

Received November 5, 2020, accepted November 23, 2020, date of publication November 26, 2020, date of current version December 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040864

Traffic State Estimation of Bus Line With Sparse Sampled Data

XIANMIN SONG¹, JING TIAN¹, PENGFEI TAO¹, HAITAO LI¹, AND CONG WU¹

School of Transportation, Jilin University, Changchun 130022, China

Corresponding author: Pengfei Tao (taopengfei@jlu.edu.cn).

This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1600500.

ABSTRACT The traffic state of the bus line is the information basis for the bus company to make bus dispatch and travel time prediction. However, the bus GPS data is severely sparse in time and space coverage of traffic state, due to the long data sampling time interval and low bus departure frequency. Because of ignoring the severe sparseness of the bus data, the existing traffic state methods cannot reconstruct the traffic state accurately. To deal with this problem, a new traffic state estimation method for the bus line, named GAN_BS, is proposed. First, an improved generative adversarial network (GAN-I) is used to generate reasonable bus data. GAN-I aims to find the probability space of the data distribution under sparse sampling. And to reduce the size of the latent space of data, the traffic knowledge is introduced as prior information layers. Then, a traffic adaptive bilateral smoothing method (BS) is used to map discrete bus data into the continuous traffic state. The BS convolves data with a bilateral kernel, which multiplies the local action kernel with a mask of traffic state similarity. Therefore, the BS can maintain transitions between different traffic patterns while separating noise from traffic state. Finally, a set of numerical experiments are performed on the real bus data set in Changchun. The results show that the GAN-I can accurately reproduce the traffic state when the missing rate of data exceeds 50%. And the BS can eliminate the noise better compared with other methods.

INDEX TERMS Traffic state estimation, generative adversarial network, bus line, bilateral smoothing.

I. INTRODUCTION

Reasonable real-time bus dispatch and reliable bus travel time prediction are important means to improve the passenger travel experience. And the traffic state of bus lines is the calculation basis for these tasks. However, due to the low frequency of bus departure and data upload, the sampling rate of bus GPS data is lower than the social vehicles. The traffic state sampled from the bus data will be missing in a large area. It is difficult to infer the real traffic state from the incomplete bus data. Estimating the traffic state of bus lines is a challenging task for bus companies.

In general, the bus departure interval is relatively large (15 minutes in normal situations) and the GPS data sampling interval is generally about 30 seconds. This makes the missing rate of bus data in the time and space more than 50%. Using the extremely sparse data to estimate the traffic state is more uncertain than using the data with a low missing rate.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed¹.

As shown in Fig. 1(a), the social vehicles (driving vehicles other than buses, such as private cars, taxis, etc.) on the road are denser than buses. They can provide more samples than buses can. It is easy to reconstruct the traffic state when the traffic state is effectively sampled by a valid number of points. However, as shown in Fig. 1(b), there are only two buses on the road. The buses have sampled two traffic state patterns respectively. The distance between the sampling locations is large. It is difficult to infer the transition position of two traffic patterns. The goal of this paper is to generate reasonable bus data at the missing location as shown by the dotted boxes in Fig. 1(c). Then, the continuous and smooth traffic state can be reconstructed based on all the bus data shown in Fig. 1(c).

Many methods have been presented to estimate the traffic state (e.g. Kriging [1], ASM [2], GAN [3], etc.). One of the popular filtering methods is ASM, which can eliminate the noise in the data and reconstruct the smooth traffic state. However, ASM requires complete data to compute. Otherwise, it has to borrow irrelevant data from remote locations. Therefore, we need to impute the sparse bus data in advance.

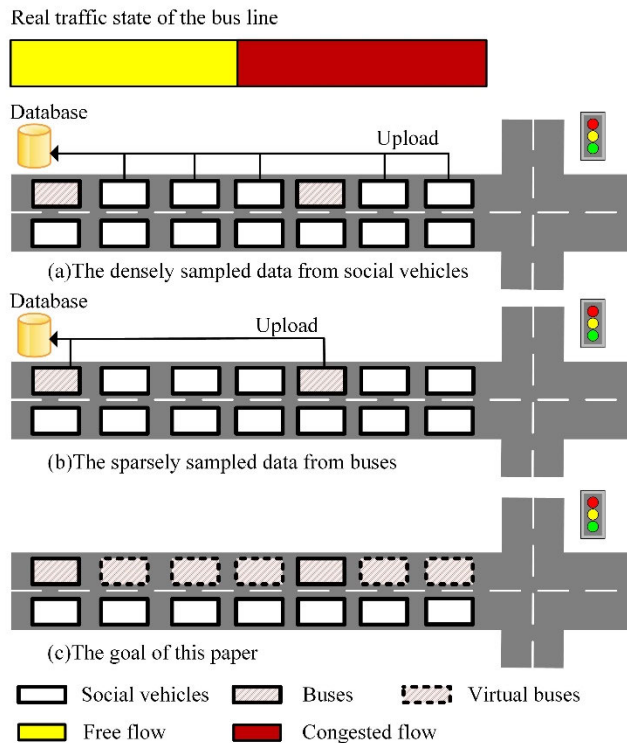


FIGURE 1. Schematic diagram of the traffic state estimation under different frequency sampling.

There are several statistical learning methods that have shown a good ability to deal with low-frequency missing data, such as KNN [4], tensor decomposition [5], LSTM [6]. But when large areas of data are missing, the size of the possibility space of the unobserved location is large. These methods cannot find the optimal solution because of their limited generalization ability. Recently, GAN [7]–[9] has been widely used in the traffic imputation field. Due to the introduction of the discriminator, GAN has an outstanding ability to generate the data as real as possible. However, to the best of the authors’ knowledge, GAN-based studies that considered the high missing rate of bus data are very limited. Generating the complete bus data for traffic state estimation is still a problem to be solved.

To deal with the problem, a new traffic state estimating method for the bus line, named GAN_BS, is proposed in this paper. This model includes two parts: data generation and traffic state reconstruction. In the data generation part, a generative adversarial network is used to generate dense data under severely sparse sampling. This network learns the traffic states probability distribution to infer the real data. To reduce the uncertainty of distribution fitting, a conditional reasoning layer packed with traffic knowledge is introduced. In the traffic state reconstruction part, a bilateral smoothing method is proposed to convert discrete bus data points into the traffic state of the bus line. This method employs the traffic-adaptive convolution to eliminate the noise of the bus data via a localized kernel. To prevent the incorrect estimation

of the edge of the traffic pattern, the smoothing method introduces an additional kernel, which limits the action scope of the localized kernel when the traffic state changes.

The main contributions of this paper can be summarized as follows:

- 1) An improved generative adversarial network is proposed to impute the sparse bus GPS data. This deep learning network can learn the spatial-temporal probability distribution of traffic state from historical data. And the prior knowledge of traffic flow is introduced to make the generated data of the network can conform to the actual traffic state. The valid data generated by the model can provide a data basis for traffic state estimation.
- 2) A traffic bilateral adaptive smoothing method is introduced to reconstruct the continuous traffic state from discrete bus data. This method considers the influence of intersections and bus stops on the traffic state and designs a bilateral kernel. This kernel can not only eliminate noise caused by driving characteristics bus also retains the edge of traffic state switching. The model can effectively express the traffic state of the bus line and provide information for bus management.
- 3) Several popular traffic state estimation methods are evaluated on real-world data with a missing rate of more than 50%. The comparison results provide a comprehensive reference for related research.
- 4) Although there are many traffic state estimation studies based on multi-source data, data sharing has not been completely realized in actual operations. GAN_BS can directly serve bus companies that only have a single data source.

The rest of this paper is organized as follows. Section II reviews the studies on traffic state estimation. Section III presents a traffic state estimation method for bus lines under the condition that the bus GPS data are pathologically sparse sampled. Section IV discusses the experimental results. Finally, Section V concludes the paper.

II. LITERATURE REVIEW

As discussed earlier, traffic state estimation is vital for both traffic managers and passengers. And to estimate the traffic state of bus lines, firstly, we need to generate dense sampling points from sparse data. Then the continuous traffic state is restored from discrete data points. Therefore, this part will discuss related research from two parts: traffic state estimation and missing data imputation.

A. TRAFFIC STATE ESTIMATION

There are many studies to estimate the traffic state of the bus line based on different state metrics (such as speed, travel time, etc.). From the perspective of model types, these methods can be divided into two categories: regression methods and filtering methods.

The regression methods [10], [11] mainly estimate the traffic state of the bus line by fitting the functional relationship

between some traffic variables and the state variables. For example, Yu *et al.* [12] proposed the relevance vector machine to estimate the probabilistic bus headway by considering the travel time of several vehicles ahead and other factors. Yu *et al.* [13] used the support vector machine to learn the relationship between the estimated bus traffic state and the travel time of the former bus. These methods assume that the traffic state detected by the former bus is similar to the current one. This assumption is real in the case of data-intensive sampling. But it may lead to wrong estimates when the bus departure interval is large.

The filtering methods directly process the sampled traffic state data into continuous traffic state data by considering traffic dynamics. For example, Chen *et al.* [14] proposed a self-adaptive exponential smoothing method based on the Kalman Filter to predict the link bus travel time. And some studies [15]–[17] improved the accuracy of the Kalman Filter by introducing traffic flow theory. For example, Duert and Yuan [18] derived the traffic control law from the Lighthill-Whitham-Richards model to estimate the traffic state on the road network. These methods estimated the traffic parameters along with the trend of the time series of the traffic state. Therefore, they are often used when the series has a single point missing. Some researchers have extended the traffic state estimation to consider the spatial-temporal correlation, which improves the robustness of the methods to the lack of more data. One of the popular methods is the adaptive smoothing method (ASM) [2]. It is based on the traffic phenomenon that the free flow wave propagates downstream and the congested flow wave propagates upstream. The kernel function of this filtering method [19] is set to capture the spatial-temporal range of traffic patterns. And F. Rampe *et al.* [20] extended the method to the field of traffic state estimation using floating car data and named this method ARS. The ARS is local low-pass filtering, which can eliminate the noise in the data. But when the data in the local computing domain is insufficient, ARS needs to borrow data with less correlation in the distance to estimate. Overall, the existing methods for traffic state estimation are based on data with a low missing rate. For highly sparse data (such as bus GPS data), it is inappropriate to use existing methods estimate the traffic state.

B. MISSING DATA IMPUTATION

Researchers have proposed various methods for missing data imputation and we discuss these methods from two parts: prediction methods and interpolation methods.

The prediction methods are usually filled in the missing data according to the relationship of the time series or spatial series of traffic state. Historical average [21], ARIMA [33], LSTM [6], [23]–[25], and other methods have been proved to be effective in predicting the data based on the developing trend of traffic state. And the trend of traffic state is analyzed by using the observations from the previous steps. However, there is often no observed samples in the bus line because the interval of bus departure is large. The bus data may be

missing in large blocks. Although deep learning has shown great performance in dealing with the problem of missing data at stationary points for fixed detectors, these methods have poor resistance to the sparseness of bus data.

The interpolation methods regress the model of traffic state relationships by analyzing the spatial-temporal distribution of the traffic. There are several common interpolation methods for estimating traffic state such as Probabilistic principal component analysis (PPCA) [26], [27], tensor decomposition [28]–[31], Convolutional neural network (CNN) [32], [33], auto-encoders [34]–[36], Fuzzy neural network [37], Random forest [38]. Li *et al.* [26] employed the PPCA to estimate the traffic state by extracting the periodic spatial-temporal dependencies in traffic flow. Chen *et al.* [5] used the Bayesian probabilistic matrix factorization to derive missing data based on the similarity of spatial-temporal traffic states. Li *et al.* [36] used two parallel auto-encoders to capture the spatial-temporal dependencies of the traffic state. These methods can deal with the problem of lost data with a low missing rate. This problem can be viewed from a probabilistic perspective. And these deep learning methods calculate the speed v_t at the time t by calculating the conditional probability $p(v_{x,t} | v_m)$, where v_m is the neighboring point in time and space. These methods select the most likely value in the probability space composed of all v_n and v_m . When the number of observations in the field is too small, the uncertainty of the data increases. It becomes harder to estimate missing data accurately.

Recently, generative adversarial networks (GAN) have gained increasing interest in traffic imputation because of its excellent capability in generating data [3], [7], [8], [39]. GAN can calculate the joint distribution $p(v_{x,t}, v_m)$ based on sampling points of surrounding locations without an explicit sequence. And the introduction of the discriminator has improved the network's ability to generate data when the data are severely sparse. For example, Xu *et al.* [9] used DEEPWALK technology to embed the road network structure in GAN and carried out the traffic state estimation based on the correlation between links. Liang *et al.* [40] used the LSTM as the network layer of GAN to capture the correlation of traffic state in space and time to estimate the traffic flow and density. However, GAN cannot eliminate the interference of noise in the data and directly give a continuous traffic state.

As mentioned above, the bus data are sparser than social vehicle data. Most of the existing methods are difficult to estimate the traffic state when the data is missing in blocks. Some advanced methods, such as GAN, can generate data under sparse sampling, but it is difficult to generate the continuous traffic state at the same time. To solve the above problem, a GAN_BS model, which can reconstruct a smooth and accurate traffic state, is proposed in this study.

III. METHOD

This section aims to propose a traffic state estimate method for the bus line, which uses the sparse and noisy bus GPS data. To achieve this task, first, the generative adversarial network

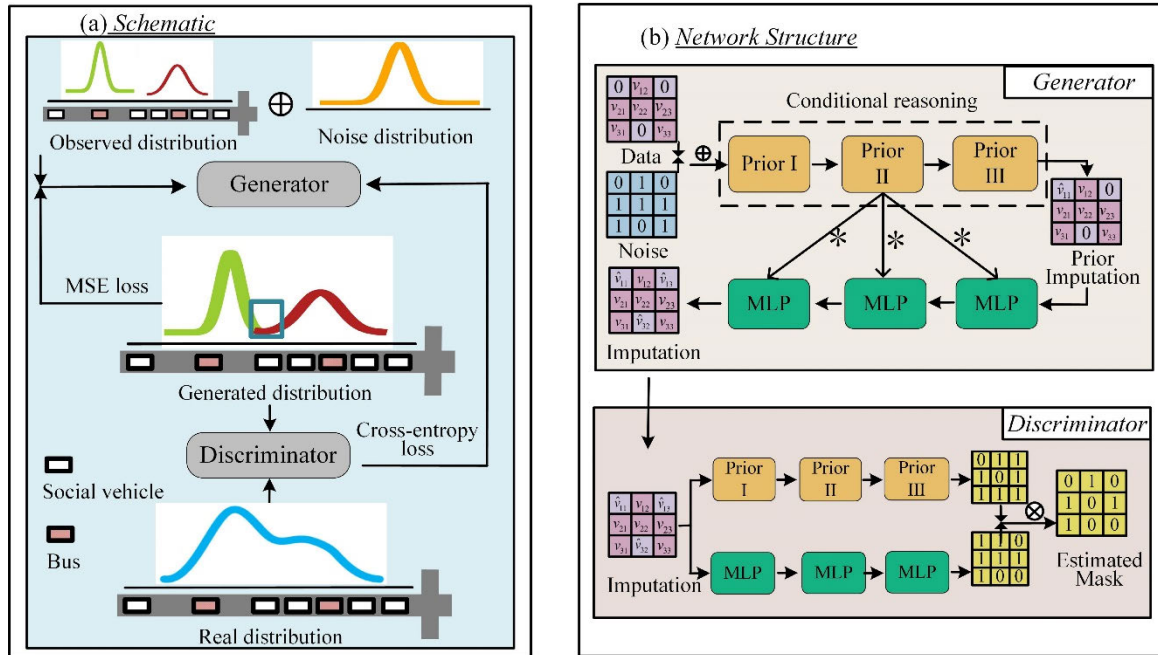


FIGURE 2. Representation of the bus data imputation model using a generative adversarial network. (a) Schematic of the working principle of GAN-I. (b) Network structure of GAN-I.

is introduced to impute the missing bus data. Then, a bilateral smoothing method is proposed to construct the continuous traffic state from the discrete bus data.

A. THE BUS DATA GENERATION BASED ON THE GENERATIVE ADVERSARIAL NETWORK

1) GRID PROCESSING OF BUS LINES

To make the calculation easy, the bus line is gridded, and the corresponding traffic state is recorded as a two-dimensional matrix in this paper. The trajectory of a bus can be written as a function $x_{bus}(t)$, and the bus speed $v(t)$ is the derivative of x . We assume that the bus starts from point (x, t) and travels through a road section of length Δx in the time interval Δt . And the traffic state in the space-time domain $[x + \Delta x, t + \Delta t]$ has been sampled as the bus speed $v(t)$. Considering that the space domain occupied by a bus includes the length of the bus and the saturation space headway, the sum x_c of these two factors is used to modify the space-time domain of traffic state collection. Therefore, the Mask function M marks whether the traffic state information is observed at point (x, t) and can be expressed as

$$M(x, t) = \begin{cases} 1 & \text{if } x_{bus}(t) \leq x \leq x_{bus}(t)+x_c \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

As shown in Fig. 2, the input of GAN includes the mask matrix, noise matrix, and speed matrix. Accordingly, when the $M(x, t)$ is 1, the speed $\tilde{v}(x, t)$ is equal to the average bus speed, and when the $M(x, t)$ is 0, the $\tilde{v}(x, t)$ uses the noise

$z(x, t)$ sampled from a specific distribution $p(z)$ to fill in.

$$\tilde{v}(x, t) = \begin{cases} v(x, t) & M(x, t) = 1 \\ z(x, t) & M(x, t) = 0 \end{cases} \quad (2)$$

2) GENERATIVE ADVERSARIAL NETWORK WITH TRAFFIC PRIOR KNOWLEDGE

Due to the large interval of bus departure and GPS data collection, the missing rate of bus data is high. It requires that the data imputation algorithm can fit the distribution of traffic state without dense observations. In this part, an improved generative adversarial network (GAN-I) is proposed to impute the bus data with a high missing rate. Fig. 2 shows the overall network architecture included two parts, a generator G , and a discriminator D . The generator G attempts to estimate the bus data \hat{v} that match the real traffic state as much as possible. And the discriminator D aims to find the fake ones of \hat{v} by estimating the Mask \hat{M} . Writing $\hat{M} = D(\hat{v})$, the objective of GAN is the minimax problem given by

$$\min_G \max_D E \left[M \log(\hat{M}) + (1 - M) \log(1 - \hat{M}) \right] \quad (3)$$

The process of solving (3) is similar to the game between G and D . The D is optimized to distinguish the estimated data by maxing the (3). The G accepts feedback from the D to improve its generation ability. And the performance of G is measured by (3). When the result of (3) is reduced, it indicates that the generated data is closer to truth and confuses the D . This interaction makes the generation of GAN is better than ordinary deep learning.

As shown in Fig. 2, the generator G regards the bus data as the sampling points of the traffic state spatial-temporal distribution $P(v, \tilde{v})$. Our model is to learn the continuous traffic state distribution from these sparse sampling points to produce the matrix \hat{v} . Before learning the patterns of the traffic state, we introduce three prior knowledge of traffic flow to guide the network to impute the bus data. In this section, these traffic laws are packaged as a conditional reasoning layer

$$\hat{v} = f(\tilde{v}, k) \quad (4)$$

where \tilde{v} is the input data, k is the prior knowledge set of traffic, \hat{v} is the reasoning output and f is the default activation function.

The application of three traffic prior information in GAN-I is shown below.

Prior I: There are fewer cars on the road in the early morning or late at night. But it is easy to know that the traffic state is in a free flow state. This condition can be expressed as

$$f(\tilde{v}(x, t)) = \begin{cases} v_{free} & t \leq t_{mor} \text{ or } t \geq t_{eve} \\ \tilde{v} & t_{mor} \leq t \leq t_{eve} \end{cases} \quad (5)$$

where t is the moment of the traffic state to be estimated, t_{mor} and t_{eve} are the critical values triggered by the Prior I, respectively representing the end time and the start time of the free state from midnight to early morning, \tilde{v} is the speed to be estimated later and v_{free} is the speed in free flow.

Prior II: The gradually changing traffic state on urban roads may be interrupted at the intersections and the bus stops due to signal control and passenger boarding and alighting, respectively. We regard the road intersections and the bus stops as the demarcation points of the traffic state and divide the bus line into several links. A step function S is used as a coefficient of activation function in the network. The S controls each neuron as a valve to ensure that each neuron only processes the link data of its corresponding position. And the S can be expressed as

$$S(x) = \begin{cases} 1 & x_{pre} \leq x \leq x_{next} \\ 0 & x \geq x_{next} \text{ or } x \leq x_{pre} \end{cases} \quad (6)$$

where x is the position of the traffic state to be estimated, where x_{pre} and x_{next} are the locations of the upstream intersection (or bus stop) and the downstream intersection (or bus stop) closest to x .

Prior III: Some areas adopt bus signal priority control to reduce bus delays at intersections. We assume that the traffic of the bus line under the signal absolute priority control keeps flowing. Therefore, a binary variable F is used to mark whether the intersection performs signal absolute priority. When $F = 1$ (there is bus priority at the intersection), the traffic state within l meters upstream of the intersection can be imputed in advance. And the traffic state can be estimated as

$$f(\tilde{v}(x, t)) = \begin{cases} v_{free} & k \leq k_{cre} \\ \frac{1}{k} \sum_{i=1}^k \tilde{v}_i & k > k_{cre} \end{cases} \quad (7)$$

where k is the number of sampled points in the space-time domain $[x - l, x]$ meters $\times [t - h, t]$ minutes and the k_{cre} is the confidence value of the k .

Data other than the above cases need to be imputed by the generator. To learn the traffic state spatial-temporal distribution, we reshape the two-dimensional input matrix into one-dimensional as GAN-I's input. And the shared multi-layer perceptron (MLP) is used in the generator to find the traffic correlations in space and time. The form of this hidden layer is

$$\hat{v} = f_{sig}(w\tilde{v} + b) \quad (8)$$

where w and b are weights and biases, the f_{sig} is the activation function which is considered as the logistic sigmoid function in this paper.

$$f_{sig}(\tilde{v}) = \frac{1}{1 + e^{-\tilde{v}}} \quad (9)$$

Then, a batch norm layer is used to normalize the result \hat{v} and improve the imputed accuracy of the algorithm. The normalized speed can be calculated as

$$v_{norm} = \frac{\hat{v} - \mu}{\sqrt{\sigma_{norm}^2 + \varepsilon}} \gamma + \beta \quad (10)$$

where σ_{norm} and μ are the standard deviation and average of \hat{v} , respectively. ε is a hyper-parameter. γ and β are the parameters to be learned.

The generator consists of several hidden layers mentioned above. The output of each layer is utilized as the input of the next layer. And the last layer maps the traffic feature extracted by the previous layers to the final imputed bus speed \hat{v} .

As in the GAN framework, discriminator D is used as an adversary to train generator. As an independent network, the discriminator is trained separately. As shown in Fig. 2, the discriminator receives the imputation results of the generator and estimates the Mask \hat{M} . Under the supervision of the real Mask M , the discriminator learns the ability to distinguish the observed and estimated data. And the discrimination results will in turn prompt the generator to optimize the generation performance in the next step. To ensure that the discriminator works well, the discriminator first marks the positions imputed by the prior conditions. Then the discriminator uses MLPs to find other estimated positions.

3) PARAMETER OPTIMIZATION

The objective function in GAN includes generator loss and discriminator loss. The generator loss includes the loss for the reconstructed data and the imputed data according to different goals. We use the mean squared error (MSE) to make the reconstructed data as close to the observed data as possible. And the imputation loss receives feedback from the discriminator to ensure that the imputed data can fool the discriminator. Overall, we train the generator end-to-end by

minimizing the L_G

$$L_G = - \sum_{i=1}^h (1 - m_i) \log(\hat{m}_i) + \alpha \sum_{i=1}^h \|v_i - \hat{v}_i\|_2 \quad (11)$$

where v_i and \hat{v}_i represent the observed speed value and estimated one, h is the total number of the training data, α is a hyper-parameter, m_i and \hat{m}_i represent the observed mask value and estimated one.

The discriminator identifies whether the data in the matrix output by the generator is true or false as a classification problem. Therefore, the discriminator loss L_D is defined as the cross-entropy loss between the estimated mask and the real mask

$$L_D = \sum_{i=1}^h m_i \log(\hat{m}_i) + (1 - m_i) \log(1 - \hat{m}_i) \quad (12)$$

In GAN, the generator and discriminator are optimized alternatively. Since the learning rate of the commonly used stochastic gradient descent (SGD) method is fixed, using SGD may fall into a local sub-optimal solution. Recently, many studies [41]–[43] have used Adam to find optimal solutions and verified its effectiveness through numerical examples. Therefore, we use Adam [44] instead of SGD to optimize the network. The training process is demonstrated in Algorithm 1. During the testing, GAN generates the bus data covering the entire bus line when the bus uploads real-time detection data.

B. THE TRAFFIC STATE RECONSTRUCTION BASED ON THE ADAPTIVE BILATERAL SMOOTHING METHOD

In this part, we want to average the speed of several sample buses and get the smoothing traffic state at location x_0 . As shown in Fig. 3, there is a set of buses that depart at a specified time interval. These buses collect the speed of location x_0 at different times, and the obtained speed curve of the traffic state is fluctuating. However, the final traffic state estimated result is supposed to vary gradually (like the dotted line in Fig. 3 (b)). Because the GAN used above cannot guarantee that the generated traffic state is smooth, a traffic state smoothing method is needed to eliminate noise and reconstruct the traffic state of the bus line accurately.

The traffic adaptive bilateral smoothing method aims to reconstruct the smooth traffic state by performing a spatial-temporal collaborative filter on sparse and noisy traffic data. It employs the convolution to derive a continuous speed field via a localized kernel. This smoothing process is expressed as

$$V(x, t) = \frac{\int_0^L \int_0^T \phi(x - x_i, t - t_i) v(x_i, t_i) dx dt}{\int_0^L \int_0^T \phi(x - x_i, t - t_i) dx dt} \quad (13)$$

where the kernel $\phi(x, t)$ determines the correlation of the traffic state in time and space. And the kernel function designed in this paper takes into account the locality and similarity of the traffic state.

Algorithm 1 GAN Training Algorithm

Input: Incomplete speed matrix v , mask A , noise z .

Output: Complete bus speed matrix V

- 1: Initialize θ, w
- 2: **While** θ, w has not converged **do**
- 3: Select k samples (v, m, z) from the $\{(V, M, Z)\}$
- 4: **(1) Discriminator optimization**
- 5: **for** k steps **do**
- 6: $\tilde{v} = v \times m + z \times (1 - m)$
- 7: $\hat{v} = G(\tilde{v})$
- 8: Update discriminator using Adam optimizer:
- 9: $\theta = \theta + \nabla_{\theta} \frac{1}{k} \sum_{i=1}^k L_D^{(i)}(m^{(i)}, D(\hat{v}^{(i)}))$
- 10: **end for**
- 11: **(2) Generator optimization**
- 12: **for** k steps **do**
- 13: $\tilde{v} = v \times m + z \times (1 - m)$
- 14: Update generator using Adam optimizer:
- 15: $w = w + \nabla_w \frac{1}{k} \sum_{i=1}^k L_G^{(i)}(v^{(i)}, G(\tilde{v}^{(i)}))$
- 16: **end for**
- 17: **End while**

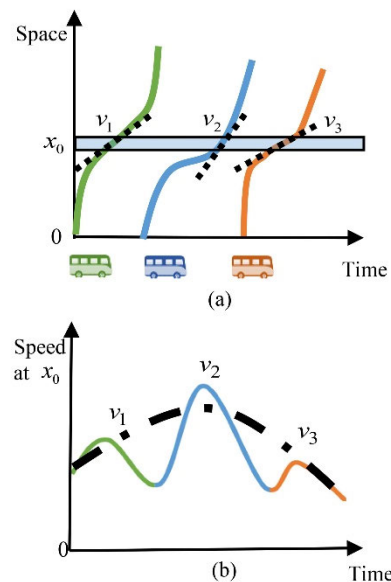


FIGURE 3. Schematic diagram of spatial-temporal superposition of multiple bus data. Picture (a) shows the trajectory of different buses. In the picture (b), the color lines represent the speeds collected by different buses. The dotted line represents the desired smoothly changing speed curve.

The locality of the traffic state means that the closer the distance in the time and space, the more similar the traffic state. And a bivariate Gaussian function is selected as the local smoothing kernel in this paper, which is supposed to increase when the distance between target location (x, t) and neighboring location (x_i, t_i) decreases. To adapt the traffic dynamics, the BS skews this isotropic kernel by introducing the characteristic wave speed parameters.

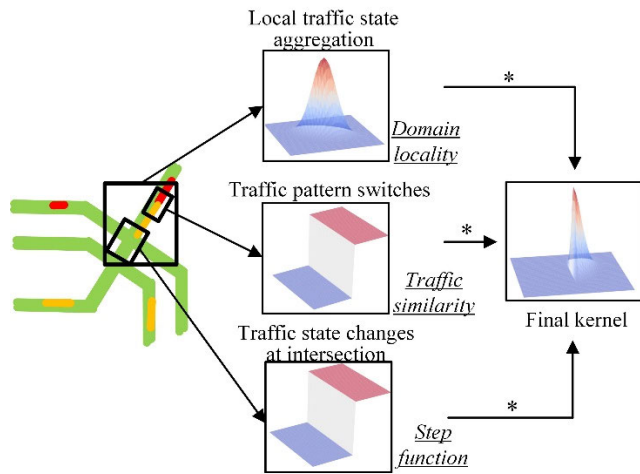


FIGURE 4. The kernel of traffic adaptive bilateral smoothing method.

The characteristic wave speed means the slope of the line between the two traffic state points in the flow-density curve and represents the propagation speed of the traffic state. The inclined Gaussian kernel is formed as the second term on the right side of (14) and shown as the kernel of domain locality in Fig. 4. The action scope of the inclined kernel can approximate the spatial-temporal propagation plane of the traffic state.

It is worth noting that the local smoothing’s assumption of slow variations in traffic state may fail at edges of traffic pattern transition. There is not a clear separation of different traffic states in the results of local smoothing, and some features such as the boundaries of the neighboring free flow wave may shrink. Therefore, the traffic state similarity is also considered in the kernel to maintain the dividing line between different traffic state patterns. The idea of the traffic state smoothing method based on the similarity is to aggregate the bus speed data with weights that decline with dissimilarity in a traffic state. The similarity kernel is formed as the third term on the right side of (14) and shown as the kernel of traffic similarity in Fig. 4. However, domain locality is still a necessary concept. Only using the state similarity to smooth the traffic state make no sense because the speed far away from point (x, t) should not affect the value at (x, t) . The appropriate solution is to combine the traffic patterns similarity and domain locality. In the flat area of the traffic state where the speed changes a few, the corresponding weight of the traffic state similarity tends to be the same. In this situation, the domain weight plays a major role, which is equivalent to perform the Gaussian filtering in this area. In the edge area of the traffic pattern, the speed changes rapidly. The difference in the similarity kernel in this area becomes larger. Therefore, the edge information can be maintained.

The typical switching edges of traffic state patterns in the bus lines are the intersections and bus stops that the bus lines pass. And the impact of this road structure on the traffic state

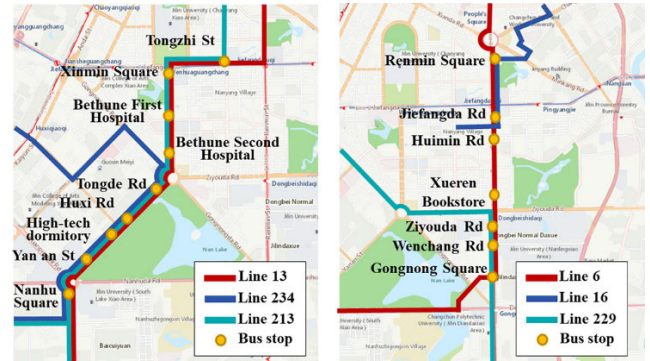


FIGURE 5. Map of test bus lines and their shared bus lines.

can be used as prior knowledge to be introduced into the smoothing method. Therefore, the step function (14) shown in Fig. 4 is introduced to limit the aggregated range of bus speed not to exceed the range of the homogeneous state. Finally, the bilateral kernel is formed as

$$\begin{aligned} \phi_b(x - x_i, t - t_i, v - v_i, x) &= S(x) \exp\left(-\frac{\|x - x_i\|^2}{2\sigma_x^2} - \frac{\|t - t_i\|^2}{2\sigma_t^2}\right) \exp\left(-\frac{\|v - v_i\|^2}{2\sigma_s^2}\right) \end{aligned} \quad (14)$$

where the constants σ_x and σ_t decide the width of the spatial-temporal action scope of the kernel, the constants σ_s decide the range scope of speed difference.

The traffic state is simply divided into two patterns: the free flow V_{free} and congested flow V_{cong} . And the traffic state in any patterns can be seen as a superposition of two speed fields via a convex combination as follows:

$$V(x, t) = w(x, t) V_{cong}(x, t) + [1 - w(x, t)] V_{free}(x, t) \quad (15)$$

where weight $w(x, t)$ dynamically controls the superposition ratio of the two speed fields according to the detection information. In this paper, $w(x, t)$ is a smooth sigmoid function which can ensure that $w(x, t)$ is 1 in the congested state and $w(x, t)$ is 0 in the free flow state.

$$w(V_{free}, V_{cong}) = \frac{1}{2} \left[1 + \tanh\left(\frac{V_c - \min(V_{free}, V_{cong})}{\Delta V}\right) \right] \quad (16)$$

where V_c is threshold between the free flow and congested flow and ΔV is the transition width.

The speed fields in two traffic patterns with corresponding characteristic wave speed are constructed by

$$\begin{aligned} V_{free}(x, t) &= \frac{\int_0^L \int_0^T \phi_b\left(x - x_i, t - t_i - \frac{x-x_i}{c_{free}}, v - v_i\right) v(x_i, t_i) dx dt}{\int_0^L \int_0^T \phi_b\left(x - x_i, t - t_i - \frac{x-x_i}{c_{free}}, v - v_i\right) dx dt} \end{aligned} \quad (17)$$

TABLE 1. Information about the bus lines used for testing.

Bus line	Length (meters)	Shared lines	Numbers of bus stops passed	Numbers of intersections passed	Starting bus stop	Ending bus stop
Line 13	4820	Line 213 Line 234	9	12	Tongzhi Street stop	Nanhu Square Stop
Line 6	3620	Line 16 Line 229	7	9	People's Square Stop	Gongnong Square Stop

$$V_{cong}(x, t) = \frac{\int_0^L \int_0^T \phi_b \left(x - x_i, t - t_i - \frac{x-x_i}{c_{cong}}, v - v_i \right) v(x_i, t_i) dx dt}{\int_0^L \int_0^T \phi_b \left(x - x_i, t - t_i - \frac{x-x_i}{c_{cong}}, v - v_i \right) dx dt} \tag{18}$$

where c_{free} and c_{cong} are the characteristic wave speed in free-flow traffic and congested traffic, respectively.

IV. EXPERIMENTS AND RESULTS

In this section, we discuss the performance of GAN_BS. And we use the real-world bus GPS data to compare the accuracy of several estimate models under the different traffic state scenarios. This section contains four parts: dataset introduction, evaluation index for the performance of traffic state estimation, comparison methods parameter setting, and methods performance analysis.

A. TEST DATA SET

The numerical examples perform on the bus GPS data collected from the Public Transport Corporation in Changchun, China. To verify the reliability of GAN_BS for different road environments, we used two bus routes, Line 6 (the exclusive bus lane) and Line 13 (the non-exclusive bus lane). The data missing rate of the two lines is about 50%. So they can be used to test the model's ability to process data sparsity. Moreover, these two lines are the main road of Changchun City, and their traffic conditions are changeable. So they can be used to verify the adaptability of the model to different traffic patterns. At the same time, we also use GPS data from other lines that overlap with the selected lines. The experiment uses two sections taken from two bus lines. And the detailed information of the sections is shown in Table 1. The geographies of the lines are shown in Fig. 5.

These GPS trajectory data were collected from October 6, 2017, to January 16, 2018, with an updating frequency of 30 seconds from 06:30 to 19:00. It is worth noting that how to set the bus GPS data aggregation scale in time and space. If the aggregation scale is too large, there is a risk of averaging different traffic states. But, if the scale is too small, it will increase the computational burden. By referencing research [2] and [29], these GPS data are aggregated as spatial-temporal matrix data with a 5-minutes time interval and a 20-meters space interval. We take the data from the first 70 days as our training set and the rest as the test set. In this case, the missing data rate of the prepared dataset is

around 50%. To verify the ability of GAN_BS to process severely sparse data, we constructed a test set by randomly removing 20% of the original data. The performance of the model can be evaluated by comparing the removed data and estimated data of the corresponding location.

B. MODEL SETTINGS AND INDEX OF PERFORMANCE

The model set contains two parts: the parameters of the GAN and the parameters of the BS. And the two test data sets use the same parameters of these two parts.

The parameters of the GAN are set as follows: the generator and the discriminator both have five hidden layers. These hidden layers are the fully connected layers and use the sigmoid function as their activation function. The number of hidden units in the network layer will affect the accuracy of estimation. We calculated the error of the estimation result when the number of units in each layer ranges from 250 to 900 (step size is 50). Finally, we found that the performance of GAN-I is best when the numbers of units are taken as 750,500,300,750, space number × time number, respectively.

The parameters of the bilateral smoothing method are also determined by the traversal method. They are set as follows: the characteristic propagation velocity under free traffic flow c_{free} is +30km/h and the characteristic propagation velocity under congestion traffic flow c_{cong} is -15km/h. These two parameters are set for reference from the literature [2], [19]. The parameters of the action scope of the kernel $\sigma_x, \sigma_t, \sigma_s$ are 500 meters, 10 minutes, 10 km/h, respectively.

To evaluate the effectiveness of the proposed model, we use three performance measures, which are the mean absolute error (MAE), the normalized mean square error (NMSE), and the root mean square error (RMSE). To measure the error of the overall traffic state estimation, MAE and RMSE are usually used in similar studies. However, these two indicators cannot reflect the ratio of the error to the observed value. We have introduced a relative error indicator, NMSE, which can measure the estimated performance under different traffic patterns. These three indicators are defined as follows

$$MAE = \frac{1}{n} \sum_{i=1}^n |v_i - \hat{v}_i| \tag{19}$$

$$NMSE = \frac{\sum_{i=1}^n |v_i - \hat{v}_i|^2}{\sum_{i=1}^n v_i^2} \tag{20}$$

TABLE 2. Comparison performance with other methods under various aggregation time scale for line 13.

Statistics	RMSE					MAE					NMSE				
	80%	70%	60%	50%	45%	80%	70%	60%	50%	45%	80%	70%	60%	50%	45%
Missing rate															
Co-kriging	9.75	9.82	9.94	10.10	10.15	9.44	9.45	9.78	9.98	10.07	0.25	0.25	0.25	0.35	0.27
KNN	9.17	9.58	9.83	9.59	9.83	7.11	7.38	7.59	7.43	7.55	0.22	0.24	0.25	0.31	0.25
TAS_LR	8.16	8.22	8.26	8.17	8.46	8.10	8.11	8.11	8.05	8.17	0.18	0.17	0.17	0.23	0.18
BGCP	7.75	7.99	8.49	7.57	9.49	6.27	6.45	6.90	5.93	7.60	0.15	0.16	0.18	0.19	0.22
PD-GAN	8.19	10.65	9.90	8.72	9.07	6.55	8.28	7.77	6.58	7.19	0.18	0.29	0.25	0.25	0.21
BS	8.15	8.72	9.14	8.18	9.63	6.35	6.71	7.05	5.93	7.32	0.17	0.19	0.21	0.22	0.23
GAN-I	8.17	8.33	8.61	8.13	8.25	6.55	6.64	6.91	6.24	6.62	0.17	0.17	0.18	0.22	0.17
GAN_BS	6.07	6.26	6.43	5.65	6.12	4.74	4.87	4.98	4.17	4.73	0.10	0.10	0.11	0.11	0.10

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^d (v_i - \hat{v}_i)^2} \quad (21)$$

where n is the total number of the testing data, v_i and \hat{v}_i represent the observed speed value and estimated one.

C. COMPARATIVE EXPERIMENTS

The comparative experiments in this paper include three parts. First, we conduct an ablation experiment to verify the effectiveness of each component in GAN_BS. Second, the performance of GAN_BS is tested by comparing it with some popular estimation methods. Last, we compare the accuracy of the methods in different periods to see its adaptability to different traffic patterns.

1) COMPARED UNDER ABLATION STUDY

To evaluate the effect of two components in the method proposed by this paper, we use the GAN-I and the bilateral smoothing method separately to estimate the traffic state and compare the results with the full pipeline method. To verify the performance of the methods using data with a high missing rate, we set the scale of GPS data aggregation in time to 1 minute, 2 minutes, 3 minutes, 5 minutes, and 10 minutes, respectively. And the data missing rate corresponding to the input matrix under these different time aggregation scales is about 80%, 70%, 60%, 50%, 45%, respectively.

2) COMPARED WITH OTHER METHODS

In this part, the proposed method compares with other classic and advanced algorithms mentioned above under the data with different degrees of sparsity. The compared traffic state estimation algorithms are as follows:

- (1) Co-kriging: This method proposed by Bae *et al.* [1] uses the uniformity of the traffic state in the time-space domain to regress the traffic state random field. We use the Gaussian model as the variogram of this method and set the minimum estimate of error $n = 0.01$, the maximum dissimilarity $s = 20$, and the distance $r = 10$.

- (2) KNN: This algorithm adopted by Tak *et al.* [4] searches for the first k numbers of historical data that are most similar to the divided section to estimate its traffic states. We set K to 10, which gets the best accuracy among values of k from 5 to 20.
- (3) TAS-LR: This method has been used for traffic state estimation in [45], which aims to explore the spatial-temporal relationship of the traffic state through low-rank decomposition. In this study, the latent rank $r = 10$, the number of neighbors $k = 10$, the parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are set as 0.5, 10, 5, and 5, respectively.
- (4) BGCP: This algorithm proposed by [5] extends the Bayesian probability decomposition model to the imputation problem of the high-order tensor of traffic state. We use the third-order tensor (space number \times day \times time number) as the input of this method.
- (5) PD-GAN: This traffic state estimation method proposed by [7] uses parallel data to be a temporal hint for GAN. We use three convolutional layers as hidden layers in the generator and discriminator. The filter size of the convolutional layers is 3×3 , and the number of kernels in each hidden layer is 150.

All the methods use the spatial-temporal matrix of speed as input so that they can consider the time and space relationship of the traffic state at the same time. And each model for comparison has been carefully tuned.

3) COMPARED UNDER DIFFERENT TRAFFIC PATTERNS

To evaluate the traffic state estimation performance under the different traffic state conditions, we have selected three time zones with different traffic patterns of 06:30, 09:30, and 18:30 to analyze the performance.

D. ANALYSIS OF RESULTS

1) PERFORMANCE COMPARISON UNDER ABLATION STUDY

The calculated error indexes of the ablation study are shown in Table 2 and Table 3. With the aggregated interval changing from 1 minute to 10 minutes, the combination of the GAN-I and ARS method improves the accuracy of the two separate

TABLE 3. Comparison performance with other methods under various aggregation time scale for line 6.

Statistics	RMSE					MAE					NMSE				
	Missing rate	80%	70%	60%	50%	80%	70%	60%	50%	45%	80%	70%	60%	50%	45%
Co-kriging	10.00	10.15	10.23	10.25	10.24	9.84	10.01	10.13	10.18	10.18	0.31	0.32	0.33	0.33	0.35
KNN	9.96	9.27	9.27	9.27	9.24	9.65	9.21	9.21	9.21	9.19	0.31	0.27	0.27	0.28	0.28
TAS_LR	8.20	8.26	8.35	8.41	8.54	8.14	8.16	8.19	8.16	8.18	0.21	0.21	0.22	0.23	0.24
BGCP	6.12	6.34	6.35	6.28	5.78	4.82	5.01	5.00	4.96	4.49	0.12	0.13	0.13	0.13	0.11
PD-GAN	11.29	10.71	10.35	9.92	9.25	8.83	8.44	8.08	7.74	7.19	0.40	0.36	0.34	0.31	0.29
BS	7.18	6.96	6.00	5.23	4.04	5.73	5.50	4.72	4.06	3.10	0.16	0.15	0.11	0.09	0.05
GAN-I	8.76	10.05	8.96	8.66	8.54	7.04	7.91	7.04	6.77	6.68	0.24	0.32	0.25	0.24	0.24
GAN_BS	5.76	5.83	5.76	5.55	5.07	4.25	4.24	4.15	3.96	3.57	0.10	0.11	0.10	0.10	0.09

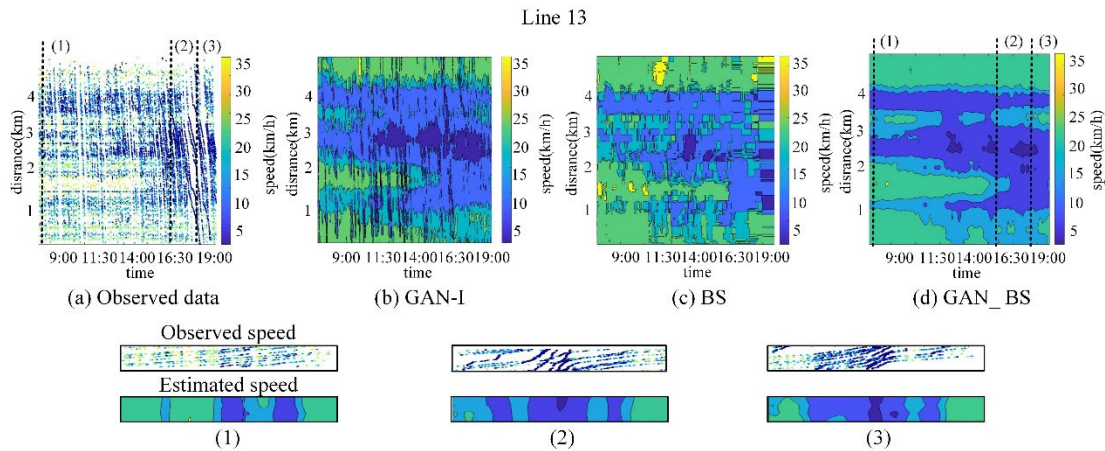


FIGURE 6. Traffic state heat maps of ablation experiment results (Line 13).

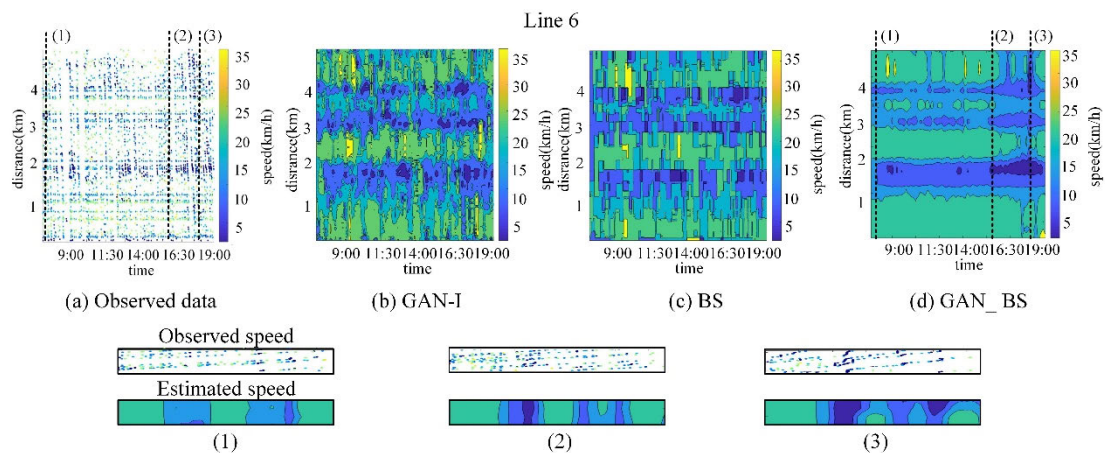


FIGURE 7. Traffic state heat maps of ablation experiment results (Line 6).

methods by nearly 20% in most cases. The difference in the results of the three methods can be intuitively seen from Fig. 6 and Fig. 7. The traffic state generated by ARS is messy because it is a local filter and must borrow data from

unrelated locations in the distance when the entire block of data is missing. The GAN-I can restore the distribution characteristics of traffic state in time and space. But it cannot avoid the interference of noise, and its generated state is

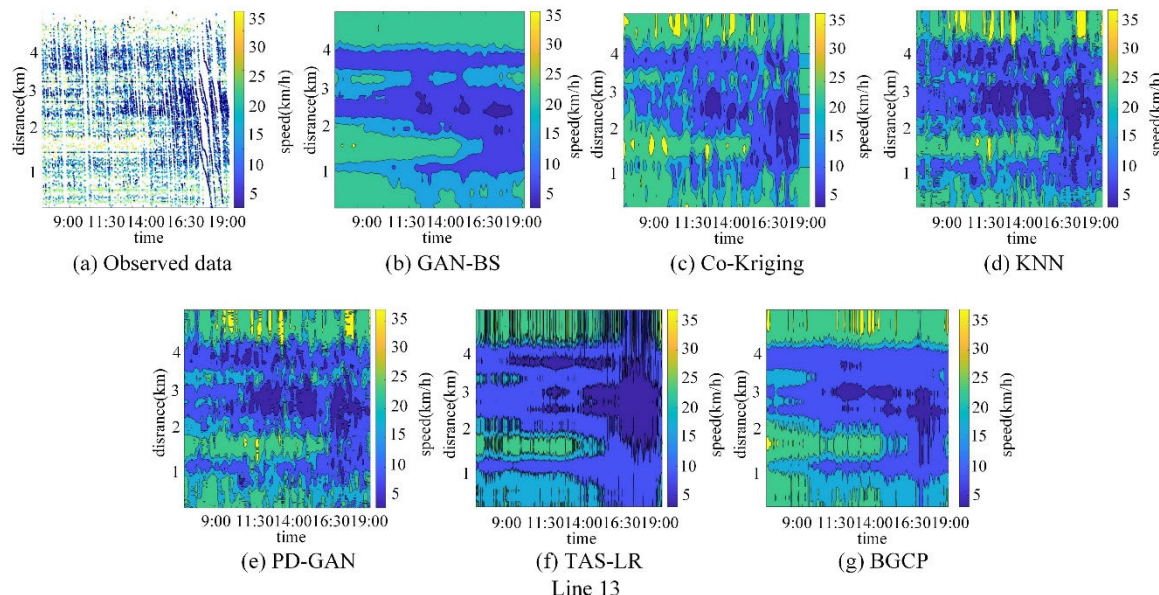


FIGURE 8. Comparison of traffic state heat maps of different methods (Line 13).

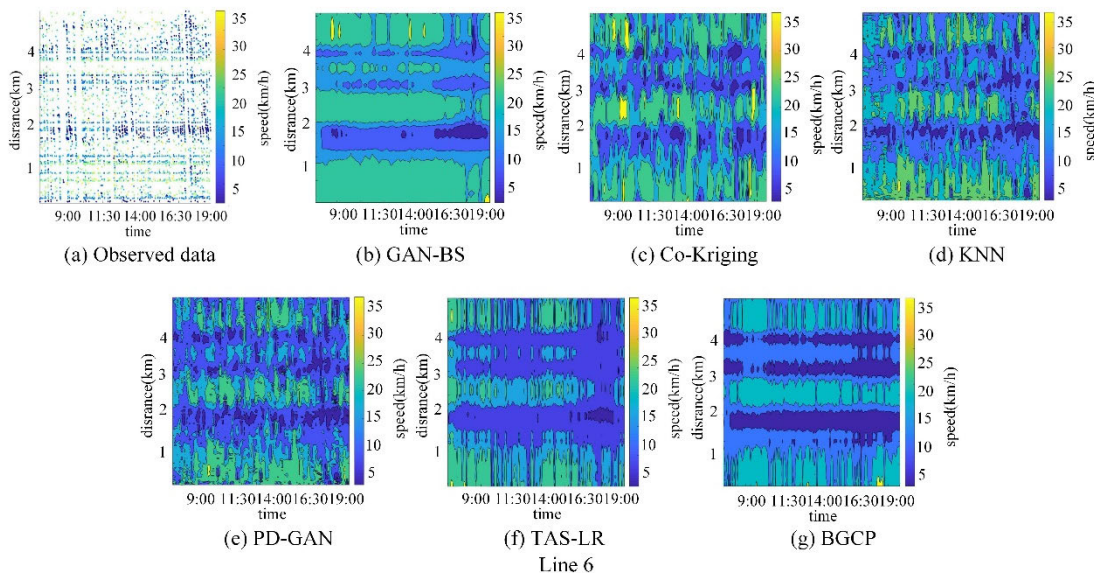


FIGURE 9. Comparison of traffic state heat maps of different methods (Line 6).

fragmented and discontinuous. But the full pipeline method can generate a continuous state from the noisy data because of the ability to fill reasonable data and eliminate noise. Besides, we compared the error indicators of GAN_BS in Table 2 and Table 3 and found that the traffic state estimation of Line 13 and Line 6 are both accurate. This model can estimate the traffic state of exclusive bus lanes and non-exclusive bus lanes well.

Furthermore, to observe the performance of GAN_BS in estimating traffic state and distinguishing the edge of traffic patterns, we have chosen the traffic state of the two bus lines at three different times and compared them with the estimated

speed (as shown in Fig. 6 (1) (2) (3) and Fig. 7 (1) (2) (3)). GAN_BS can reconstruct the traffic state from sparse data points. Because of the smoothing method, it tends to ignore the interference of noise and generate a continuous state.

2) COMPARISON RESULTS WITH OTHER METHODS

It can be observed from Table 2 and Table 3 that GAN_BS achieves better performance than the compared methods in terms of all evaluation metrics. More specifically, we can see the estimated results of each method under the data missing rate of 50% on January 9th from Fig. 8 and Fig. 9. For the Co-kriging interpolation method, it tends to fit a smooth state

TABLE 4. Comparison of performances results of line 13 with other estimation methods under different traffic patterns.

Traffic Scenarios	Statistics	Co-kriging	KNN	SVD	BGCP	PD-GAN	BS	GAN	GAN_BS
Free Flow At 06:30	RMSE	10.52	8.75	8.09	6.47	10.17	5.86	9.63	3.80
	MAE	10.32	6.98	7.94	5.13	8.07	4.65	7.49	2.63
	NMSE	0.21	0.14	0.12	0.08	0.19	0.06	0.17	0.03
Moring Peak At 09:30	RMSE	9.78	9.72	8.20	9.47	7.74	7.09	6.89	5.98
	MAE	9.55	7.41	8.00	7.38	6.07	5.90	5.56	4.76
	NMSE	0.25	0.24	0.17	0.23	0.15	0.14	0.12	0.08
Evening Peak At 18:30	RMSE	10.04	7.90	8.04	6.31	10.02	6.62	7.25	4.80
	MAE	9.90	7.18	7.99	5.08	8.39	6.07	6.44	4.04
	NMSE	0.61	0.97	0.12	0.70	0.79	0.77	0.94	0.36

TABLE 5. Comparison of performances results of line 6 with other estimation methods under different traffic patterns.

Traffic Scenarios	Statistics	Co-kriging	KNN	SVD	BGCP	PD-GAN	BS	GAN	GAN_BS
Free Flow At 06:30	RMSE	10.00	8.43	8.20	5.18	8.54	3.68	7.40	3.38
	MAE	9.95	8.28	8.06	4.27	6.93	2.42	5.95	2.41
	NMSE	0.34	0.34	0.22	0.09	0.25	0.05	0.18	0.04
Moring Peak At 09:30	RMSE	9.80	6.57	8.02	6.45	9.11	5.27	7.15	4.92
	MAE	9.72	6.56	7.92	4.83	7.19	3.82	5.19	3.70
	NMSE	0.22	0.23	0.14	0.09	0.19	0.06	0.11	0.05
Evening Peak At 18:30	RMSE	10.37	11.29	8.03	6.23	11.81	6.30	10.64	5.05
	MAE	10.28	10.17	7.85	5.05	9.19	5.29	8.62	3.86
	NMSE	0.44	0.42	0.25	0.16	0.64	0.17	0.46	0.11

surface while ignoring the intermittent effects of bottlenecks such as road intersections and bus stops. The result of KNN is similar to the output of GAN. It is fragmented because the KNN looks for similar values in the historical data to fill in, and the original data set is noisy. On average GAN_BS have relatively 1.90 lower RMSE, 2.57 lower MAE, 0.08 lower NMSE than the two tensor decomposition methods, including TAS_LR and BGCP in two test data sets. It is easy to see that the tensor decomposition method can eliminate noise while filling the traffic state. But it is easy to overestimate the congestion range at the temporal-spatial location (16 : 15, 3000km) of Line 13. Last, we use the PD-GAN test whether the convolutional network can filter noise better than the fully connected layer. The result shows that PD-GAN still cannot directly produce a smoothing traffic state of the bus lane that conforms to the traffic pattern.

Moreover, as the aggregation scale of data in time shrinks, the missing rate of data increases, and the calculation accuracy of GAN_BS is generally declining. The acceptable accuracy means that GAN_BS is effective in estimating the traffic state when the missing rate of the data ranges from 45% to 80%. However, the accuracy of Co-kriging and TAS_LR on Line 6 decreases with the increase of the aggregation time scale. It may be because the boundaries between different states are getting closer as the aggregation scale increases. And these two methods are less capable of distinguishing

the boundaries between different traffic patterns. On the Line 13 test set, the accuracy of estimation methods is worse than other cases when the aggregation scale is 3 minutes and 10 minutes. During the experiment, it is found that the noise interference is the most obvious in the original data when the aggregation scale is 3 minutes. Therefore, the performance of all test methods has declined. However, the methods with noise immunity, such as SVD, BGCP, and GAN_BS perform relatively better. And the assumption that the traffic state remains stable within 10 minutes is weak for the bus lane of Line 13. Therefore, the test methods will have some distortions in restoring the traffic state patterns.

3) PERFORMANCE COMPARISON UNDER DIFFERENT TRAFFIC PATTERNS

The observed and estimated values of the corresponding traffic state are shown in Table 4 and Table 5. The accuracy of these methods under free-flow conditions is better than under congestion conditions. And we can see that GAN_BS can output accurate estimation results under different traffic conditions. Fig. 10 compares the estimated speed from GAN_BS and observed speed from the bus as scatter diagrams. Generally speaking, the estimated speed is likely to be higher than the observed speed in the congested state with speed slower than 20km/h. In the free flow state, the estimated speed is

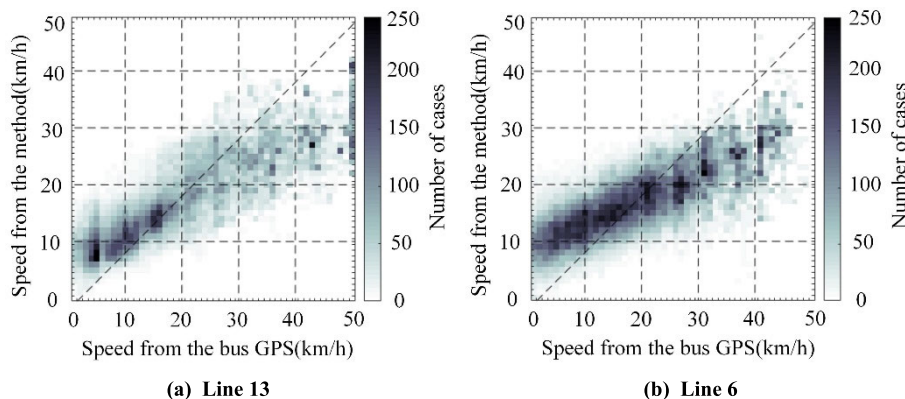


FIGURE 10. Scatter diagrams of estimated speed from GAN_BS vs observed speed from bus GPS.

lower than the observed speed, and the error distribution is more scattered.

V. CONCLUSION

The present study proposed a method named GAN_BS for estimating the traffic state of the bus line. This method first proposes an improved generative adversarial network to generate the bus data under sparse sampling. Then, a traffic adaptive bilateral smoothing method is proposed to reconstruct the accurate traffic state pattern from discrete speed. It can be seen through a series of numerical experiments that GAN_BS can generate accurate traffic state under sparse sampling. In particular, it outperformed other traffic estimation methods in terms of noise elimination and traffic pattern boundary maintenance. GAN_BS also behaved quite robustly with respect to the sparse data whose missing rate ranges from 45% to 80%. GAN_BS can not only impute the sparse data, but also accurately estimate the traffic state. However, parameters such as characteristic wave speed in BS are fixed. Future work should improve the model in dynamically adjusting parameters to adapt to different traffic patterns.

REFERENCES

- [1] B. Bae, H. Kim, H. Lim, Y. Liu, L. D. Han, and P. B. Freeze, "Missing data imputation for traffic flow speed using spatio-temporal cokriging," *Transp. Res. C, Emerg. Technol.*, vol. 88, pp. 124–139, Mar. 2018.
- [2] M. Treiber, A. Kesting, and R. E. Wilson, "Reconstructing the traffic state by fusion of heterogeneous data," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 26, no. 6, pp. 408–419, Aug. 2011.
- [3] K. Zhang, Z. He, L. Zheng, L. Zhao, and L. Wu, "A generative adversarial network for travel times imputation using trajectory data," *Comput.-Aided Civil Infrastruct. Eng.*, early access, Jun. 2020, doi: [10.1111/mice.12595](https://doi.org/10.1111/mice.12595).
- [4] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1762–1771, Jun. 2016.
- [5] X. Chen, Z. He, and L. Sun, "A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 73–84, Jan. 2019.
- [6] Q. Zhaowei, L. Haitao, L. Zhihui, and Z. Tao, "Short-term traffic flow forecasting method with M-B-LSTM hybrid network," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 29, 2020, doi: [10.1109/TITS.2020.3009725](https://doi.org/10.1109/TITS.2020.3009725).
- [7] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.
- [8] L. Han, K. Zheng, L. Zhao, X. Wang, and H. Wen, "Content-aware traffic data completion in ITS based on generative adversarial nets," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11950–11962, Oct. 2020.
- [9] D. Xu, C. Wei, P. Peng, Q. Xuan, and H. Guo, "GE-GAN: A novel deep learning framework for road traffic state estimation," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102635.
- [10] B. Yu, H. Wang, W. Shan, and B. Yao, "Prediction of bus travel time using random forests based on near neighbors," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 4, pp. 333–350, Apr. 2018.
- [11] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Syst. Appl.*, vol. 120, pp. 426–435, Apr. 2019.
- [12] H. Yu, Z. Wu, D. Chen, and X. Ma, "Probabilistic prediction of bus headway using relevance vector machine regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1772–1781, Jul. 2017.
- [13] B. Yu, W. H. K. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop with multiple routes," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1157–1170, Dec. 2011.
- [14] G. Chen, X. Yang, J. An, and D. Zhang, "Bus-arrival-time prediction models: Link-based and section-based," *J. Transp. Eng.*, vol. 138, no. 1, pp. 60–66, Jan. 2012.
- [15] N. Bekiaris-Liberis, C. Roncoli, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3484–3497, Dec. 2016.
- [16] Y. Yuan, J. W. C. van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.
- [17] C. P. I. J. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen, "Localized extended Kalman filter for scalable real-time traffic state estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 385–394, Mar. 2012.
- [18] A. Duret and Y. Yuan, "Traffic state estimation based on Eulerian and lagrangian observations in a mesoscopic modeling framework," *Transp. Res. B, Methodol.*, vol. 101, pp. 51–71, Jul. 2017.
- [19] X. Chen, S. Zhang, L. Li, and L. Li, "Adaptive rolling smoothing with heterogeneous data for traffic state estimation and prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1247–1258, Apr. 2019.
- [20] F. Rempe, P. Franek, U. Fastenrath, and K. Bogenberger, "A phase-based smoothing method for accurate traffic speed estimation with floating car data," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 644–663, Dec. 2017.
- [21] B. L. Smith, W. T. Scherer, and J. H. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1836, no. 1, pp. 132–142, Jan. 2003.
- [22] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transp. Res. C, Emerg. Technol.*, vol. 12, no. 2, pp. 139–166, Apr. 2004.
- [23] J. Zhao, Y. Nie, S. Ni, and X. Sun, "Traffic data imputation and prediction: An efficient realization of deep learning," *IEEE Access*, vol. 8, pp. 46713–46722, Mar. 2020.

- [24] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. C, Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102674.
- [25] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, "Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 70–81, Mar. 2019.
- [26] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 108–120, Sep. 2013.
- [27] L. Qu, J. Hu, L. Li, and Y. Zhang, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.
- [28] C. Gong and Y. Zhang, "Urban traffic data imputation with detrending and tensor decomposition," *IEEE Access*, vol. 8, pp. 11124–11137, Jan. 2020.
- [29] Q. Li, H. Tan, Y. Wu, L. Ye, and F. Ding, "Traffic flow prediction with missing data imputed by tensor completion methods," *IEEE Access*, vol. 8, pp. 63188–63201, Mar. 2020.
- [30] G. Pastor, "A low-rank tensor model for imputation of missing vehicular traffic volume," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8934–8938, Sep. 2018.
- [31] J. Zhao, Y. Gao, J. Tang, L. Zhu, and J. Ma, "Highway travel time prediction using sparse tensor completion tactics and-nearest neighbor pattern matching method," *J. Adv. Transp.*, vol. 2018, pp. 1–16, Mar. 2018.
- [32] O. Benkraouda, B. T. Thodi, H. Yeo, M. Menendez, and S. E. Jabari, "Traffic data imputation using deep convolutional neural networks," *IEEE Access*, vol. 8, pp. 104740–104752, Jun. 2020.
- [33] Y. Zhuang, R. Ke, and Y. Wang, "Innovative method for traffic data imputation based on convolutional neural network," *IET Intell. Transp. Syst.*, vol. 13, no. 4, pp. 605–613, Apr. 2019.
- [34] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102622.
- [35] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.
- [36] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, "Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105592.
- [37] H. Li, M. Li, X. Lin, F. He, and Y. Wang, "A spatiotemporal approach for traffic data imputation with complicated missing patterns," *Transp. Res. C, Emerg. Technol.*, vol. 119, Oct. 2020, Art. no. 102730.
- [38] J. Tang, X. Zhang, W. Yin, Y. Zou, and Y. Wang, "Missing data imputation for traffic flow based on combination of fuzzy neural network and rough set theory," *J. Intell. Transp. Syst.*, vol. 2020, pp. 1–16, Jan. 2020, doi: [10.1080/15472450.2020.1713772](https://doi.org/10.1080/15472450.2020.1713772).
- [39] K. Zhang, N. Jia, L. Zheng, and Z. Liu, "A novel generative adversarial network for estimation of trip travel time distribution with trajectory data," *Transp. Res. C, Emerg. Technol.*, vol. 108, pp. 223–244, Nov. 2019.
- [40] Y. Liang, Z. Cui, Y. Tian, H. Chen, and Y. Wang, "A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 45, pp. 87–105, Dec. 2018.
- [41] J. Li, J. Jia, and D. Xu, "Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9159–9163.
- [42] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: A point cloud upsampling adversarial network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–10.
- [43] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*. [Online]. Available: <http://arxiv.org/abs/1703.10717>
- [44] K. P. Diederik and B. Jimmy, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1–15.
- [45] Y. Wang, Y. Zhang, X. Piao, H. Liu, and K. Zhang, "Traffic data reconstruction via adaptive spatial-temporal correlations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1531–1543, Apr. 2019.



XIANMIN SONG received the B.S. and Ph.D. degrees in traffic engineering from Jilin University, Changchun, China, in 2005 and 2008, respectively. She is currently a Professor with the School of Transportation, Jilin University. Her research interests include intelligent traffic control, traffic flow theory, and traffic big data.



JING TIAN received the B.S. degree in traffic engineering from Jilin University, Changchun, China, in 2020, where she is currently pursuing the Ph.D. degree with the School of Transportation. Her research interests include traffic state prediction and deep learning.



PENGFEE TAO received the B.S. degree in computer science and technology and the M.S. and Ph.D. degrees in traffic engineering from Jilin University, China. He is currently an Associate Professor with the School of Transportation, Jilin University. His research interests include driving behavior analysis and autonomous vehicles.



HAITAO LI received the B.S. degree in traffic engineering from Jilin University, Changchun, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Transportation. His research interests include pattern recognition and traffic information processing.



CONG WU received the bachelor's degree in transportation (planning and management) from Jilin University, Changchun, China, in 2018, where she is currently pursuing the master's degree with the School of Transportation. Her research interests include traffic flow theory and driving behavior.