Neural Computing and Applications (2023) 35:6545–6567
https://doi.org/10.1007/s00521-022-07961-4

ORIGINAL ARTICLE

Multistate time series imputation using generative adversarial network with applications to traffic data

Haitao Li[1] · Qian Cao[1] · Qiaowen Bai[1] · Zhihui Li[1] · Hongyu Hu[2]

Received: 23 November 2021 / Accepted: 17 October 2022 / Published online: 23 November 2022

Abstract
Time series missing data is a pervasive problem in many fields, especially in intelligent transportation system, which hinders the application of timing analysis methods and the fine adjustment of control strategies. The prevalent imputation approaches reconstruct missing data with a high accuracy by exploiting a precise distribution model. But the multistate characteristic of time series data and the uncertainty of imputation process increase the difficulty of modeling temporal data distribution and reduce the imputation performance. In this paper, a novel time series generative adversarial imputation network (TGAIN) model is proposed to deal with time series data missing problem. The model combines the advantages of GAN's data distribution modeling and multiple imputation's uncertainty handling. Specifically, the TGAIN network is designed and adversarial trained to learn the multistate distribution of missing time series data. Through the conditional vector constraint and adversarial imputation process, the latent distribution for each missing position under different states can be effectively estimated based on implicit relationships with partial observation information. Then the corresponding multiple imputation strategy is proposed to deal with the uncertainty of imputation process and it can determine the best fill value from the learned distribution. Furthermore, sufficient experiments have been conducted in two real traffic flow datasets. The comparative results show the proposed TGAIN not only has better ability on time series data distribution modeling and imputation uncertainty handling, but also performs more robustly and stability even with the missing rate increases.

Keywords Generative adversarial network · Multiple imputation · Time series data · Imputation

1 Introduction

Multistate time series data recording time-varying values of variables not only reflects the mode fluctuation trend of each variable, but also implies the coupling relationships among variables. Thus, the analyses of multistate time series data are important in various actual applications, especially in intelligent transportation system (ITS). Most traffic services for smart cities, like traffic signal control [1], traffic congestion forecasting [2] and incident detection [3], have been mining the multistate temporal characteristic of traffic flow for precision management. However, the unavoidable data missing problem caused by hardware malfunction, failure of transmission and data corruption may bring great difficulties to accurate data analysis. The descent of analysis efficiency, more complicated analysis procedure and even bias conclusion owing to the differences between observed and missing data are the typical serious problems caused by data missing [4, 5]. To avoid these problems, it is necessary to properly process the missing values (MVs) in multistate time series data.

Generally, missing values processing methods can be divided into three categories. The first category is simply to delete the samples with MVs. However, this case deletion tends to lose partial useful information and may distort sample distribution especially in the limited samples or high missing rate situation [6]. The second category is the

✉ Qiaowen Bai
victorbty@foxmail.com

Haitao Li
lihait@jlu.edu.cn

[1] College of Transportation, Jilin University, Changchun 130022, People's Republic of China

[2] College of Automotive Engineering, Jilin University, Changchun 130022, People's Republic of China

single imputation which attempts to model the data missing process by available partial data information, and estimates a reasonable value by various imputation models. This method mainly includes regression imputation (K-nearest neighbor (KNN) [7, 8], local least square (LLS) [9, 10], nearest neighbor regression (NNR) [11], support vector machine (SVM) [12, 13]), probabilistic model imputation (probabilistic principal component analysis (PPCA) [14, 15], singular value decomposition (SVD) [16, 17]), matrix completion imputation (low-rank matrix completion (LRMC) [18, 19] and self-representation (SRSP) [20, 21]). However, these models share a common problem: They impute a single value for each missing position and then treat the imputed data the same as the observed data in subsequent analysis. In fact, the MVs obey an implicit distribution determined by available observed information, rather than a certain value [22]. The single imputation implicitly assumes that the imputation models and results are perfect, but fails to account for the uncertainty of missing data in the imputation process. That can be overcome by replacing each missing value with several slightly different imputed values, and this kind of imputation framework is the third category—multiple imputation which is the most effective framework for missing data analysis [23]. The idea of multiple imputation is to model the missing data distribution through multiple filling MVs and evaluate all filling values fitness and give the optimal filling value [24, 25]. However, the performance of multiple imputation models relies on the correct distribution assumption of the selected imputation model [24]. So, the accurate distribution solution is imperative to increase the availability of multiple imputation.

Usually, as time goes by, time series data tends to exhibit regularity and randomness, and show different trends and data distributions under different states and conditions [3, 5]. Taking the traffic flow data of two adjacent detecting points in Fig. 1 as an example, there are significant differences in time series data distribution and trend of traffic flow parameters in different time periods, and these differences are still obvious even for adjacent points at the same periods. This is traffic flow multistate distribution characteristic, will further increase the difficulty of data distribution solution in the missing data imputation task. Existing imputation methods adopt fixed preset statistic distribution or simple solution by superficial model and they cannot accurately realize the adaptive modeling of multistate distribution for time series data. Therefore, it is necessary to explore a more effective distribution solution model to improve imputation performance in time series data missing problem.

Recently, generative adversarial network (GAN) gives us a better choice in modeling data distribution as a latest generative model. More specifically, by training with original data, GAN can capture the distribution of original data by making the distribution of generated samples approximate original data distribution [26, 27]. It has been successfully applied to image completion [28] and sentence generation [29], but the limitation of directly using common GAN for MVs imputation is the network requires a complete dataset for training which is impossible for incomplete time series dataset. To deal with incomplete input data in imputation task, J. Yoon et. al. proposed a novel generative adversarial imputation network (GAIN) to learn missing part distribution by adversarial learning between imputation and discrimination [30]. The subsequent models [5, 31–33] pay more attention to the multivariate characteristic of data by designing a specific generator like multichannel or feature convolution. But the multistate distribution characteristic of time series data will increase the difficulty of distribution learning process, the existing models lack corresponding targeted structural design and may decrease the distribution learning performance. Meanwhile, GAN obtains input variables by random generation and computes the corresponding results, this generating way brings the variety of generated results but also increases their uncertainty [26, 30]. For MVs imputation task, the uncertainty of generated results will affect the imputation accuracy. Up to now, the effective solution of multistate distribution of time series data and the uncertainty handling of imputation process are still challenging in time series imputation task.

To deal with these problems, a novel imputation network framework combining with the GAN's modeling data distribution ability and the uncertainty handling ability of multiple imputation is proposed in this paper. The imputation framework consists two stages. In the first stage, a time series generative adversarial imputation network (TGAIN) is constructed to overcome the hardship of modeling the multistate distribution for time series missing values. TGAIN utilizes the conditional information and sequence generation structure to direct the data imputation process. Through large sample adversarial training by incomplete dataset, the well-trained TGAIN model can learn the distribution of missing data under different states, the implicit relationships between variables and the temporal information of time series data. In the second stage, to deal with uncertainty in imputation and determine the 'best' filling value, multiple imputation strategy is adopted in this stage. Multiple-input 'noise' of TGAIN's generator is utilized to generate multiple filling values which obey the learned distribution, TGAIN's discriminator evaluates the imputation fitness of each of the filling values, and a max-pooling structure is designed to overall determine the final best filling value. In experiments, the compared experiments on two real-world traffic datasets show the proposed method has better ability in dealing with the

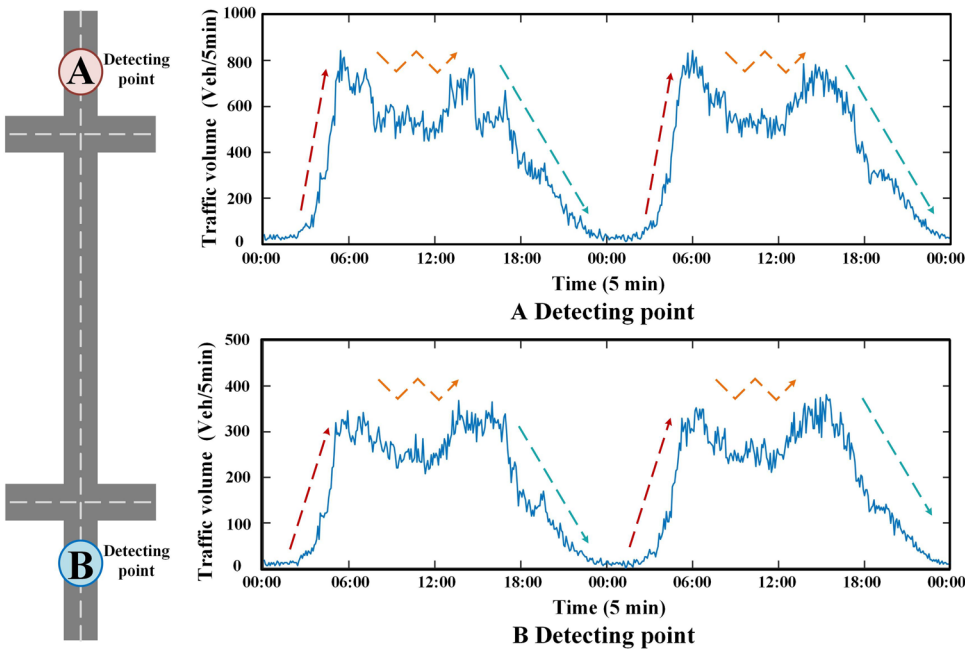

Fig. 1 Multistate time series of traffic sensor data

uncertainty of imputation and the multistate time series distribution solution. The main contribution of this paper can be summarized as follows:

1. To deal with the distribution solution for multistate time series data and the uncertainty of missing values imputation, a novel TGAIN network imputation framework is constructed combined the generative adversarial network and multiple imputation. It is a new multiple imputation network for multistate time series data.
2. To realize the accuracy distribution solution for each state of time series data, TGAIN network designs the condition information and sequence generation structure to direct the generative adversarial imputation process. The well-trained TGAIN can realize multistate distribution learning and utilize the latent temporal information among datasets.
3. To better capture the uncertainty of imputation process, a multiple imputation strategy based on TGAIN is designed. Multiple-input 'noise' of TGAIN's generator is utilized to generate multiple fill values which match the learned distribution, and by a max-pooling structure to overall determine the best filling value.
4. The TGAIN imputation model outperforms several state-of-the-art methods in various missing patterns, even without complete observations for the model training. Even in the case of high missing rate, the imputation performance still remains excellent.

The rest of this paper is organized as follows. In Sect. 2, we introduce the missing data imputation related works. In Sect. 3, we present the TGAIN imputation framework and algorithm process. Experiments and comparison results with several state-of-the-art imputation methods are shown in Sect. 4. Section 5 makes a conclusion and discusses the future work.

2 Related work

Data missing is a common and confusing problem in actual applications, especially for multistate time series data. Over the past decades, a number of advanced methods have been proposed for data imputation and demonstrated significantly improved imputation performance by exploiting the data correlation and the implicit data distribution [20, 34]. From the perspective of imputation structure, the imputation methods can be divided into two categories: single imputation models and multiple imputation models.

2.1 Single imputation models

The single imputation models utilize the data correlation between observed values and missing values to give a reasonable value to replace the missing part. This category also can be roughly divided into following three classes.

2.1.1 Regression-based imputation

This class methods attempt to model the inherent relationship between MVs and observed values by means of regression techniques, such as KNN [7, 8], LLS [9, 10], NNR [11] and SVM [12, 13]. Despite some differences in terms of specific regression models, these methods share a

common motivation, which is aim to use observed values to predict the missing partial. For example, KNN-based imputation method [8] takes advantage of similarity measure to find several samples from dataset which most similar to the MVs samples, and MVs can be estimated as the weighted average of those selected samples. Instead of weighted average in KNN, LLS [10] imputation describes the relation between the MVs and observed values by least square regression, allowing more flexibility than weighted average. Though these methods utilize the local relation of data to recover the MVs individually, they all failed to consider the consistency across the sample space and the performance impact of multistate distribution. Even worse, if a whole series of consecutive data is lost, the imputation performance of these methods will decrease rapidly.

2.1.2 Probability-based imputation

To this kind of methods, the complete data is supposed to follow a probabilistic distribution with specific form but the distribution parameters are unknown, e.g., mixed Gaussian model. Based on the observed partial data, both the distribution parameters and missing data can be iteratively estimated. The Markov chain Monte Carlo (MCMC) [35] and probabilistic principal component analysis (PPCA) [14, 15] are the representative methods and have shown promising results in traffic data imputation. The major disadvantage of these methods is the imputed performance heavily depends on the prior assumption about data distribution, which is unknown in practice. Especially, due to the multistate characteristic of time series data, it may be improper to postulate a uniformed distribution for time series in different state.

2.1.3 Matrix completion-based imputation

This class methods organize whole or partial data into a matrix, and the missing values are recovered based on specific assumption about data matrix. Low-rank matrix completion (LRMC) [18, 19] assumes that the whole data matrix has a global low-rank structure, and MVs can be recovered through rank minimization on the whole matrix. Recently, matrix self-representation (SRSp) [20, 21] has been proposed which assumed each sample can be well represented by linear combination of other samples. Despite encouraging results, the assumption of these methods led to disregard the local relation of data.

Above three kinds of single imputation methods have their own unique theoretical value, but all single imputation methods cannot resolve the uncertainty in imputation process effectively. This problem limits the imputation effectiveness of the algorithm itself.

2.2 Multiple imputation models

Multiple imputation (MI) is a general framework that incorporates the uncertainty into the imputation process. MI is comprised of three stages: imputation stage, in which there is a need to calculate the dataset statistic parameters and distribution, and variability is put into the imputed values to create multiple complete datasets; analysis stage, in which each of the complete datasets is analyzed using a complete data technique; and the last stage, in which the results from the analysis are combined in order to yield a final result and this stage combines the uncertainty in the data and the uncertainty due to missing values. The original multiple imputation [22], Bayesian MI [36] and deep learning [24] make certain expansion on the multiple imputation methods, and even some improved algorithms by single methods also draw on the idea of multiple fillings [37, 38]. The core problem is how to effectively model the data distribution of missing data location. Even though existing methods have been tried by fusing the results of different algorithms or multiple results of a single algorithm [37, 38], the accurate distribution solution of missing data based on partial observation information is still a hard problem, especially for multistate time series data.

In all, the main foundation of missing data imputation utilizes the latent data correlation and implicit data distribution. But how to model the correct distribution of data and handle the uncertainty simultaneously are the technical difficulties in improving the effect of multiple imputation.

3 Method

To deal with the uncertainty of imputation and the multistate distribution solution in time series data imputation task, a novel imputation framework is designed and described in Fig. 2. From Fig. 2, the framework divides into two stages: The first stage is the distribution solution for time series missing data. Inspired by the advantage of GAN data distribution modeling [26], we proposed a novel TGAIN architecture to adversarial learning missing data distribution by incomplete data samples. Considering the multistate characteristic and utilization of temporal correlation, the conditional vector and seq-to-seq temporal generator are introduced into the TGAIN to direct the impute process for the missing values. By large sample training, the trained TGAIN can learn the data distribution under different states, the implicit relationships between observation and the temporal information of data. The second stage is multiple imputation by TGAIN to determine the best imputed values. Multiple ‘noises’ are input into TGAIN’s generator to generate multiple fill values

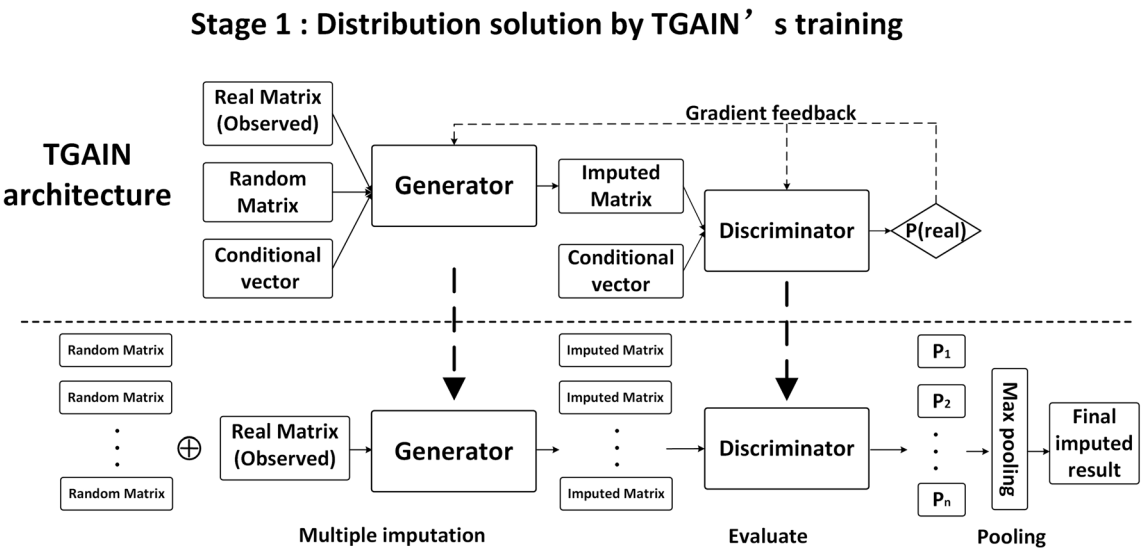

Fig. 2 Framework of time series generative adversarial imputation network (TGAIN)

which obey the learned distribution, and the discriminator gives the probability that each imputed value is close to the original incomplete time series. A max-pooling structure is designed to select the maximum probability result and gives the final reasonable fill values for each missing position.

Formally, given a collection of multivariate time series data with d dimensions $X = [x_{t_0}, ..., x_{t_i}, ..., x_{t_{n-1}}] \in R^{d \times N}$, where $x_{t_i} \in R^d$ denotes the t_i th observation of X and $x_{t_i}^j$ is the j th feature of x_{t_i}.

It is worth noticing that in missing data imputation problem, the observation time series X is incomplete, let $\tilde{X}$ denotes the uncompleted time series matrix. The mask matrix $M \in R^{d \times N}$ is introduced to indicate whether the values of X exist or not, i.e., if $x_{t_i}^j$ exists, $M_{t_i}^j = 1$; otherwise, $M_{t_i}^j = 0$. For example:

$$\tilde{\mathrm{X}} = \begin{bmatrix} x_{t_1}^1 & * & x_{t_3}^1 & * \\ * & * & x_{t_3}^2 & x_{t_4}^2 \\ x_{t_1}^3 & x_{t_2}^3 & * & x_{t_4}^3 \end{bmatrix} \quad \mathrm{M} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

The architecture of TGAIN and multiple imputation process will be described in detail in the following parts.

3.1 TGAIN architecture

In order to replace missing values in multimodal time series dataset, a time series generative imputation adversarial network (TGAIN) is constructed to unsupervised learning the distribution of original time series dataset under various conditions. In this custom GAN architecture as shown in Fig. 3, some condition information C corresponds to $\tilde{X}$ as the additional input vector to direct the data imputing process. The generator utilizes a random noisy matrix and condition vector to generate the fake imputed values, and the discriminator is trained to distinguish the fake imputed parts and real observation parts in certain conditions. With the iteration of adversarial imputation training, it will achieve a Nash equilibrium between the imputation ability of the generator and the discernment ability of the discriminator. Finally, the generator learns a mapping function $G(z)$ that tries to map the random noise vector z to a realistic time series.

3.1.1 Generator

The generator G takes $\tilde{X}$, M, random matrix Z and condition vector C as inputs and output is a complete matrix $\overline{X}$. Here, the Z must be independent of all other variables to avoid being influenced by variable uncertainty. The generator calculation is expressed as follows:

$$\overline{X} = G(\tilde{X} \odot M + (1 - M) \odot Z, C) \tag{1}$$

$$\hat{X} = M \odot \tilde{\mathrm{X}} + (1 - M) \odot \overline{\mathrm{X}} \tag{2}$$

where $\odot$ denotes element-wise multiplication. It notes that the directly generated matrix $\overline{X}$ is completely a fake matrix whether the component was observed or missing. So $\hat{X}$ is the completed imputed data matrix by replacing missing component * in $\tilde{X}$ with the corresponding value of $\overline{X}$ as Eq. (2).

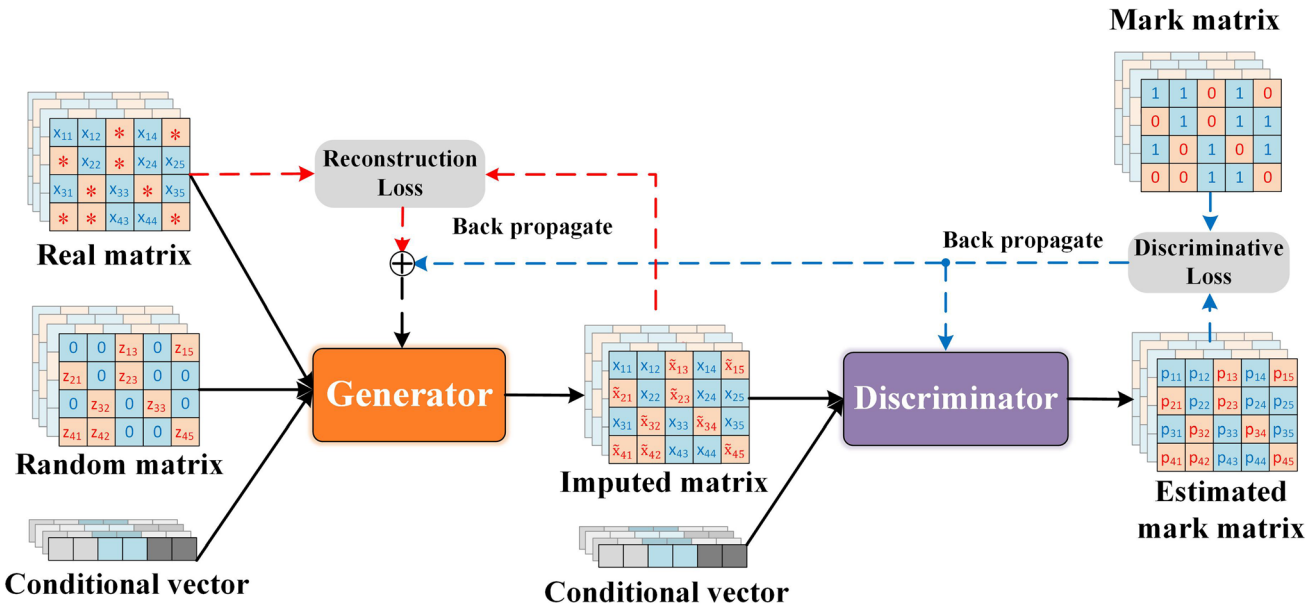

Fig. 3 TGAIN network architecture

To effectively utilize the latent data relationships (time series tendency and the different feature correlation) and the guidance of conditional information in the matrix generation process, a novel generator is constructed as shown in Fig. 4. The TGAIN generator adopts sequence generation architecture: First, the incomplete time series matrix is combined with the random matrix by $\tilde{X} \odot M + (1 - M) \odot Z$ as the input of generator and extracts spatial feature between different variables by multilayer fully connected encoder; then, spatial feature vectors and conditional vector will be fused by concat function to guide subsequent calculations. The LSTM layer is adopted to capture and utilize the temporal relation, and the final feature vectors are input a multilayer fully connected decoder to generate the imputed matrix in chronological order.

The whole generation processes not only utilized the spatial–temporal information of different variables, but also added the conditional information, which will conducive to the accurate estimation of missing data.

3.1.2 Discriminator

In TGAIN framework, the discriminator D is introduced as an adversary to train G. Due to the imputation task different, unlike in a standard GAN where the output of generator is entire real or fake, in this setting the output is composed of real and fake components. So, the discriminator of TGAIN attempts to distinguish which components are real (observed) or fake (imputed), rather than identify whether an entire vector is real or fake. The results of discriminator are equivalent to predicting the mask vector M which is predetermined by the dataset. It notes that the discrimination process also needs the guidance of correspondent conditional information which could satisfied the needs of data multistate distribution.

Formally, the discriminator is a function $D : \hat{M} = D(\hat{X}, C)$, with the i th component of output $\hat{M}$ corresponding to the probability that the i th component of $\hat{x}$ was observed under condition C.

3.1.3 Conditional vector

Considering the difference on latent relation and distribution of multimodal time series dataset, the conditional vector is constructed to direct the missing value imputing process. It makes the whole network extended as a conditional model because the generator and discriminator are conditioned on same extra information C. The C could be any kind of auxiliary information for different kinds of data imputation task, such as class labels or data from other modalities. It notes that the condition information C should be potentially related to the multistate characteristic of time series data, so that C could direct data generation and imputation process under different data distribution situation. Meanwhile, C should be appropriately settings for different tasks. More conditional information may increase the complexity of network learning and need more data under each condition, but less conditional information may lead to the poorly direct modeling performance.

In the task of traffic flow data imputation, the traffic data showed obvious different temporal distribution patterns between workday and non-workday and different time periods on the same day. So, the week label and time label

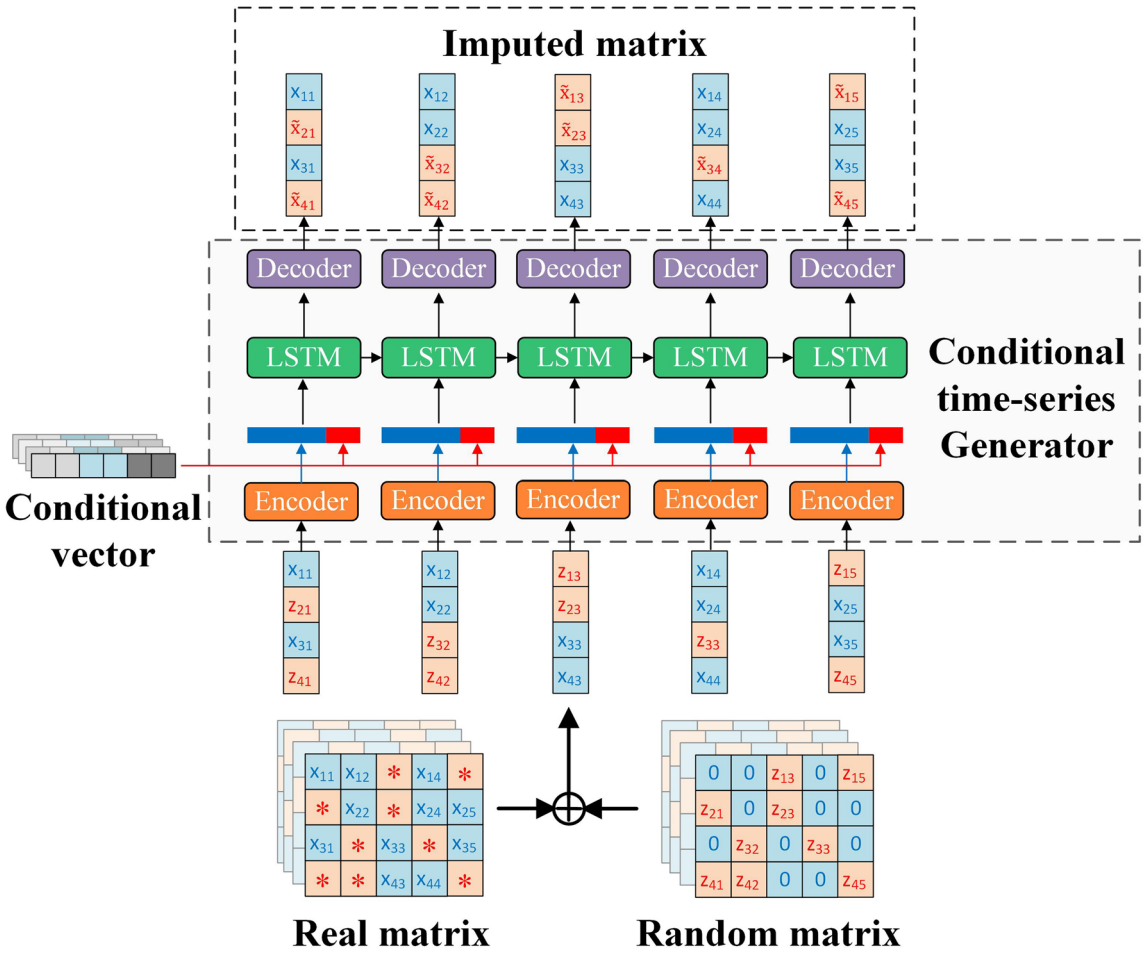

Fig. 4 Conditional time series generator of TGAIN

are selected as conditional vector in this paper, the specific partition is determined by real traffic data distribution. In order to ensure the independence of the labels, the labels are processed by one-hot encoding [39]; that is, each label is given an effective encoding bit.

$$C=[week\ lable\ ,\ \ time\ label] \tag{3}$$

3.1.4 Objective

In TGAIN network, the discriminator D is trained to maximize the probability of correctly predicting M, the generator G is trained to minimize the probability of D to predict M, so we define the objection function as

$$V(D,G) = \mathbb{E}_{\hat{X},M,C}[M^T \log D(\hat{X},C) + (1-M)^T \log(1 - D(\hat{X},C))] \tag{4}$$

where log is element-wise logarithm and dependence on G is through $\hat{X}$.

Then, the objective of TGAIN is a minimax game problem given by

$$\min_G \max_D V(D,G) \tag{5}$$

Writing $\hat{M} = D(\hat{X},C)$, Eqs. (4) and (5) can be rewritten as:

$$\min_G \max_D \mathbb{E}_{\hat{X},M,C}\left[M^T \log(\hat{M}) + (1-M)^T \log(1-\hat{M})\right] \tag{6}$$

3.1.5 Loss

TGAIN attempts to model the latent distribution of missing data by multiple generative adversarial learning processing rather than just the statistical expectation. So, we solve the minimax optimization problem of the network in an iterative manner.

We first optimize the discriminator D with a fixed generator G using mini-batch method in [40]. For each sample in the mini-batch $(\tilde{x}(j), m(j), c(j))$, we draw k_D independent random samples $z(j)$. We define the discriminator loss function to train the discriminator as

Table 1 TGAIN network iterative training pseudo-code

Constant parameter: the Generator and discriminator architecture, hyper-parameter α in [10].

Input: Training dataset $\left\{\left(\tilde{x}(j), m(j), c(j)\right)\right\}_{j=1}^{\text{Train}}$

Output: the trained TGAIN network parameters (G and D)

1: While training loss has not converged **do**

Stage 1: Discriminator optimization--fixed generator parameters.

2: Draw k_D samples from training dataset $\left\{\left(\tilde{x}(j), m(j), c(j)\right)\right\}_{j=1}^{k_D}$

2: Draw k_D i.i.d. samples, $\left\{z(j)\right\}_{j=1}^{k_D}$ of Z.

3: **for** $j = 1, \cdots, k_D$ **do**

4: $\overline{x}(j) \leftarrow G\left(\tilde{x}(j) \odot m(j) + (1 - m(j)) \odot z(j), c(j)\right)$

5: $\hat{x}(j) \leftarrow m(j) \odot \tilde{x}(j) + (1 - m(j)) \odot \overline{x}(j)$

6: $\hat{m}(j) \leftarrow D(\hat{x}(j), c(j))$

7: **end for**

8: Update D using stochastic gradient descent

9: $\nabla_D \sum_{j=1}^{k_D} L_D(\tilde{x}(j), m(j), c(j), z(j))$

Stage 2: Generator optimization--fixed discriminator parameters

10: Draw k_G samples from training dataset $\left\{\left(\tilde{x}(i), m(i), c(i)\right)\right\}_{i=1}^{k_G}$

11: Draw k_G i.i.d. samples, $\left\{z(i)\right\}_{i=1}^{k_G}$ of Z.

12: **for** $i = 1, \cdots, k_G$ **do**

13: $\overline{x}(i) \leftarrow G\left(\tilde{x}(i) \odot m(i) + (1 - m(i)) \odot z(i), c(i)\right)$

14: $\hat{x}(i) \leftarrow m(i) \odot \tilde{x}(i) + (1 - m(i)) \odot \overline{x}(i)$

15: $\hat{m}(i) \leftarrow D(\hat{x}(i), c(i))$

16: **end for**

17: Update G using stochastic gradient descent

18: $\min_G \sum_{i=1}^{k_G} L_M\left(\tilde{x}(i), \overline{x}(i), m(i)\right) + \alpha L_G\left(\tilde{\mathrm{x}}(i), \mathrm{m}(i), \mathrm{c}(i), \mathrm{z}(i)\right)$

19: End while

$$
\begin{aligned}
&L_D(\tilde{X}, M, C, Z) \\
&\quad = \min_D - \sum_{j=1}^{k_D} \left[m(j)^T \log D(\hat{x}(j), c(j)) + (1 - m(j)^T)\right. \\
&\quad \left.\log(1 - D(\hat{x}(j), c(j)))\right]
\end{aligned} \tag{7}
$$

Second, we optimize the generator G using the newly updated discriminator D with mini-batches of size k_G. It notes that G in fact outputs the value for entire data matrix (including values for the components we observed). Therefore, in training G, the loss function should ensure not only the imputed values for missing components ($m_j = 0$) successfully fool the discriminator, but also the values outputted by G for observed components ($m_j = 1$)

are close to those actually observed. This also assures that the representations learned in G suitably capture the information combined in $\tilde{X}$ (Just like an auto-encoder).

To achieve this purpose, a two-part loss function is defined to evaluate the fitness of imputation. The following paragraphs will describe the discriminative loss and masked reconstruction loss in details.

3.1.6 Masked reconstruction loss

The masked reconstruction loss makes sure that the generated samples by G are close enough to the original incomplete time series $\tilde{X}$ in non-missing components. It is defined by masked squared errors between the original sample $\tilde{X}$ and the generated sample $\overline{X}$ by $G(\tilde{X}, Z, C)$.

$$L_M\left(\tilde{X}, \overline{X}, M\right) = \min_G - \sum_{i=1}^{k_G} \|\tilde{x}(i) \odot m(i) - G(\tilde{x}(i), z(i), c(i)) \odot m(i)\| \tag{8}$$

3.1.7 Discriminative loss

The discriminative loss forces the generate sample by G as real as possible in missing components. It stands for the generated sample $G(\tilde{X}, Z, C)$'s degree of authenticity. It is based on the output of the discriminator D in missing components which represents the confidence level of the generated sample being real.

$$L_G(\mathrm{X}, \tilde{M}, C, Z) = \min_G - \sum_{i=1}^{k_G} (1 - m(i)) \log(m(i)) \tag{9}$$

L_G is smaller when $\hat{m}_i$ is closer to 1. It means L_G is smaller when D is less able to identify the imputed values as being imputed (D falsely categorizes them as observed).

As can be seen from their definitions, L_G applies to the missing components ($m_i = 0$) and L_M applies to the observed components ($m_i = 1$). The generator G is then trained to minimize the weighted sum of the two losses as follows:

$$\min_G \sum_{i=1}^{k_G} L_M(\tilde{x}(i), \bar{x}(i), m(i)) + \alpha L_G(\tilde{\mathrm{x}}(i), \mathrm{m}(i), \mathrm{c}(i), \mathrm{z}(i)) \tag{10}$$

where α is a hyper-parameter to balance two parts loss function and affect the final imputation performance.

Table 1 illustrates the TGAIN network training process in the first stage of TGAIN imputation framework. Firstly, we fixed the generator parameters and computer the L_D to direct the discriminator parameters update. Secondly, the discriminator parameters are fixed and we computer the mixed loss $L_M + \alpha L_G$ to direct the generator parameters update. Two steps will be loop iterated until the network loss converges. By adversarial learning process between imputation and discrimination, the imputed false values will gradually close to the latent 'real' values. When the TGAIN network converges, the distribution of imputed values by G is consistent with the latent distribution of missing data. This will be confirmed in the convergence analysis experiment of TGAIN.

3.2 Multiple imputation by TGAIN

Through TGAIN network training, the generator G can learn a mapping function $G(z) = z \mapsto x$ that maps the random noise vector z to the imputation value satisfied latent

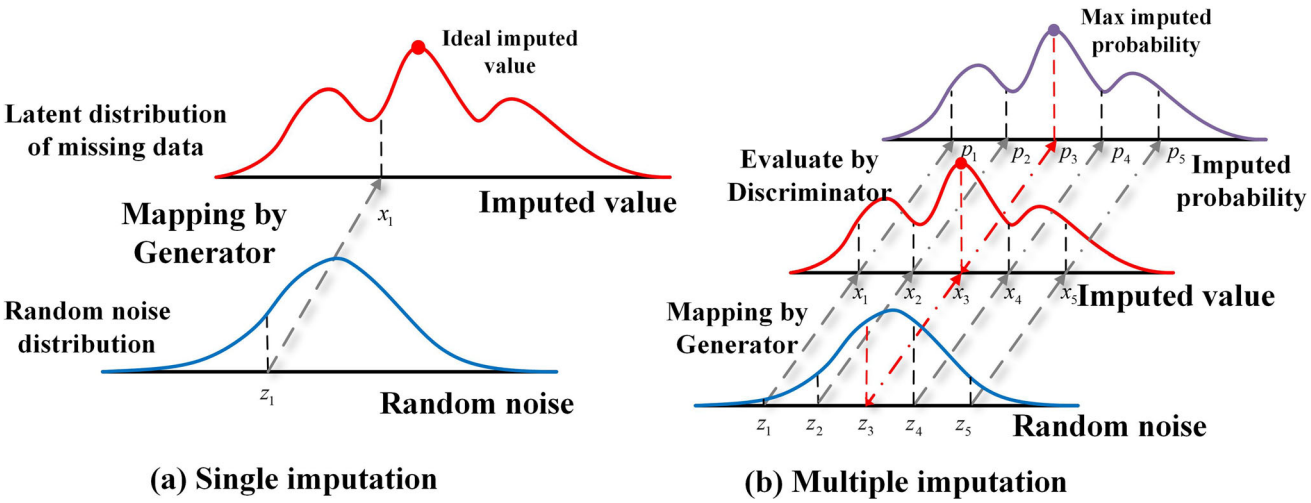

Fig. 5 Schematic diagram of multiple imputation by TGAIN

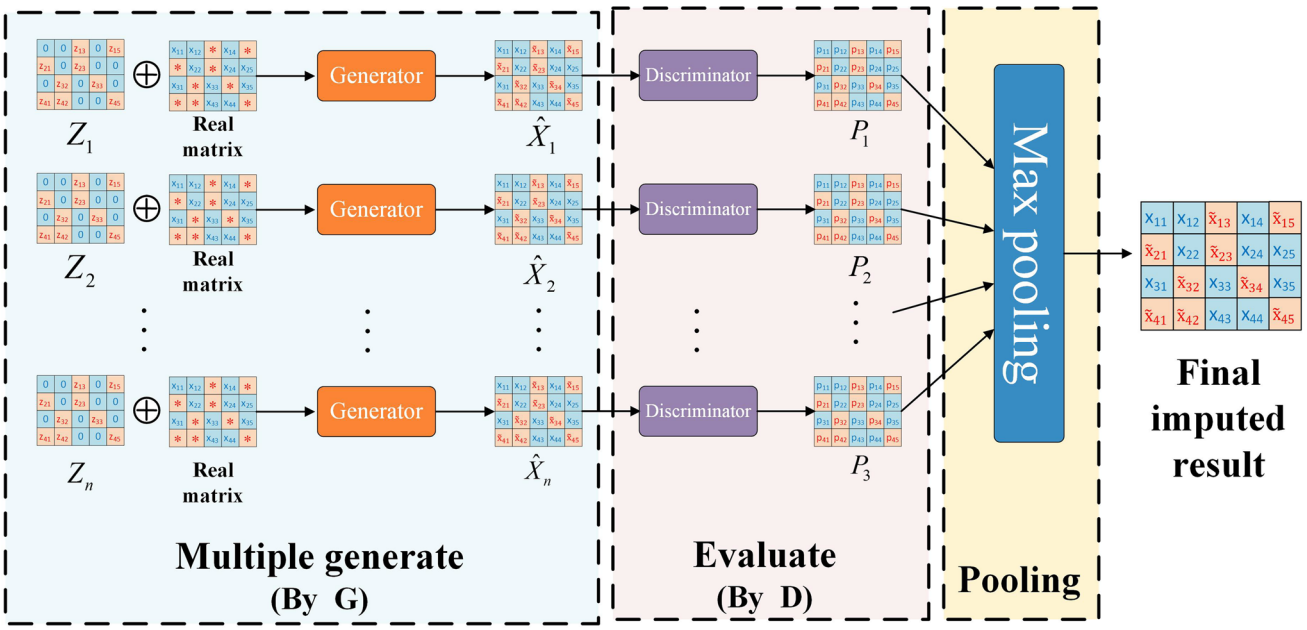

Fig. 6 Multiple imputation stage by TGAIN

Table 2 Multiple imputation pseudo-code by TGAIN

Precondition : The well-trained TGAIN by stage 1. (G and D), the imputation number n.

Input: Testing dataset $\left\{\left(\tilde{x}(j), m(j), c(j)\right)\right\}_{j=1}^{\text{Test}}$

Output: the final imputed matrix.

1: Draw k_M samples from the dataset $\left\{\left(\tilde{x}(k), m(k), c(k)\right)\right\}_{k=1}^{k_M}$

2: For $k = 1, \cdots, k_M$ **do**

3: Draw N i.i.d. samples, $\{z(n)\}_{n=1}^{N}$ of Z.

4: **for** $n = 1, \cdots, N$ do

5: $\bar{x}(n) \leftarrow G\left(\tilde{x}(k) \odot m(k) + (1 - m(k)) \odot z(n), c(k)\right)$

6: $\hat{x}(n) \leftarrow m(k) \odot \tilde{x}(k) + (1 - m(k)) \odot \bar{x}(n)$

7: $\hat{m}(n) \leftarrow D(\hat{x}(n), c(k))$

8: **end for**

9: $\max \hat{m}(n) \leftarrow \text{pooling}\ (\hat{m}(1), \cdots, \hat{m}(N))$

10: The final imputed matrix $\hat{x}(n)$ corresponding to $\max \hat{m}(n)$.

11: end for

distribution. However, this still remains a problem, the random noise vector is randomly sampled from a latent space, e.g., Gaussian distribution. It means that the generated values may change in a range with the changing of the input random noise z. In other words, the imputed value generated by single random sample z may has a distance to the ideal imputed value as shown in Fig. 5a.

Inspired by the uncertainty solution of multiple imputation [22, 23], a novel multiple imputation by TGAIN is designed as the second stage of our imputation framework. This stage function is to find a best vector z from the latent input space so that the generated sample $G(z)$ can be mostly close to the latent ideal value $\tilde{x}$. To do this, multiple random samples are input into the well-trained TGAIN's generator to generate imputed values, and the well-trained TGAIN's discriminator is utilized to measure the degree of imputation fitness for each generated sample. The maximum imputation fitness corresponds to the ideal imputed value. This multiple imputation by TGAIN is shown in Fig. 5b. Therefore, the multiple imputation network by TGAIN is designed as the combination of multiple generate–evaluate pooling as shown in Fig. 6. Here, the max-pooling structure is used to integrate the evaluation results, compute the maximum imputation fitness and give the most reasonable imputed value.

The multiple imputation by TGAIN stage is presented in Stage 2 as follows (Table 2):

4 Experiments and analysis

4.1 Traffic sensor data and setting

In this study, we evaluate the imputation performance of proposed TGAIN by two real-world traffic flow dataset. (a) Traffic section volume data by fixed road sensor in I90, Seattle, USA. (b) Traffic speed raster data by floating car GPS in Changchun, China. The testing sites and datasets were screened out to ensure the experiment data completeness for evaluate imputation performance conveniently and objectively, even though two types of data often have the problem of missing data in most cases.

4.2 I90 volume database

The data comes from Interstate 90 (I90) interstate highways in Seattle, USA, and collected by Digital Roadway Interactive Visualization and Evaluation Network (DRIVENet), which is an open-access database (http://wsdot.uwdrive.net). The selected data contains the vehicle volume counts record by 15 sensors which have the upstream and downstream correlation relation. The selected sub-area road sensor is shown in Fig. 7. The time period is January 01, 2015, to December 31, 2015, and holiday data are excluded for reducing the experiment complex. The sampling interval is 5 min. The traffic data from January 01, 2015, to September 30, 2015 (9 months, 75% of the total data) are used as training dataset, and the others (25%) are used as testing dataset. Here, to determine the optimal parameters for each of the experiment models and make sure unbiased estimate of the performance, the training dataset (75% of the total data) are split into training dataset A (25% of the total data) and training dataset B (50% of the total data). Training dataset A is used to search and determine the optimal models' architecture parameters and training dataset B is used to training the experiment models.

In the I90 dataset experiment, the conditional vector of TGAIN is set as the week label connect the time label, the time interval is set as 4 h for a better distinguish performance which the traffic distribution and tendency have obviously differences. The time label section is shown in Fig. 8.

4.2.1 Changchun speed database

This database comes from the GPS equipment installed on about 15,000 taxis in Changchun, China. The acquisition time period is April 01, 2018, to May 31, 2018, from 08:00 to 22:00 and exclude holidays for reducing the experiment

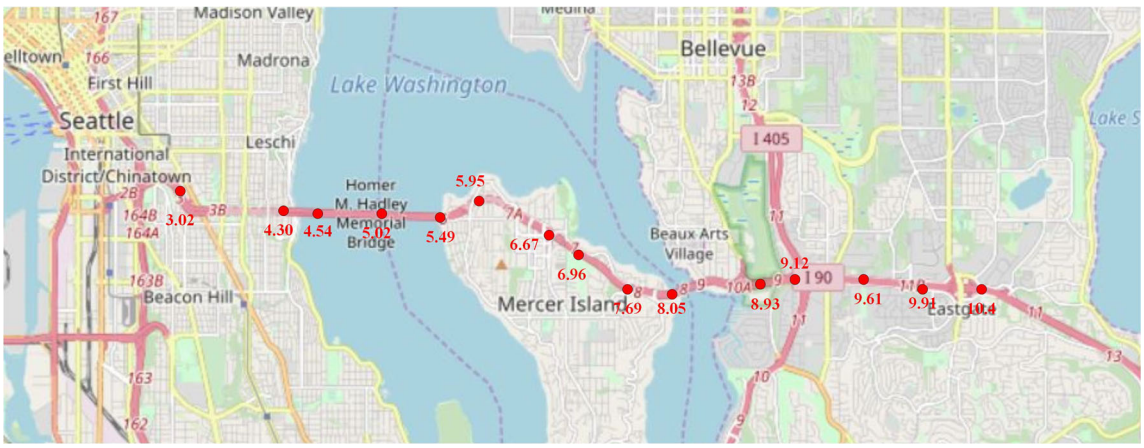

Fig. 7 Observation location on I90 interstate highways in Seattle, USA

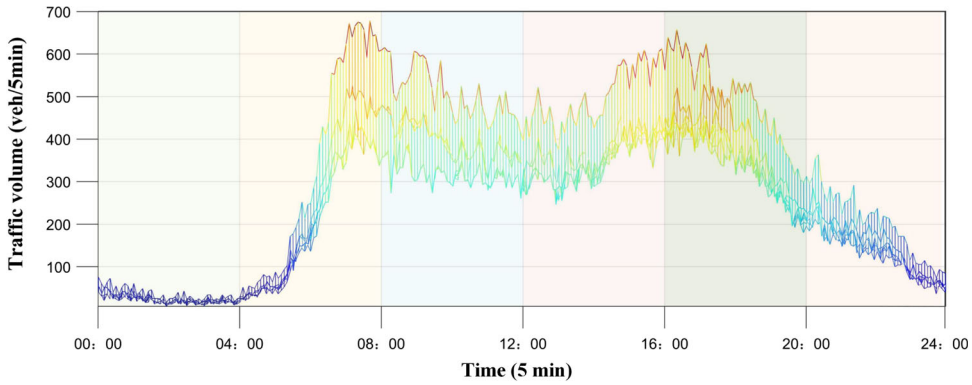

Fig. 8 Multistate time series traffic volume data partition by time label

Fig. 9 Test site on Weixing Street in Changchun, China

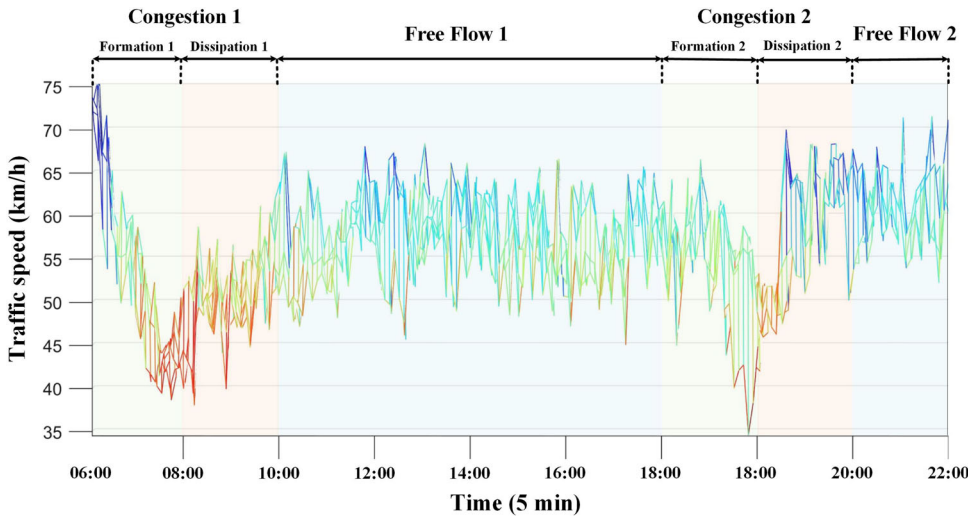

Fig. 10 Multistate time series traffic speed data partition by time label

complex. The test site located at Weixing road west-to-east direction in Changchun is shown in Fig. 9. The road network is gridded at about 150 m. We used ARCGIS software to filter the dataset and mapped the raw GPS point data into road segments by map matching algorithm in [41] and calculated the travel speed of each floating taxi car. Then the speed value for each road grid was calculated by average travel speed of floating taxi car within 5 min. The traffic data from April 01, 2018, to May 16, 2018 (41 days, 75% of the total data) are used as training dataset, and the others (25%) are used as testing dataset. The training dataset is also divided into two parts and used in the same way as I90 database.

The morning and evening traffic congestion is a common phenomenon in city road network. The traffic speed data shows obviously multistate characteristic during the congestion formation, dissipation and free flow state as shown in Fig. 10. In Changchun dataset experiment, the

conditional vector of TGAIN is still set as the week label connect the time label, but the time label is determined according to the speed data distribution time changing trends. The time label section is shown in Fig. 10.

4.3 Missing pattern and evaluation index

To reflect the complex distribution of MVs, the experiments simulate three common MVs pattern. (i) Missing completely at random (MCAR) where the propensity for a data point to be missing is completely random, i.e., independent of the observed data and the other missing data. In this pattern, MVs appear as a set of isolated points randomly distributed. (ii) Missing at random (MAR) where the occurrence of MVs depends on its neighboring MVs. As a result, this pattern looks like a group of successive MVs. (iii) A mixture of MCAR and MAR (MIXED), where the mixing ratio for MCAR and MAR is 0.5, indicating half of the MVs are from MCAR while the other half are from MAR. We also define missing ratio δ as the ratio of the number of MVs to the total number of values and change the value of δ from 0.1 to 0.9 with step 0.1 so as to simulate imputation problem with varying difficulties. In addition, the missing ratio δ of TGAIN's training stage is set to 0.3 to make sure relatively much information to learn the multistate distribution, which is a relatively high missing rate in general. In actual, the training dataset is the real detection dataset which could mix various missing rate samples.

To comprehensively evaluate the effectiveness of TGAIN, we compare it with several state-of-the-art imputation methods, including mean imputation, KNN [8], NNR [11], multiple imputation [22], SVD [17], PPCA [15], LLS [10], LRMC [18], SRSP [20], VIGAN [32], CollaGAN [33], GAIN [30] and MISGAN [42]. Among them, mean imputation is usually regarded as the baseline for MVs imputation, while the other belongs to different classes of methods (according to the taxonomy described in Sect. 2), e.g., regression model, probabilistic model and matrix completion model. The VIGAN, CollaGAN, GAIN and MISGAN are the variants of GAN for data imputation. The experiments are implemented in python 3.6. There are some parameters need to be set in each method, we first make the initial settings of model parameters according to the definition of corresponding algorithms and pervious research [20, 30, 43], the parameters in each method will further optimized by particle swarm optimization algorithm (PSO) [44, 45] to achieve the best imputation performance for traffic flow data in the experiment.

In experiments, missing scenarios are generated artificially and then different imputation methods are used to get a corresponding estimation. In order to quantitatively measure the recovery performance of imputation methods, the root mean squared error (RMSE) and mean absolute percentage error (MAPE), two widely evaluation indexes, are selected compute the differences between the imputed values and real values.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i^{impute} - x_i^{true}\right)^2} \tag{11}$$

$$\mathrm{MAPE} = \sum_{i=1}^{n}\left|\frac{x_i^{impute} - x_i^{true}}{x_i^{true}}\right| \times \frac{100\%}{n} \tag{12}$$

where n denotes the number of MVs, x_{true} and x_{impute} denote the real value, respectively. Considering the randomness when artificially simulating missing entries, the experiment was repeated five times for each method and calculate the average imputation error to ensure the imputation effect and stability.

4.4 Network parameters setting

In TGAIN network input parameters settings, the distribution of noise matrix elements is set as a standard Gaussian distribution, which ensures the unbiasedness of the generated input noise z. The conditional vector length depends on the label setting, the combination of week label and time label is adopted in the I90 volume database and Changchun speed database experiments. After One-hot encoding [39], the week label length is set as 7 (e.g., Monday code is [0000001], Sunday code is [1000000]), the time label division is shown in Figs. 8 and 10, and its length is set as 6. More importantly, the size of the input matrix $X \in R^{d\times N}$ is an extremely important parameter. Due to the division by conditional labels, the length N of under different C may be inconsistent, so it depends on the maximum sample length. If the length of a sample is less than N, it is filled with 0 elements and the marker matrix corresponding element is set as 1. The I90 volume database contains 15 sensors detecting data and time label interval is set as 4 h, so the input matrix dimension is set as 15*48. The Changchun speed database contains 18 sections detecting data and max time label interval is set as 8 h, so the input matrix dimension is set as 18*96.

The TGAIN network structure have an important influence on the imputation performance. The generator and decimator network layer and nodes number are the key parameters. In the experiment, the particle swarm optimization algorithm [39, 40] is adopted to determine the optimal network parameters. First, set different network layers and number of nodes for TGAIN, and calculate the corresponding imputation error. Then, the layer setting and node number are set as variables, the minimum imputation error is set as the optimization goal, and the network parameters are optimized by using PSO. Figure 11 gives

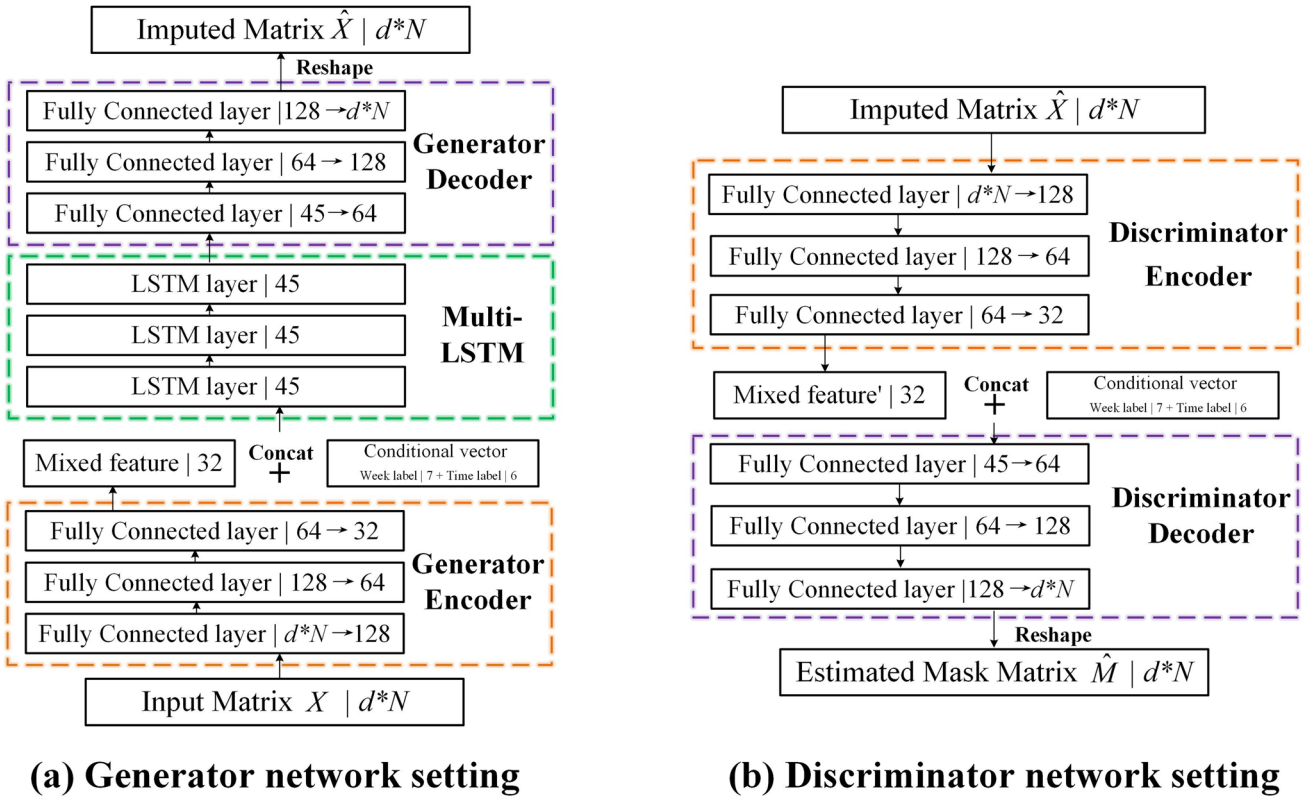

Fig. 11 TGAIN network parameters setting

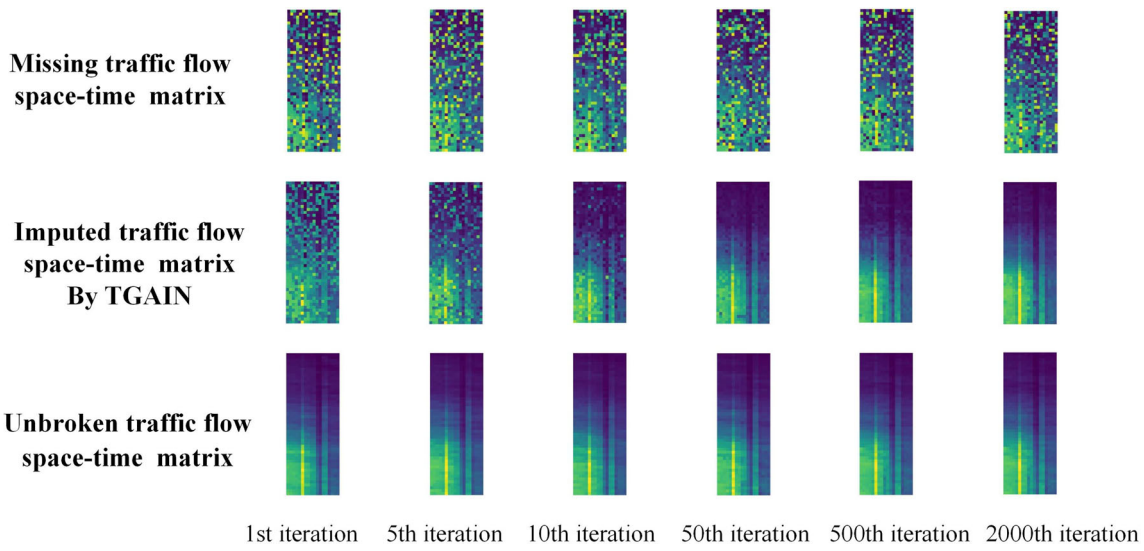

Fig. 12 Visualization of TGAIN training process (MCAR, δ = 0.5, 04:00–08:00) (The process is training and testing by G)

the optimal network parameters setting. Besides, the maximum number of training is set as 20,000, the network learning rate is set as 0.001, and the activation function and network optimizer are set as ReLU and Adam.

4.5 Convergence analysis of TGAIN

In the TGAIN, the generative adversarial imputation and MVs distribution learning process is optimized by iteration training. Next, we investigate the convergence behavior of this algorithm under varying missing ratios and different missing patterns. The visualization of training process on

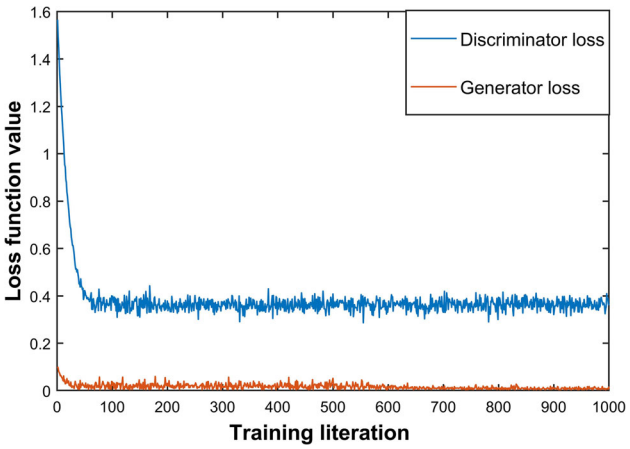

Fig. 13 TGAIN generator and discriminator loss convergence curve (MCAR, $\delta = 0.5$)

I90 dataset in the case of MCAR pattern and $\delta = 0.5$ is shown in Fig. 12 and some convergence curves of TGAIN's G and D loss function obtained in the experiments are shown in Fig. 13. As shown in Fig. 12, the third line is the unbroken traffic flow space–time matrix which could represent the ideal imputed matrix and the first line is the randomly generated missing matrix in the case of MCAR pattern and $\delta = 0.5$. The second line is the imputed matrix by TGAIN. With the increase in the iteration training, the imputed matrix by TGAIN is more similar and closer to the unbroken matrix. Meanwhile, the TGAIN generator and discriminator loss is iteratively convergent quickly in Fig. 13. It confirms the effectiveness and convergence of MVs distribution learning of TGAIN and the TGAIN can realize an effective mapping $G(z) = z \mapsto x$ based on learned distribution.

Table 3 Imputation error obtained by different methods under MCAR missing pattern. (I90 dataset)

Method	δ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean	RMSE	66.44	65.30	64.51	64.63	68.27	70.59	74.67	82.05	101.29
	MAPE	32.17%	33.60%	34.73%	34.41%	35.25%	37.24%	39.82%	43.63%	48.82%
KNN [8]	RMSE	42.03	42.90	43.61	43.08	48.92	49.50	52.66	58.45	59.59
	MAPE	18.37%	18.76%	18.98%	18.49%	20.23%	21.48%	23.79%	26.88%	29.02%
NNR [11]	RMSE	40.69	41.26	41.89	42.38	44.08	46.28	50.15	55.15	58.12
	MAPE	15.85%	16.65%	16.83%	17.21%	18.86%	19.48%	22.52%	25.98%	28.96%
MI [22]	RMSE	46.38	42.72	42.42	43.36	45.54	48.85	54.30	64.25	87.57
	MAPE	19.08%	17.47%	17.01%	18.61%	18.93%	20.09%	24.34%	34.24%	45.69%
SVD [17]	RMSE	39.45	42.37	43.57	45.64	48.68	51.00	52.50	68.76	80.25
	MAPE	15.91%	16.76%	17.08%	18.49%	20.23%	21.48%	23.79%	32.88%	42.02%
PPCA [15]	RMSE	36.78	38.57	40.33	43.25	46.69	50.25	54.56	62.49	78.65
	MAPE	13.87%	15.23%	17.03%	18.56%	20.07%	22.08%	24.78%	30.69%	39.26%
LLS [10]	RMSE	38.03	38.57	39.33	40.54	42.87	50.24	59.09	65.80	89.84
	MAPE	14.87%	15.65%	16.03%	17.97%	19.21%	21.23%	27.23%	32.34%	42.66%
LRMC [18]	RMSE	38.66	39.78	41.57	42.18	46.85	49.79	57.69	70.79	85.59
	MAPE	14.24%	15.45%	17.09%	17.17%	19.21%	21.19%	25.23%	34.34%	41.28%
SRSP [20]	RMSE	35.25	36.49	38.99	39.65	41.27	46.66	50.49	60.22	72.48
	MAPE	13.42%	13.85%	15.08%	15.23%	18.31%	21.18%	24.24%	29.32%	38.45%
VIGAN [32]	RMSE	40.57	41.24	43.56	44.22	45.33	48.2	51.61	54.63	58.62
	MAPE	16.51%	17.23%	18.52%	18.92%	19.56%	21.35%	23.62%	26.37%	28.64%
CollaGAN [33]	RMSE	39.65	40.97	42.69	42.97	44.93	47.63	49.64	52.14	55.62
	MAPE	15.62%	16.92%	17.83%	18.32%	19.85%	20.95%	22.61%	25.45%	27.93%
GAIN [30]	RMSE	35.21	36.86	38.48	39.27	40.47	43.39	47.16	48.55	52.96
	MAPE	12.91%	13.29%	15.39%	15.63%	17.72%	19.06%	20.70%	21.06%	24.50%
MISGAN [42]	RMSE	34.92	37.13	37.92	39.33	40.62	42.62	46.27	49.57	50.96
	MAPE	12.62%	13.21%	14.23%	15.61%	17.92%	18.67%	19.85%	21.37%	23.96%
TGAIN	RMSE	**34.03**	**35.36**	**35.08**	**35.49**	**36.57**	**38.38**	**40.96**	**41.02**	**42.99**
	MAPE	**12.52%**	**13.23%**	**13.01%**	**13.96%**	**14.69%**	**15.73%**	**17.98%**	**18.71%**	**20.33%**

The bold is used to highlight the results of the TGAIN

Table 4 Imputation error obtained by different methods under MAR missing pattern. (I90 dataset)

Method	δ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean	RMSE	65.09	64.62	65.26	65.38	68.63	70.72	74.68	82.10	100.18
	MAPE	32.54%	33.39%	34.26%	34.64%	36.66%	36.98%	37.70%	40.31%	45.30%
KNN [8]	RMSE	44.65	46.06	47.11	46.19	49.24	49.62	51.50	54.29	59.79
	MAPE	18.73%	19.34%	19.12%	20.66%	23.35%	23.38%	25.86%	27.87%	30.10%
NNR [11]	RMSE	41.26	42.24	43.02	44.38	45.72	47.92	50.12	53.61	58.92
	MAPE	16.02%	16.92%	17.12%	18.52%	20.23%	22.69%	24.61%	26.43%	28.64%
MI [22]	RMSE	43.74	44.02	43.91	44.14	46.77	49.30	54.56	64.87	89.30
	MAPE	18.62%	18.42%	18.35%	19.48%	20.24%	22.78%	28.21%	32.69%	41.78%
SVD [17]	RMSE	40.24	42.13	44.21	45.26	50.15	55.13	58.99	72.07	89.42
	MAPE	15.25%	17.31%	18.13%	19.21%	25.35%	26.45%	27.35%	36.57%	40.65%
PPCA [18]	RMSE	41.55	42.54	42.53	45.24	48.36	54.12	60.59	64.48	82.16
	MAPE	16.15%	17.08%	17.96%	19.12%	23.24%	25.18%	29.43%	31.35%	39.45%
LLS [10]	RMSE	40.02	41.15	43.31	42.50	49.85	58.86	71.48	82.59	91.27
	MAPE	15.29%	16.59%	18.08%	17.09%	22.25%	27.26%	35.67%	40.52%	43.77%
LRMC [18]	RMSE	40.15	41.27	42.29	44.99	48.24	50.94	57.17	63.15	70.48
	MAPE	15.12%	16.06%	17.14%	19.23%	22.15%	23.18%	27.23%	31.35%	36.45%
SRSP [20]	RMSE	39.87	40.79	41.58	42.89	45.48	49.58	55.59	59.55	65.29
	MAPE	15.45%	15.25%	16.755	18.23%	20.18%	23.12%	25.92%	26.67%	31.32%
VIGAN [32]	RMSE	42.64	43.27	43.78	45.23	46.26	48.67	49.86	52.07	53.37
	MAPE	16.23%	17.82%	18.07%	19.62%	20.43%	22.62%	23.85%	24.96%	25.13%
CollaGAN [33]	RMSE	41.27	42.61	43.63	45.92	46.87	47.69	48.56	51.24	54.13
	MAPE	15.96%	17.23%	18.23%	19.06%	19.82%	21.48%	22.65%	25.46%	26.85%
GAIN [30]	RMSE	38.36	40.39	40.46	42.39	43.31	45.08	45.20	49.43	50.95
	MAPE	14.92%	15.67%	16.43%	16.72%	18.96%	20.35%	20.67%	22.32%	22.62%
MISGAN [42]	RMSE	38.97	40.49	41.27	42.97	43.03	44.63	45.36	46.26	48.62
	MAPE	14.96%	15.82%	16.37%	17.43%	18.52%	19.67%	20.38%	21.26%	23.04%
TGAIN	RMSE	38.53	39.14	38.95	39.53	40.59	40.48	41.09	42.74	44.82
	MAPE	14.87%	14.96%	14.11%	15.87%	16.96%	17.01%	18.12%	18.96%	20.13%

4.6 Comparison experiments on I90 dataset

Tables 3, 4, 5 list the imputation errors of different algorithms under MCAR, MAR and MIXED missing patterns, respectively. We can see some interesting points from these tables. Firstly, MCAR and MAR are the easiest and hardest situation among three missing patterns, respectively. The reason is continuous data missing will cause the losing of more valuable information among the dataset, and increase the difficulty of accurate imputation. Secondly, the baseline mean imputation is the worst in terms of imputation accuracy because it relies on the fixed distribution assumption while ignoring the difference of multistate distribution. Thirdly, the imputation error of each method increases with the missing rate. These imputation methods could performance well in the low missing rate, because enough information could provide to direct the model statistic and calculation. The mean imputation, SVD, PPCA, LLS, LRMC, SRSP will rapidly degrade when missing ratio increases, the reason is their models mainly utilize the data local correlation and less consider the regularity information contained in historical dataset. The imputation error of KNN, NNR, VIGAN, CollaGAN, GAIN, MISGAN and TGAIN have a slowly growth relatively and they essentially utilize the data under same distribution state. The results also indicate these models possess better robustness and can satisfied the missing rate fluctuation in reality. Fourth, the VIGAN, CollaGAN,

Table 5 Imputation error obtained by different methods under MIXED missing pattern. (I90 dataset)

Method	δ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean	RMSE	64.79	64.75	65.25	66.45	68.07	70.76	74.44	82.07	101.43
	MAPE	31.32%	32.99%	33.40%	34.97%	35.36%	35.33%	37.49%	40.85%	48.21%
KNN [8]	RMSE	42.20	44.48	46.43	45.70	47.32	48.96	53.55	56.89	59.88
	MAPE	18.21%	19.12%	20.65%	20.18%	21.11%	22.15%	26.23%	28.38%	29.40%
NNR [11]	RMSE	40.38	41.25	41.68	42.15	43.25	47.95	52.12	53.85	57.32
	MAPE	15.93%	16.23%	16.89%	17.62%	18.96%	21.45%	25.63%	26.89%	28.96%
MI [22]	RMSE	42.88	42.21	43.35	44.78	44.90	48.89	53.44	63.95	87.48
	MAPE	18.39%	18.32%	18.45%	19.36%	19.56%	21.65%	26.79%	29.45%	40.75%
SVD [17]	RMSE	41.24	42.36	43.90	46.49	50.15	53.56	57.65	71.85	88.17
	MAPE	16.75%	17.24%	17.59%	20.23%	22.48%	25.65%	27.51%	35.98%	42.86%
PPCA [15]	RMSE	38.60	40.67	41.53	44.42	47.26	52.16	56.17	63.42	80.65
	MAPE	14.12%	15.08%	16.25%	18.96%	18.29%	25.45%	26.35%	30.56%	39.65%
LLS [10]	RMSE	40.53	39.71	42.95	44.31	46.73	54.68	60.80	69.89	85.03
	MAPE	15.06%	15.12%	17.58%	18.09%	20.16%	24.38%	27.26%	33.57%	41.84%
LRMC [18]	RMSE	38.95	40.12	41.23	42.22	45.27	50.17	56.47	68.17	82.16
	MAPE	14.57%	15.14%	16.12%	17.96%	19.25%	23.15%	26.35%	33.36%	39.96%
SRSP [20]	RMSE	37.55	38.56	40.46	41.15	42.21	48.26	52.22	60.55	68.16
	MAPE	14.18%	14.23%	15.42%	16.54%	18.28%	22.45%	24.35%	27.35%	32.45%
VIGAN [32]	RMSE	40.89	41.91	42.27	42.95	43.51	47.62	50.12	53.65	55.78
	MAPE	16.42%	17.13%	17.82%	18.91%	19.64%	22.49%	23.61%	25.62%	27.39%
CollaGAN [33]	RMSE	40.27	41.36	43.29	43.62	44.03	46.33	51.02	53.16	56.14
	MAPE	15.81%	16.87%	17.94%	18.53%	19.71%	21.65%	22.68%	24.57%	26.43%
GAIN [30]	RMSE	38.43	39.06	40.40	41.54	43.63	45.72	47.72	48.26	49.27
	MAPE	14.13%	14.96%	15.74%	16.76%	19.79%	21.12%	21.58%	22.07%	22.91%
MISGAN [42]	RMSE	37.86	39.62	40.36	42.27	42.99	44.66	46.91	47.02	48.69
	MAPE	13.96%	14.82%	15.27%	17.53%	18.26%	19.57%	20.85%	21.37%	22.86%
TGAIN	RMSE	**36.40**	**36.80**	**38.66**	**39.04**	**40.53**	**41.53**	**42.69**	**43.02**	**44.10**
	MAPE	**13.54%**	**13.79%**	**14.85%**	**15.21%**	**16.83%**	**17.91%**	**18.09%**	**19.37%**	**20.91%**

The bold is used to highlight the results of the TGAIN

GAIN and MISGAN could have an excellent imputation performance from various missing situation. They verify the distribution approach ability of GAN. Among them, the MISGAN and TGAIN are relatively better, because they both utilize the hint information to guide the distribution learning process. In particular, TGAIN achieves best performance in most cases and the comprehensive comparisons confirm the handling ability of uncertainty of imputation and multimodal distribution solution.

To clarify the imputation performance, Fig. 14 shows some imputation results obtained by different methods in the case of MCAR pattern and $\delta = 0.3$. The imputation residual obtained by TGAIN is smallest and the imputation tendency and performance is satisfied.

4.7 Influence of parameters on imputation error

4.7.1 The α of balancing the masked reconstruction and discriminative loss of TGAIN's G

According to Eq. (10), the α is an important factor which balances the generated matrix reconstructed error for observed portion and imputed error for missing portion. We change the α in range of {0.1, 0.5, 1, 3, 5, 7, 10, 20} and record the imputation error under different missing situation. Some experiment results under MCAR pattern and different missing rates are shown in Fig. 15. As shown in Fig. 15, the imputation error decreases and then increases with increasing α, and $\alpha = 5$ works best. It means

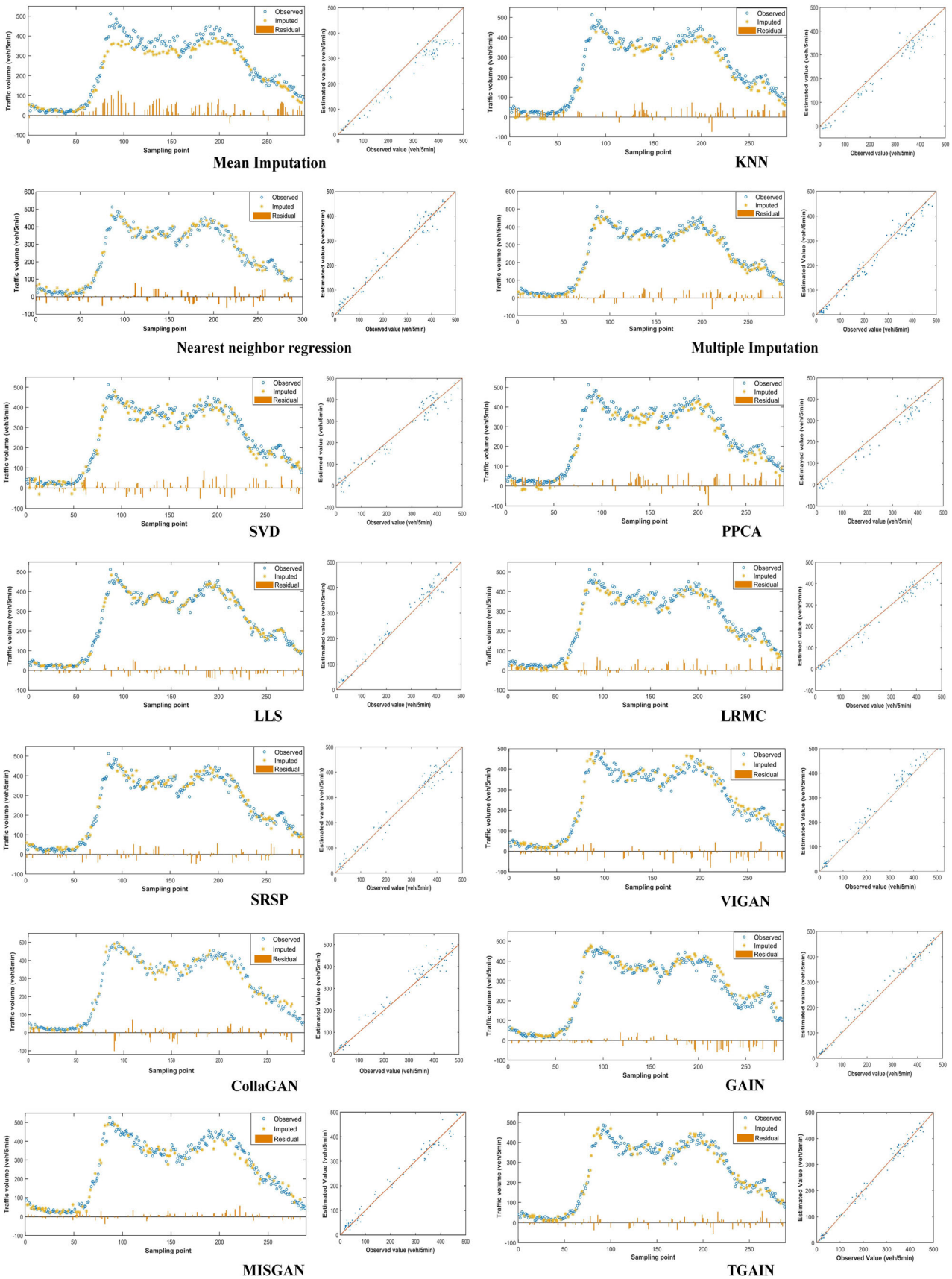

Fig. 14 Partial imputation results obtained by different methods (4.30 observation spot, MCAR, $\delta = 0.3$)

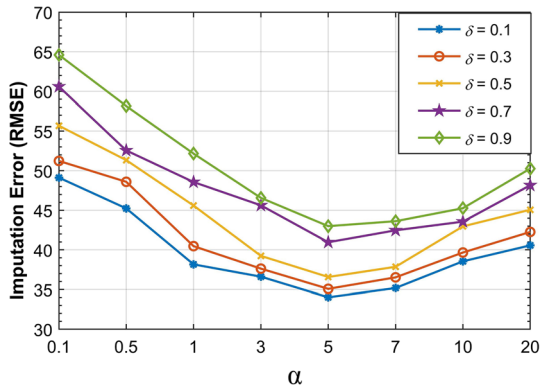

Fig. 15 Imputation error in different value of α (MCAR)

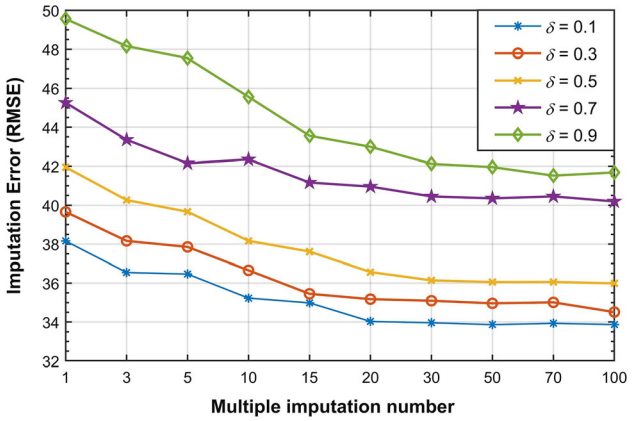

Fig. 16 Imputation error in different multiple imputation number

that TGAIN first make sure the 'real' portion of generated matrix is same as observed portion, and then let the 'fake' portion successfully fool the discriminator. When the α is too large, it will lead to overfitting problem for 'real' portion and weak the imputation probability of G. In the experiment, the α is set as 5.

4.7.2 The imputation number *n* of multiple imputation by TGAIN

The imputation number n is the key of TGAIN multiple imputation. It effects the determination of 'best' imputation value and imputation performance. In the parameter search phase, we change the number of multiple random matrixes and try to find the satisfactory result. Some experiment results under MCAR pattern and different missing rates are shown in Fig. 16. As shown in Fig. 16, the imputation performance gets better with the imputation number n increases. This further verifies the success of MVs distribution learning by TGAIN training stage and the significance of dealing with uncertainty by TGAIN multiple imputation stage. More imputation number need more compute resources and computation time. The multiple imputation number needs be comprehensive consideration between imputation performance and computational efficiency. In the experiment, the imputation number n is set as 30.

4.8 Comparison experiments on changchun dataset

To further verify the applicability of TGAIN algorithm, Changchun speed dataset was utilized into the traffic MVs imputation. Table 6 lists the imputation errors of different algorithms under MCAR missing patterns. From these results, the speed MVs imputation task is easier than volume MVs imputation, because the multistate characteristic of traffic speed is more obviously and data fluctuation range is relatively smaller. The results show that most imputation methods have relatively small imputation error (RMSE and MAPE) in low missing rate and the error rapidly degrade with the increase of missing rate δ. Overall, the TGAIN has the minimum absolute and relative error in most missing condition and also has a better filling performance stability.

In order to better show the imputation performance under different traffic distribution state, the evaluation indexes were calculated, respectively, for each of the experiment methods according to the partition in Fig. 10. Figure 17 shows RMSE and MAPE of different methods under different traffic state during weekday and weekend. The imputation performance changes along with traffic state, the speed data under congestion formation state has sharp decline trend and strong randomness, this increases MVs imputation difficulty and the imputation error in this state is the biggest. The error of congestion dissipation state is relatively small for the data distribution stability and tendency consistency. Due to the random fluctuation of free flow state, there is no obvious difference between weekday and weekend. On the whole, the TGAIN has a superior imputation ability under different data states, and the results confirmed TGAIN's

Table 6 Imputation error obtained by different methods under MCAR missing pattern. (Changchun dataset)

Method	δ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Mean	RMSE	10.29	11.56	11.90	12.26	12.87	13.36	14.56	14.22	15.62
	MAPE	18.63%	18.95%	18.98%	20.12%	21.85%	22.16%	23.25%	23.14%	25.47%
KNN [8]	RMSE	5.35	5.65	5.58	6.15	6.72	8.95	10.08	12.06	14.91
	MAPE	7.99%	8.21%	8.35%	8.96%	9.57%	12.27%	15.78%	19.45%	23.16%
NNR [11]	RMSE	3.75	3.96	4.21	4.68	5.28	7.05	9.12	11.02	13.25
	MAPE	5.26%	5.89%	6.45%	7.12%	8.35%	10.92%	13.78%	17.96%	21.63%
MI [22]	RMSE	4.92	5.15	5.65	6.13	6.73	8.94	10.15	11.70	15.70
	MAPE	7.45%	7.92%	8.65%	9.15%	9.85%	14.13%	15.92%	18.66%	25.22%
SVD [17]	RMSE	4.45	4.79	5.27	5.36	6.16	7.49	8.96	11.89	13.26
	MAPE	6.52%	7.18%	7.28%	7.85%	8.75%	11.59%	13.66%	18.66%	21.56%
PPCA [15]	RMSE	4.11	4.25	4.64	5.15	5.73	7.78	8.79	10.92	12.27
	MAPE	6.12%	6.48%	7.15%	7.85%	8.92%	12.25%	13.96%	17.66%	19.87%
LLS [10]	RMSE	3.30	3.61	3.89	4.19	5.75	7.96	9.35	12.23	13.76
	MAPE	5.12%	5.34%	5.98%	6.56%	9.09%	12.36%	14.59%	19.66%	22.57%
LRMC [18]	RMSE	3.29	3.53	3.89	4.32	5.17	6.39	8.04	10.77	12.38
	MAPE	5.01%	5.45%	6.12%	6.84%	8.16%	10.25%	13.26%	17.90%	20.16%
SRSP [20]	RMSE	3.10	3.13	3.68	3.95	4.70	5.89	7.15	10.53	12.45
	MAPE	4.83%	5.03%	5.68%	6.16%	7.24%	9.36%	11.65%	17.56%	20.45%
VIGAN [32]	RMSE	4.33	4.86	4.90	5.07	5.54	5.98	6.92	10.98	11.05
	MAPE	6.42%	6.87%	7.06%	7.83%	8.36%	9.21%	10.59%	16.32%	17.96%
CollaGAN [33]	RMSE	4.42	4.91	5.01	5.17	5.42	6.24	7.02	11.24	12.02
	MAPE	6.82%	7.05%	7.12%	7.69%	8.03%	9.47%	11.26%	16.92%	18.15%
GAIN [30]	RMSE	3.13	3.08	3.45	4.09	4.72	5.60	6.22	9.72	11.33
	MAPE	4.95%	5.12%	5.87%	6.25%	7.31%	8.92%	10.26%	15.69%	18.36%
MISGAN [42]	RMSE	3.03	3.18	4.02	4.78	4.91	5.47	6.34	9.62	10.58
	MAPE	4.62%	5.31%	5.96%	6.37%	7.18%	8.82%	10.76%	14.92%	17.69%
TGAIN	RMSE	**2.99**	**3.06**	**3.85**	**3.66**	**4.33**	**5.07**	**5.77**	**7.62**	**8.18**
	MAPE	**4.58%**	**4.78%**	**5.12%**	**5.64%**	**6.78%**	**7.89%**	**8.65%**	**12.68%**	**13.63%**

The bold is used to highlight the results of the TGAIN

handling ability of uncertainty of imputation and multi-modal distribution solution.

5 Conclusion

To deal with the distribution solution for multistate time series missing data and the uncertainty of imputation process, a novel MV imputation framework is proposed based on generative adversarial network and multiple imputation. In the first stage, a novel TGAIN is built and it utilizes adversarial imputation process to suitably capture MVs latent distribution and information learning for multistate time series data. The adjustable condition vector and novel time series generator is constructed to direct the adversarial learning for each data state. In the second stage, to reduce the uncertainty of imputation, a new multiple imputation by TGAIN is adopted to determine the best filling value. TGAIN network structure is skillfully combined with multiple imputation process to overcome data distribution predefined defect. We apply the proposed method to two real-world traffic sensor datasets and the experiments results show the TGAIN multiple imputation has superior robustness and imputation performance, and better ability

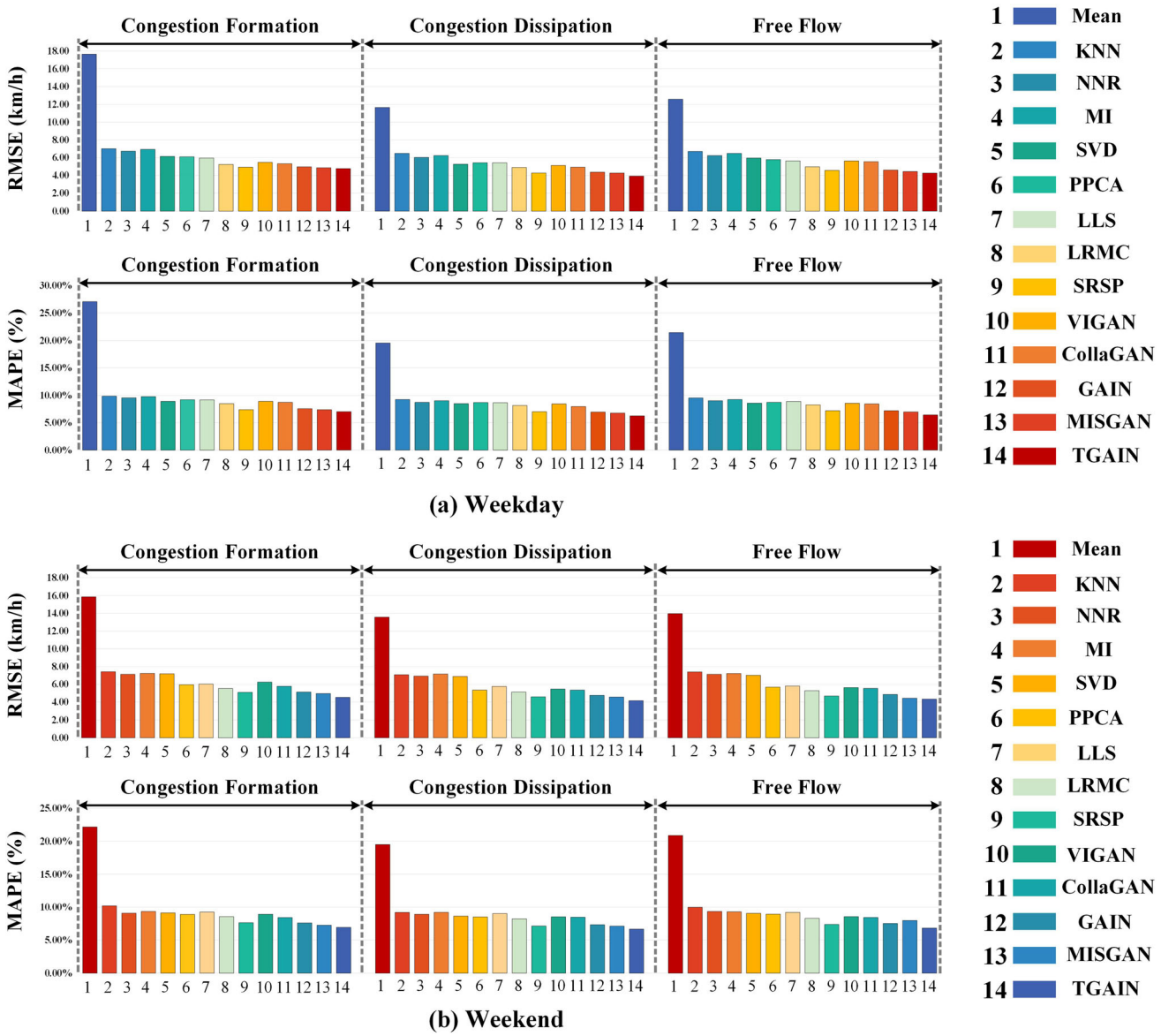

Fig. 17 Imputation error obtain by different methods under different traffic states during weekday and weekend (MCAR, $\delta = 0.5$)

in dealing with the uncertainty and distribution solution for time series data imputation than other state-of-the-art methods.

Furthermore, the proposed TGAIN imputation framework can be used as a general method for multistate time series MVs imputation in more fields (such as CPS and Health). The specific functional design of generator and conditional vector in TGAIN can be adjustable in other imputation tasks, and this would become a remarkable extension of TGAIN.

Acknowledgements This research is supported by the National Natural Science Foundation of China (Key Program) (52131202) and the Natural Science Foundation of Jilin Province (20190201107JC). The authors would like to thank the Digital Roadway Interactive Visualization and Evaluation Network (DRIVENet) for providing the traffic volume data used to validate this methodology.

Data availability All datasets and code supporting the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Li Z, Cao Q, Zhao Y et al (2018) Signal cooperative control with traffic supply and demand on a single intersection. IEEE Access 6:54407–54416. https://doi.org/10.1109/ACCESS.2018.2870172
2. Qu Z, Li H, Li Z et al (2020) Short-term traffic flow forecasting method with M-B-LSTM hybrid network. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2020.3009725. Accessed29July
3. Kalair K, Connaughton C (2021) Anomaly detection and classification in traffic flow data from fluctuations in the flow-density relationship. Transp Res Pt C-Emerg Technol 127:103178. https://doi.org/10.1016/j.trc.2021.103178
4. Farhangfar A, Kurgan LA, Pedrycz W (2007) A novel framework for imputation of missing values in databases. IEEE Trans Syst Man Cybern Syst 37(5):692–709. https://doi.org/10.1109/TSMCA.2007.902631
5. Guo Z, Wang Y, Ye H (2019) A data imputation method for multivariate time series based on generative adversarial network. Neurocomputing 360:185–197. https://doi.org/10.1016/j.neucom.2019.06.007
6. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR (2010) Pattern classification with missing data: a review. Neural Comput Appl 19(2):263–282. https://doi.org/10.1007/s00521-009-0295-6
7. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR et al (2009) K nearest neighbours with mutual information for simultaneous classification and missing data imputation. Neurocomputing 72(7–9):1483–1493. https://doi.org/10.1016/j.neucom.2008.11.026
8. Zhang S (2012) Nearest neighbor selection for iteratively KNN imputation. J Syst Softw 85(11):2541–2552. https://doi.org/10.1016/j.jss.2012.05.073
9. Kim H, Golub GH, Park H (2005) Missing value estimation for DNA microarray gene expression data: local least squares imputation. Bioinformatics 21(2):187–198. https://doi.org/10.1093/bioinformatics/bth499
10. Yu Z, Li T, Horng SJ et al (2017) An iterative locally auto-weighted least squares method for microarray missing value estimation. IEEE Trans Nanobiosci 16(1):21–33. https://doi.org/10.1109/TNB.2016.2636243
11. Buza K, Nanopoulosb A, Nagy G (2015) Nearest neighbor regression in the presence of bad hubs. Knowledge-Based Syst 86:250–260. https://doi.org/10.1016/j.knosys.2015.06.010
12. Wang G, Lu J, Choi KS et al (2020) A transfer-based additive LS-SVM classifier for handling missing data. IEEE T Cybern 50(2):739–752. https://doi.org/10.1109/TCYB.2018.2872800
13. Razzaghi T, Roderick O, Safro I et al (2016) Multilevel weighted support vector machine for classification on healthcare data with missing values. PLoS ONE 11(5):e0155119. https://doi.org/10.1371/journal.pone.0155119
14. Qu L, Li L, Zhang Y et al (2009) PPCA-based missing data imputation for traffic flow volume: a systematical approach. IEEE Trans Intell Transp Syst 10(3):512–522. https://doi.org/10.1109/TITS.2009.2026312
15. Folch-Fortuny A, Arteaga F, Ferrer A (2015) PCA model building with missing data: new proposals and a comparative study. Chemometrics Intell Lab Syst 146:77–88. https://doi.org/10.1016/j.chemolab.2015.05.006
16. Yuan X, Han L, Qian S et al (2019) Singular value decomposition based recommendation using imputed data. Knowledge-Based Syst 163:485–494. https://doi.org/10.1016/j.knosys.2018.09.011
17. Chen X, He Z, Wang J (2018) Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition. Transp Res Pt C-Emerg Technol 86(2018):59–77. https://doi.org/10.1016/j.trc.2017.10.023
18. Asif MT, Mitrovic N, Garg L et al (2013) Low-dimensional models for missing data imputation in road networks. In: EEE international conference on acoustics, speech and signal processing. IEEE, pp. 3527–3531
19. Chen X, Wei Z, Li Z et al (2017) Ensemble correlation-based low-rank matrix completion with applications to traffic data imputation. Knowl-Based Syst 132:249–262. https://doi.org/10.1016/j.knosys.2017.06.010
20. Chen X, Cai Y, Ye Q et al (2018) Graph regularized local self-representation for missing value imputation with applications to on-road traffic sensor data. Neurocomputing 303:47–59. https://doi.org/10.1016/j.neucom.2018.04.029
21. Chen X, Cai Y, Liu Q et al (2018) Nonconvex l(p)-Norm regularized sparese self-representation for traffic sensor data recovery. IEEE Access 6:24279–24290. https://doi.org/10.1109/ACCESS.2018.2832043
22. Harel O, Zhou XH (2007) Multiple imputation: review of theory, implementation and software. Stat Med 26(16):3057–3077. https://doi.org/10.1002/sim.2787
23. Murray JS (2018) Multiple imputation: a review of practical and theoretical findings. Stat Sci 33(2):142–159. https://doi.org/10.1214/18-STS644
24. Gondara L, Wang L (2018) Mida: multiple imputation using denoising autoencoders. Pacific-asia conference on knowledge discovery and data mining. Springer, Berlin, pp 260–272
25. Enders CK, Mistler SA, Keller BT (2016) Multilevel multiple imputation: a review and evaluation of joint modeling and chained equations imputation. Psychol Methods 21(2):222–240. https://doi.org/10.1037/met0000063
26. Goodfellow I, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680
27. Arjovsky M, Chintala S, Bottou L, (2017) Wasserstein generative adversarial networks. In: International conference on machine learning, pp. 214–223
28. Xu S, Zhu Q, Wang J (2020) Generative image completion with image-to-image translation. Neural Comput Appl 32(11):7333–7345. https://doi.org/10.1007/s00521-019-04253-2
29. Yang Y, Wang L, Xie D et al (2021) Multi-sentence auxiliary adversarial networks for fine-grained text-to-image synthesis. IEEE Trans Image Process 30:2798–2809. https://doi.org/10.1109/TIP.2021.3055062
30. Yoon J, Jordon J, Schaar M (2018) GAIN: missing data imputation using generative adversarial nets. In: International conference on machine learning, pp. 5675–5684
31. Luo Y, Cai X, Zhang Y, et al (2018) Multivariate time series imputation with generative adversarial networks. in: 32nd conference on neural information processing systems (NIPS), 2018, vol.31
32. Shang C, Palmer A, Sun J et al. (2017) VIGAN: missing view imputation with generative adversarial networks. In: 2017 IEEE International conference on big data (Big Data), pp. 766–775
33. Lee D, Kim J, Moon W J et al. (2019) CollaGAN: collaborative GAN for missing image data imputation. In: IEEE/CVF conference on computer vision and pattern recognition, pp: 2487–2496
34. Schafer JL, Olsen MK (1998) Multiple imputation for multivariate missing-data problems: a data analyst's perspective. Multivariate Behav Res 33(4):545–571. https://doi.org/10.1207/s15327906mbr3304_5
35. Ni D, Leonard JD (2005) Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data, Transp. Res. Record. In: 84th annual meeting of the transportation-research-board. 1935(1):57–67

36. Nielsen SF (2003) Proper and improper multiple imputation. Int Stat Rev 71(3):593–607
37. Li D, Li L, Li X et al (2020) Smoothed LSTM-AE: a spatio-temporal deep model for multiple time-series missing imputation. Neurocomputing 411:351–363. https://doi.org/10.1016/j.neucom.2020.05.033
38. Zhu J, Raghunathan TE (2015) Convergence properties of a sequential regression multiple imputation algorithm. J Am Stat Assoc 110(511):1112–1124. https://doi.org/10.1080/01621459.2014.948117
39. Yu L, Zhou R, Chen R et al (2022) Missing data preprocessing in credit classification: one-hot encoding or imputation? Emerg Mark Financ Trade 58(2):472–482
40. Li M, Zhang T, Chen Y et al. (2014) Efficient mini-batch training for stochastic optimization. In: 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp: 661–670
41. Kong QJ, Zhao Q, Wei C et al (2013) Efficient traffic state estimation for large-scale urban road networks. IEEE Trans Intell Transp Syst 14(1):398–407. https://doi.org/10.1109/TITS.2012.2218237
42. Li SCX, Jiang B, Marlin B (2019) MisGAN: Learning from incomplete data with generative adversarial networks. In: International conference on learning representations
43. Fan J, Chow TWS (2017) Matrix completion by least-square, low-rank, and sparse self-representations. Pattern Recognit 71:290–305. https://doi.org/10.1016/j.patcog.2017.05.013
44. Gao S, Zhou M, Wang Y et al (2019) Dendritic neuron model with effective learning algorithms for classification, approximation and prediction. IEEE Trans. Neural Netw. Learn. Syst 30(2):601–614. https://doi.org/10.1109/TNNLS.2018.2846646
45. Wang J, Kumbasar T (2019) Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods. IEEE/CAA J Autom Sinica 6(1):247–257