a) is the ideal estimation, which is the average speed calculated from 100% data. It is notable that there still exists one missing speed road segment. Fig. 10(b) is the Original estimation, which is the average speeds calculated from 5% data. It has a big difference compared with the ideal estimation. On the one hand, we can see that there are many road segments in pure red, meaning that the number of missing speed roads has increased. On the other hand, the Original estimated speeds of most roads are different from the ideal estimation. Fig. 10(c) is the estimation obtained by TGASA. Generally, we can see that it is much more similar to the ideal estimation compared with the Original estimation. In addition, there is no missing speed in the road network since the missing data has been imputed by TGASA with the temporal and spatial correlations.

The TGASA performs well for all the road segments. To further illustrate the spatial performance of TGASA within a day, we present the average of the estimation errors for 900 min for each road segment on Monday for each road segment in Fig. 11. The average estimation error of the Original is shown in Fig. 11(a). We can see that most road segments are colored with red or orange. The average errors of most roads are above 5 km/h. Among 500 road segments, the maximum average estimation error is 23.07 km/h and the minimum average estimation error is 2.63 km/h. Fig. 11(b) shows the average estimation error of TGASA. After recovery, the maximum average error among 500 road segments becomes 5.91 km/h and the minimum average error becomes 1.41 km/h. Most roads are colored with green, meaning that the average errors of most roads are below 5 km/h. The difference between the former two figures is presented in Fig. 11(c), showing the reduced average estimation error by TGASA. The maximum reduced error is 17.96 km/h while the minimum reduced error is 1.09 km/h.

The TGASA has a higher TSE accuracy than others. We plot the cumulative distribution function (CDF) for their averages of 900 min estimation errors on Monday of each road segment in Fig. 12 to compare the performance between the TGASA and the other 6 methods. GraphSAGE has the lowest CDF of all methods, and TGCN has the second lowest CDF with a short tail. The CDFs of the DNN, DCRNN, and STGNCDE are similar, with a long tail. The LSTM has the most comparable performance with the TGASA. From the enlarged subfigure, we can see that their CDFs are similar in shape, but the LSTM is still slightly behind.

4) Individual Road Performance: In this section, we analyze the estimation results of individual roads. Figs. 13 and 14 show the recovery results visualization for the road segment with abundant and fewer participants, respectively. For each

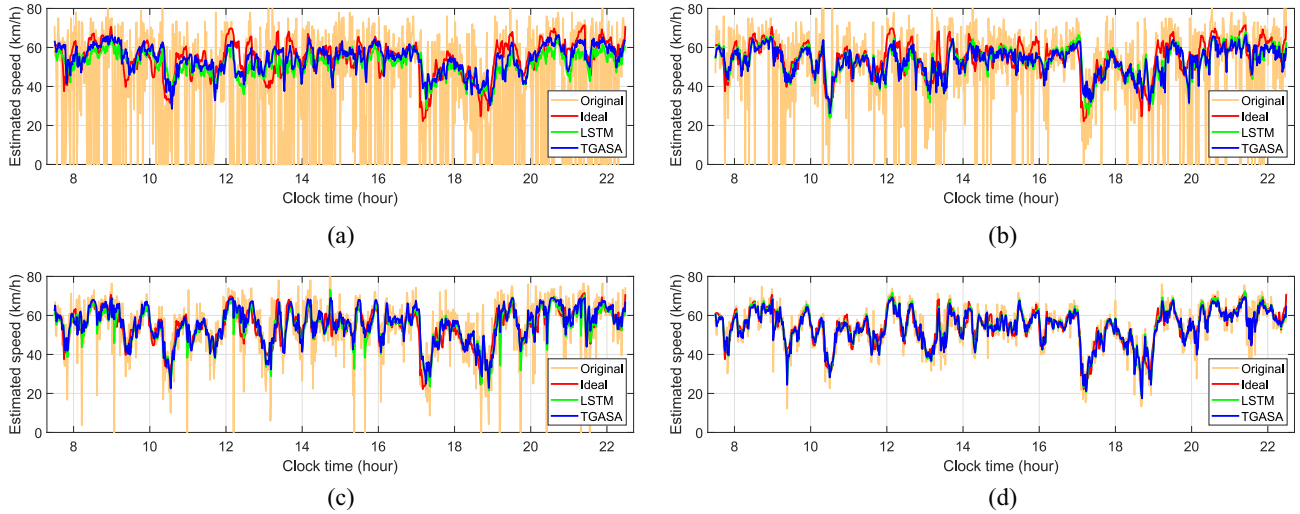


Fig. 14. Recovery results visualization for a high-missing rate road. (a) 5% sparsity. (b) 10% sparsity. (c) 20% sparsity. (d) 40% sparsity.

road, we present 4 cases at different sparsity, including 5%, 10%, 20%, and 40%, where it is doubled in every step. From these two group graphs of estimation results, we can observe that the following.

The TGASA can improve the estimation accuracy and stability. In all 8 graphs, we can see that the TGASA is more similar to the Ideal compared with the Original in both aspects of values and tendencies. The instability and noise of the Original estimation decrease as the sparsity increases. Fig. 13 shows a road segment with abundant participants of four different sparsity cases. The RMSE of the Original is 4.88, 3.41, 2.24, and 1.32 km/h, respectively. The TGASA has reduced the RMSE to 2.11, 1.88, 1.33, and 0.99 km/h, respectively.

The TGASA is robust and has strong speed imputation ability. Fig. 14 shows the estimation results for a road segment with fewer participants, where the speed of Original is assumed to be 0 km/h if no driving information is collected within the time slot. The figure illustrates that the lack of participants results in many time slots with no data available for speed estimation, highlighting the need for a speed imputation capability in the recovery algorithm. For the case of sparsity 5%, 10%, 20% and 40%, the RMSE of Original is 23.07, 13.53, 7.47, and 3.90 km/h, respectively. TGASA has reduced the RMSE to 5.11, 4.67, 3.50, and 2.46 km/h, while LSTM has reduced the RMSE to 6.16, 4.93, 3.68, and 2.47 km/h, respectively. The RMSE of LSTM is 20.55% larger than the RMSE of TGASA when the sparsity is 5%. It is difficult for LSTM to achieve an accurate TSE using only temporal correlation, due to frequent missing data. TGASA has a significant advantage over LSTM in TSE tasks for the roads with fewer participants, as it can use data from other roads for supplement. Thus, TGASA has the synergy of capturing both temporal and spatial correlations to achieve accurate and robust TSE.

VI. Conclusion

In this article, we have studied the cost-effective sparse MCS framework for TSE that leverages driving data from vehicular networks. We analyzed the challenges of MCS data sparsification and its impact on decreasing the estimation accuracy. To address this issue, we have designed the TGASA model to improve the estimation accuracy by effectively leveraging the spatial and temporal correlation of traffic flow. Extensive simulation results on the real-world vehicular network data set have verified the effectiveness of the proposed sparse MCS framework and the superiority of the TGASA model in sequential graph data processing. Our framework offers a cost-effective solution for TSE, especially in scenarios with limited MCS data availability. For future work, we plan to explore the traffic flow prediction based on TSE using sparse MCS data.

References

[1] R. Al Mallah, A. Quintero, and B. Farooq, "Prediction of traffic flow via connected vehicles," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 264–277, Jan. 2022.

[2] A. Abdalrahman and W. Zhuang, "PEV charging infrastructure siting based on spatial–temporal traffic flow distribution," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6115–6125, Nov. 2019.

[3] Y. Xu, H. Zhou, T. Ma, J. Zhao, B. Qian, and X. Shen, "Leveraging multiagent learning for automated vehicles scheduling at nonsignalized intersections," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11427–11439, Jul. 2021.

[4] J. Wang, Y. Huang, Z. Feng, C. Jiang, H. Zhang, and V. C. M. Leung, "Reliable traffic density estimation in vehicular network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6424–6437, Jul. 2018.

[5] C. Zha, J. Xue, Q. Shen, and H. Zhou, "Low-cost traffic perception for road detector data estimation: A deep learning approach," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, 2022, pp. 1–5.

[6] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, 3rd Quart., 2019.

[7] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2022.

[8] M. C. Lucic, X. Wan, H. Ghazzai, and Y. Massoud, "Leveraging intelligent transportation systems and smart vehicles using crowdsourcing: An overview," *Smart Cities*, vol. 3, no. 2, pp. 341–361, 2020.

[9] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary V2X technologies toward the Internet of Vehicles: Challenges and opportunities," *Proc. IEEE*, vol. 108, no. 2, pp. 308–323, Feb. 2020.

[10] W. Wu et al., "AI-native network slicing for 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.

[11] H. Du et al., "Exploring attention-aware network resource allocation for customized Metaverse services," *IEEE Netw.*, early access, Dec. 26, 2022, doi: 10.1109/MNET.128.2200338.

[12] J. Wang, H. Du, Z. Tian, D. Niyato, J. Kang, and X. Shen, "Semantic-aware sensing information transmission for Metaverse: A contest theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5214–5228, Aug. 2023.

[13] Z. Liu, P. Zhou, Z. Li, and M. Li, "Think like a graph: Real-time traffic estimation at city-scale," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2446–2459, Oct. 2019.

[14] L. Wang et al., "SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–28, 2017.

[15] W. Xu et al., "Internet of Vehicles in big data era," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.

[16] F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, "Quality of information in mobile crowdsensing: Survey and research challenges," *ACM Trans. Sensor Netw.*, vol. 13, no. 4, pp. 1–43, 2017.

[17] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.

[18] J. Xue, T. Zhang, W. Wu, H. Zhou, and X. Shen, "Sparse big data for vehicular network traffic flow estimation: A machine learning approach," in *Proc. IEEE Glob. Commun. Conf.*, 2022, pp. 4959–4963.

[19] A. Abdelraouf, M. Abdel-Aty, and N. Mahmoud, "Sequence-to-sequence recurrent graph convolutional networks for traffic estimation and prediction using connected probe vehicle data," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1395–1405, Jan. 2023.

[20] M. Saberi et al., "A simple contagion process describes spreading of traffic jams in urban networks," *Nat. Commun.*, vol. 11, no. 1, p. 16, 2020.

[21] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[22] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.

[23] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[24] O. Gkountouna, D. Pfoser, and A. Züfle, "Traffic flow estimation using probe vehicle data," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, 2020, pp. 579–588.

[25] J. Liu, G. P. Ong, and X. Chen, "GraphSAGE-based traffic speed forecasting for segment network with sparse data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1755–1766, Mar. 2022.

[26] C. Wang, Z. Xie, L. Shao, Z. Zhang, and M. Zhou, "Estimating travel speed of a road section through sparse crowdsensing data," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3486–3495, Sep. 2019.

[27] D. Xu, H. Peng, C. Wei, X. Shang, and H. Li, "Traffic state data imputation: An efficient generating method based on the graph Aggregator," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13084–13093, Aug. 2022.

[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2014, pp. 1724–1734.

[31] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[32] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Trans. GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[33] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transp. A, Transp. Sci.*, vol. 15, no. 2, pp. 1688–1711, 2019.

[34] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[37] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–15.

[39] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1234–1241.

[40] X. Wang et al., "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, 2020, pp. 1082–1092. [Online]. Available: https://doi.org/10.1145/3366423.3380186

[41] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 1–10.

[42] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.

[43] J. A. Rice, *Mathematical Statistics and Data Analysis*. Boston, MA, USA: Cengage Learn., 2006.

[44] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[45] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 6367–6374.

Jianzhe Xue (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China.

His current research interests include Internet of Vehicles, orthogonal time frequency space modulation, and machine learning for wireless communications.

Yunting Xu (Student Member, IEEE) received the B.S. degree in communication engineering from Nanjing University, Nanjing, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering.

He mainly focuses on the dynamic resource management and networking optimization in the field of emerging wireless networks.

Wen Wu (Senior Member, IEEE) received the B.E. degree in information engineering from South China University of Technology, Guangzhou, China, in 2012, the M.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2015, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019.

He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. He is currently an Associate Researcher with the Frontier Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests include 6G networks, network intelligence, and network virtualization.

Tianqi Zhang (Graduate Student Member, IEEE) received the B.S. degree in electronical information science and technology from Nanjing University, Nanjing, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Electronic Science and Engineering.

His current research interests include Internet of Vehicles and intelligent transportation system.

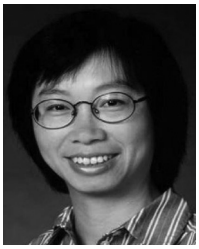
Qinghong Shen received the B.S. degree in information engineering from Xidian University, Xi'an, China, in 1991, and the M.S. degree in circuit and system and the Ph.D. degree in radio physics from Nanjing University, Nanjing, China, in 2000 and 2010, respectively.

He was a Lecturer with the School of Electronic Science and Engineering, Nanjing University, in 2000, where he has been an Associate Professor, since 2004. He has authored more than 50 articles and coedited a textbook. His research interests include circuit and system, wireless sensor networks, and intelligent transportation.

Haibo Zhou (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2014.

From 2014 to 2017, he was a Postdoctoral Fellow with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His research interests include resource management and protocol design in vehicular ad hoc networks, cognitive networks, and space–air–ground integrated networks.

Dr. Zhou was a recipient of the 2019 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award. He served as an Invited Track Co-Chair for ICCC'2019, VTC-Fall'2020 and a TPC Member for many IEEE conferences, including GLOBECOM, ICC, and VTC. He served as an Associate Editor for the IEEE Comsoc technically co-sponsored the *Journal of Communications and Information Networks* from April 2017 to March 2019, and a Guest Editor for the IEEE COMMUNICATIONS MAGAZINE in 2016, *International Journal of Distributed Sensor Networks* (Hindawi) in 2017, and *IET Communications* in 2017. He is currently an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, the IEEE NETWORK MAGAZINE, and the IEEE WIRELESS COMMUNICATIONS LETTER.



Weihua Zhuang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 1993.

She has been with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada, since 1993, where she is a University Professor and a Tier I Canada Research Chair of Wireless Communication Networks.

Dr. Zhuang was the recipient of the 2021 Women's Distinguished Career Award from IEEE Vehicular Technology Society, the 2021 Technical Contribution Award in Cognitive Networks from IEEE Communications Society, and the 2021 R. A. Fessenden Award from IEEE Canada. She was the Technical Program Chair/Co-Chair of IEEE VTC 2017/2016 Fall, the Technical Program Symposia Chair of IEEE Globecom 2011, and an IEEE Communications Society Distinguished Lecturer from 2008 to 2011. She is the President and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013. She is a Fellow of the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada.