ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/368589001

Bidirectional Spatial-Temporal Traffic Data Imputation via Graph Attention Recurrent Neural Network

Article *in* Neurocomputing · February 2023

DOI: 10.1016/j.neucom.2023.02.017

CITATIONS
10

READS
161

6 authors, including:

Xiangjie Kong
Zhejiang University of Technology
221 PUBLICATIONS **7,448** CITATIONS

SEE PROFILE

All content following this page was uploaded by Xiangjie Kong on 01 March 2023.

The user has requested enhancement of the downloaded file.

Bidirectional Spatial-Temporal Traffic Data Imputation via Graph Attention Recurrent Neural Network

Guojiang Shen[a], Wenfeng Zhou[a], Wenyi Zhang[a], Nali Liu[a], Zhi Liu[a] and Xiangjie Kong[a,*]

[a]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

ARTICLE INFO

Keywords:
Missing data imputation
Spatiotemporal
Graph attention network
Recurrent neural network

ABSTRACT

Spatiotemporal traffic data is increasingly important in transportation services with the development of intelligent transportation system (ITS). However, due to various unpredictable disruptions in the data collection and storage process, traffic data is often incomplete which will seriously hinder downstream tasks if not handled properly. Most existing methods for traffic data imputation either impose too strong assumptions on the data distribution or almost ignore the interdependencies across time steps and the information expressed by missingness. In this article, we propose a graph attention recurrent neural network (GARNN) for traffic data imputation. In our model, we impute data from both temporal and spatial perspectives. First, we model the observations and missingness separately via two LSTMs to treat the missingness of data as another special information distinct from observations. Then, a decay mechanism and graph attention network (GAT) are applied to learn the interdependencies across time steps and capture the spatial correlations respectively to generate temporal estimation and spatial estimation. Finally, those two estimations are integrated into the ultimate imputation. The whole process is in a bidirection. The proposed method is evaluated on two public datasets under three different missing scenarios. Experimental results show the effectiveness of the proposed model compared with other baselines.

1. Introduction

With the rapid development of intelligent transportation system (ITS), many data-driven approaches based on deep neural networks have been proposed and applied to tankle traffic-related problems, such as traffic prediction [1], congestion recognition [2] and driving fraud detection of taxi [3]. These deep learning methods highly rely on sufficient historical spatial-temporal traffic data of high-quality. However, affected by natural and human factors, e.g., sensors failure, network communication error or storage anomaly, traffic data collected from the real world by various sensor devices are often incomplete or corrupted. It is reported that the complete traffic data facilitates subsequent tasks [4]. Thus, it is necessary to build appropriate algorithms to extract patterns from multidimensional spatial-temporal traffic features to impute missing values.

Over the past decades, many previous works have been proposed to address the problem of missing data. Classical statistical time series models such as ARMA and ARIMA [5] can be used to impute missing values, but they usually have many assumptions about the data such as linearity or smoothness, and the performance less desirable when the assumptions do not hold. Novel machine learning methods represented by support vector regression [6] and K-nearest neighbor (KNN) [7] are generally superior to classical methods when dealing with complex traffic problems such as prediction and incident detection with missing data, due to their ability to capture nonlinear relationships. Matrix factorization-based methods [8, 9, 10] and dimensional reduction techniques based on tensor decomposition [11, 12] are proposed to solve the traffic data imputation problem and have good performance in practice. However, these methods require matrices of low rank and dimensional information is critical to them. Recently, deep learning-based approaches, as a branch of machine learning, have shown extraordinary appeal for estimating the missing values. Methods based on DAE (Denoising Autoencoders) [13] like DSAE (Denoising Stacked Autoencoder) [14] are typical deep learning models and perform well. A class of algorithms based on Recurrent Neural Network (RNN) have been proposed and achieved state-of-the-art results for time series and sequence data [15, 16]. With the excellent performance of Convolutional Neural Network (CNN) in the image domain [17], the methods of modeling traffic data into images, and then using CNN to extract features have also been proposed, e.g. [18, 19]. CNN is efficient in handling Euclidean structured data but is insufficient for extracting non-Euclidean spatial relationships.

With the booming of deep learning, Generative Adversarial Networks (GANs) [20] provides an innovative perspective for data generation [21] and data imputation. They learn a mapping from random noise to real data distribution by playing an adversarial game with a pair of generator and discriminator. On the basis of GAN, Yoon et al. proposed GAIN [22], which provides the discriminator with additional auxiliary information about the missing position through a hint vector to improve the discriminative ability of the discriminator. Then, building on this work, Wang et al. [23] utilized potential category information to further enhance the imputation power by generating pseudo-labels through clustering for samples as a conditional input when training the model. Meanwhile, Kazemi et al. introduced an iterative GAN structure, called Iterative Generative Adversarial Network (IGANI), which iterates the generator by randomly shuffling the mask matrix to regenerate imputation data and

*Corresponding author.
gjshen1975@zjut.edu.cn (X. Kong); xjkong@ieee.org (X. Kong)
ORCID(s): 0000-0003-1064-1250 (G. Shen)

improve network generation capability [24]. These general inference models usually focus on data distribution rather than spatiotemporal correlations between data.

To tackle the problems listed above, in this paper, we propose a bidirectional spatiotemporal imputation approach for traffic data via graph attention recurrent neural network (GARNN) which is a combination of the RNN-based and Graph-based. We handle missing values through a bidirectional recurrent imputation process from both temporal and spatial perspectives, in which missing values at each timestamp are generated by merging the corresponding temporal estimations and spatial estimations. We model the observation and missingness through two independent LSTM structures, and a fusion layer with a decay mechanism is introduced to learn the time regularities of the data to generate the temporal estimation. To excavate the spatial dependencies among the traffic flow for generating spatial estimation from the spatial perspective, a Graph Attention Network (GAT) and a LSTM are used. In this way, GARNN achieves remarkable performance for traffic data imputation.

The main contributions of this article are summarized as follows:

- We design a novel spatial-temporal model which captures complex spatiotemporal correlations to generate temporal estimation and spatial estimation separately in a bi-directional recurrent process to impute traffic missing values. During backpropagation, delay gradients in the forward and backward directions are obtained separately under the consistency constraints, making the estimation more accurate.

- We utilize a union of LSTMs and a combination of GAT and LSTM to model the temporal correlations and spatial dependencies of traffic data respectively, which make the imputation result more reasonable. Furthermore, we design a new bidirectional fusion method based on historical observation, which improve the effect of bidirectional imputation fusion.

- We conduct comprehensive case studies on two real-world traffic datasets. In addition, we evaluate the model performance in three practical scenarios under different missing rate. The experimental results show the effectiveness of the model.

The rest of the paper is organized as follows: the existing studies on traffic data imputation are reviewed in Section 2. Section 3 formulates the imputation problem and presents the missing data types. Section 4 introduces our method. Section 5 discusses the experiment of the tested method. Conclusions are drawn in Section 6.

2. Related work

In this section, we review the existing works of traffic data imputation. The literature is divided into three categories: statistical methods, RNN-based methods, and GCN-based methods.

2.1. Statistical methods

Traffic data with missing is often unavoidable. Most algorithms are based on the absence of missing data, and thus data pre-processing is necessary. Deletion is the most straightforward method to handle this but may appear to lose useful information if there are a large number of missing values [25]. A more preferable method is to use the existing historical data for inference. Some simple examples of the statistical methods try to directly utilize some statistical features to fill in the missing positions. Yin et al. took the historical mean value of the previously known data from the same sensor over the same period to replace the unobserved data [26]. Little and Rubin [27] introduced a method of repairing missing data with the last observation, which is based on the assumption that the data fluctuates little over a short period of time. KNNs imputation can be regarded as a pattern-similarity estimation which computes the estimated values by averaging the known values of the k neighbors in the corresponding position [28]. Chang et al. combined KNN with local least squares to enhance the interpolation accuracy by considering the relationship between traffic flow patterns [29]. Although those methods can achieve good imputation results to a certain extent, they are based on the assumption that traffic flow is high similar from day to day, which means that they cannot be adjusted according to random events [30]. Principal component analysis (PCA)-based methods including probabilistic PCA [31], kernel PCA [32], functional PCA [33], and Bayesian PCA [34] usually assume a prior distribution and perform data imputation based on the prior. However, it may cause errors while the actual distribution of data is unknown.

2.2. RNN-based methods

The essence of traffic data is time series data [35]. Imputation of time series is a key component of missing traffic data imputation. RNN and its variants are generally very expert in dealing with time series, thus, there are many RNN-based methods to solve this problem. The first time RNN developed to solve the time series imputation problem was proposed by Bengio and Gingras [36]. Their proposed method first pads the vacant position with mean imputation, and then the imputation values will be continuously updated by the feedback links of an RNN until the model converges. Furthermore, bidirectional imputation also appeared. Yoon et al. [37] employed a bi-directional RNN to acquire the temporal dependencies and exploited a fully connected layer to learn the correlations of different variables. From the temporal perspective, in an incomplete time series, information from different elapsed time steps will contribute differently to the current missing values estimate. For this consideration, Tian et al. designed a novel LSTM variant LSTM-M to infer missing data through a multi-scale temporal smoothing method. They applied a decay vector to the cell state of the LSTM to measure the effect of erratic time gap, and added the missingness vector directly to the internal computation process of the LSTM to model complex polymorphic missing patterns [38]. Cao et al. proposed an

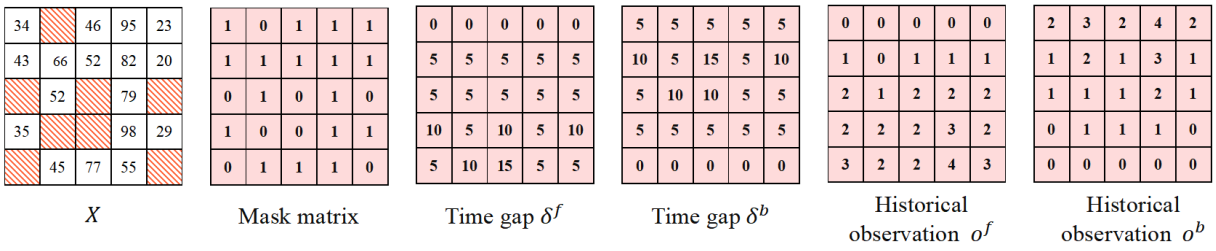


Figure 1: Illustration of the mask matrix, time interval matrices and historical obsercation matrices.

innovational RNN-based model for multivariate time series imputation, named Bidirectional Recurrent Imputation for Time-series (BRITS) [39]. In the model, a temporal decay factor and a special linear layer were designed to capture the spatial-temporal features of data with irregular intervals. In many RNN-based models, they cannot directly process data with null values, and usually use some pre-defined values, like zeroes, mean of historical, or last observation values to initialize missing values which results in the model learning biased parameters [40]. To enable the LSTM to process the sequence data with null values without being filled with any pre-defined values, Cui et al. constructed a new LSTM structure with an imputation unit, in which the missing values at the current time step were inferred from preceding LSTM cell states and hidden states in a bi-direction [41], and the methods achieved impressive performance in traffic data imputation tasks. Although the above methods can achieve certain effects in traffic data repair, they neglect the network topology so that spatial correlations cannot be modeled well.

2.3. GCN-based methods

In recent years, Graph Convolutional Networks (GCN) [42], has received constant focus which benefits from its strong ability to model non-Euclidean spatial correlations by aggregating graph node information through edges [43]. As graph structure arises naturally in the traffic Network, GCN-based methods [44, 45] have shown great promise in the traffic field. Wu et al. used a fixed-weight adjacency matrix predefined by distance to represent spatial dependencies to design a GCN-based model to recover unobservable sensor data [46]. The spatial structure in many scenarios often changes dynamically evolving time. Liang et al. introduced a novel traffic data imputation model including a graph structure estimation layer and a graph convolutional layer which are used to model dynamic spatial correlations and capture spatial dependencies respectively [47]. Moreover, Zhong et al. modeled the dynamic correlations between different road segments from the perspective of heterogeneous graph and developed a spatial-temporal framework for tasks of traffic prediction with incomplete data [48]. These methods usually consider the local rather than the global information. To address this issue, a motif-based graph aggregation method was proposed in [49]. This method first presents a graph aggregation through motif discovery and then utilizes the GCN to capture the high-order spatial correlations on the aggregation results. As an improvement of GCN, Graph Attention Network [50] can assign different weights to different neighbors through a self-attention mechanism, thus it can learn spatial dependencies better compared with GCN's equal-looking approach. Ye et al. [51] first applied GAT to traffic data imputation task. They combined GAT with CNN to adaptively capture the spatiotemporal correlations. In [52], GAT was introduced into GAN as a component of generator and discriminator to improve the spatial and temporal modeling capabilities of the model and has achieved promising imputation accuracy. However, those methods pay more attention to the spatial modeling, while ignoring the interdependencies across time steps and the missingness information.

3. PRELIMINARY

3.1. Preliminary and Problem Formulation

Given traffic data collected from sensors from N locations in a specific region, we define a road network as an undirected graph $G = (V, E, A)$, where V is a set of vertexes representing road segments in the road network with $|V| = N$. E is a set of edges denoting the spatial correlations between vertexes. The adjacent matrix of graph G is structured by matrix $A \in \mathbb{R}^{N\times N}$. Where, the sensor i and sensor j are neighbors if $a_{ij} = 1$, otherwise $a_{ij} = 0$. At each timestamp $t \in \{t_0, t_1, t_2, \cdots, t_T\}^T$, the ground truth traffic flow from all sensors in the traffic network is denoted as $X_t = [x_t^1, x_t^2, \cdots, x_t^N] \in \mathbb{R}^{1\times N}$, and $\mathscr{X} = [X_1, X_2, \cdots, X_T] \in \mathbb{R}^{T\times N}$means the collected data over all the T timestamps. Then we introduce a mask matrix $\mathscr{M} = (M_1, M_2, \cdots, M_T) \in \mathbb{R}^{T\times N}$to record the missing positions in $\mathscr{X}$. The mask matrix is defined as follow:

$$m_t^i = \begin{cases} 1, & \textit{if } x_t^i \textit{ is observed.} \\ 0, & \textit{if } x_t^i \textit{ is missing.} \end{cases} \tag{1}$$

On the basis of the mask matrix, we regard the non-missing observations as valid information, and define a historical observation variable $\mathscr{O} = (O_1, O_2, \cdots, O_T) \in \mathbb{R}^{T\times N}$ to measure how much valid information $X_{1:t-1}$ can provide to impute X_t. The historical observation can be computed as follow:

$$o_t^i = \begin{cases} m_{t-1}^i + o_{t-1}^i, & t > 1 \\ 0, & t = 1 \end{cases} \tag{2}$$

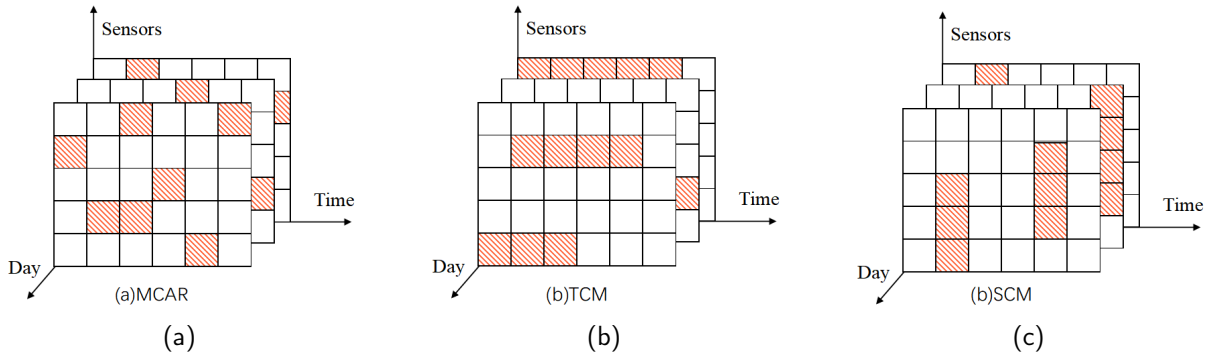


Figure 2: Visualization example for different missing types. Cells with slashes refer to missing positions, while the white ones are observations. They represent missing completely at random, temporally continuous missing and spatially continuous missing from left to right.

The temporal correlations between the observed data and missing data are of great concern to us since we intend to impute the missing data from a temporal perspective. However, the contribution of observations to missing value estimations may depend on the time interval between them, especially when the data are continuously missing along the time axis [53]. Therefore, we introduce a time slot matrix $\triangle = (\delta_1, \delta_2, \cdots, \delta_T) \in \mathbb{R}^{T \times N}$ to record the time interval information. $\triangle$ is defined to be the time gap from the last observation to the present unobserved value. As the missing data can be imputed from the forward and backward direction, the δ_i includes δ_i^f and δ_i^b which denotes the forward time gap and backward time gap. We can formulate the forward $\delta_{i,t}^f$ by the following equation:

$$\delta_{i,t}^f = \begin{cases} |s_t - s_{t-1}| + \delta_{i,t-1}^f, & if\, t > 1, m_{t-1}^i = 0 \\ |s_t - s_{t-1}|, & if\, t > 1, m_{t-1}^i = 1 \\ 0, & if\, t = 1 \end{cases} \quad (3)$$

where s_t represents the sampling time point of the data. Similarly, when calculating δ_i^b, we reverse the timestamps to $\bar{t} \in \{t_T, t_{T-1}, t_{T-2}, \cdots, t_0\}^T$. Fig.1 shows an example of notation specification. The time interval is 5-min.

In this article, we aim to make the imputed data as close as possible to the real data, thus, the imputation problem can be formulated as:

$$\min_{\hat{\mathcal{X}}} \sum_{i=t_0}^{T} \sum_{j=1}^{N} |\hat{x}_i^j - x_i^j|$$

$$s.t. \min_{\hat{\mathcal{X}}} \sum_{i=t_0}^{T} \sum_{j=1}^{N} m_i^j |\hat{x}_i^j - x_i^j| = 0 \quad (4)$$

where $\hat{x}_i^j \in \hat{\mathcal{X}}$ denotes the imputation of x_i^j.

3.2. Data Missing Types

Missing data have different types which have an important impact on the performance of the method. In our research, we mainly study the following two missing types: 1)**MCAR:** The data is missing completely at random; 2)**MAR:** The data is missing at random. MAR contains two missing modes: 1)Temporally continuous missing (**TCM**); 2)Spatially continuous missing (**SCM**). Both of them indicate that the data is continuously missing in the time or space dimension. In MCAR, which is typically caused by transmission error, all the missing positions appear randomly. TCM and SCM are more commonly caused by power failure to one within consecutive timestamps or more sensors in a timestamp. The visualization examples of MCAR and MAR can be seen from Fig.2.

4. Methodology

In this section, we introduce the proposed traffic data imputation framework. We start with the overview of the proposed model, and then give a detailed description of the calculation process.

4.1. Network Architecture

We propose a graph attention recurrent neural network (**GARNN**) for traffic data imputation in this article. Fig.3 describes the whole architecture of GARNN. As we can see in the figure. 3b, the model, contains three LSTM structures (a mask LSTM, an imputation LSTM, and a spatial LSTM) and one GAT structure. It takes a mask matrix, a value matrix, and a road network adjacency matrix as inputs to generate imputed data as outputs. The initial processing of GARNN is a union of two LSTMs to extract information from the value and the mask. Then we design a fusion layer with residual to combine the hidden vectors of these two LSTMs and map the new hidden vector to be a temporal decay estimation simultaneously. The temporal estimation is combined with the original value through the mask vector to generate preliminary complete data which serves as one of the inputs to the GAT to learn spatial dependencies at each timestamp. After this, the spatial LSTM synthesizes spatial information and generates a spatial hidden vector which is mapped to be a spatial estimation. Finally, we introduce an imputation layer that received these two candidates to

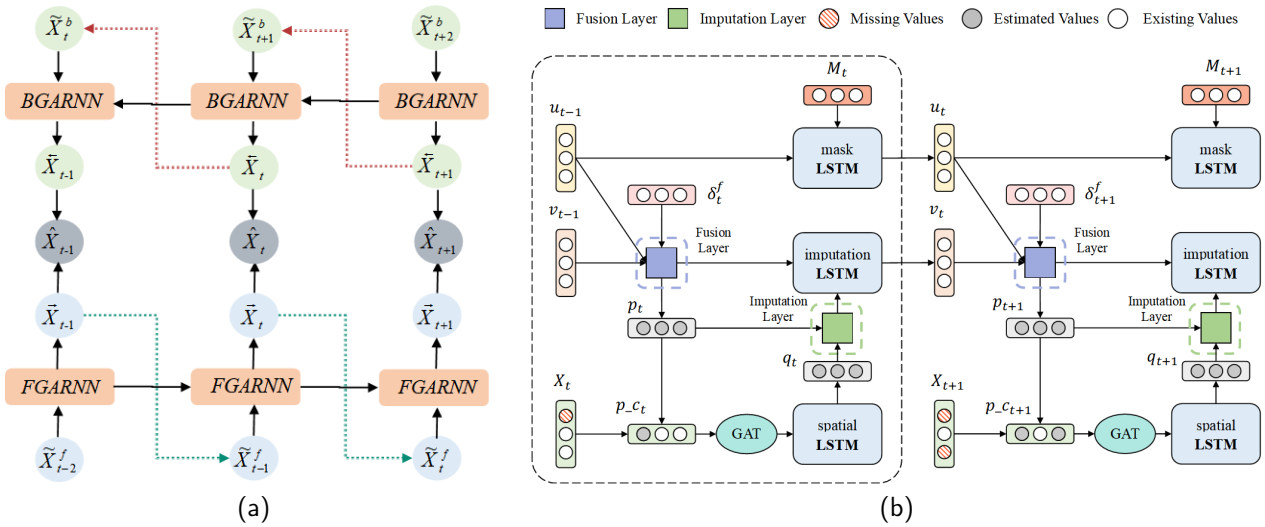


Figure 3: The architecture of model (left), and the unfolding of the forward process of GARNN (right). In the architecture, FGARNN and BGARNN represent the forward and backward directions of GARNN, the dotted line represents the replacement operation.

generate the final imputed data. Details will be given in subsequent subsections.

4.2. Exploration of Temporal Correlations and Spatial Dependencies

The previous analysis in Section 2 emphasizes the importance of the temporal correlations and spatial dependencies for traffic data imputation. The visualization of spatio-temporal dependencies is given in Fig 4. Accordingly, we

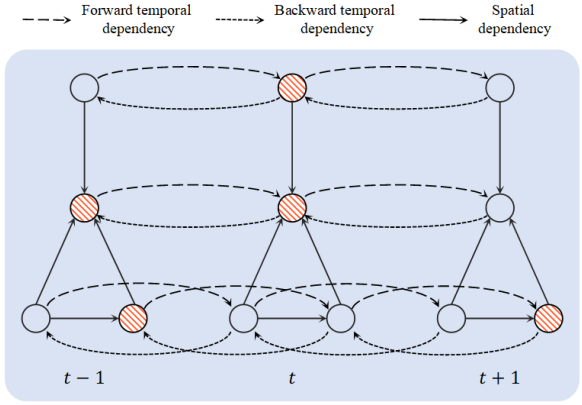


Figure 4: Illustration of spatio-temporal dependency. Circle with slashes indiccates missing.

impute the missing values from two perspectives. On the one hand, based on the temporal correlations learned from previous observations, we estimate missing data decaying with a missing pattern, called temporal decay estimation. On the other hand, we explore the dependencies of different sensors for imputation which we call spatial dependencies estimation. Finally, these two estimations are combined by a trade-off coefficient which is learned from the time gap and the corresponding masking information, to obtain the final unidirectional imputation result.

4.3. Temporal Decay Estimation

Temporal Information Extraction: In the previous work such as [39], the value and missingness are treated as a kind of information through concatenation operation, which may result in the loss of some important potential information. It is reasonable to regard them as two kinds of information since the value and the missingness are two different data types: the value is numeric, while the missingness is boolean [54]. Moreover, the mask matrix records the missing patterns of the data [53]. Thus, in this research, the value and missingness are handled separately as in [54] by two standard LSTMs. The shorthand formulas for these two LSTMs are $LSTM_X$ and $LSTM_M$, which are modeled for the value matrix and mask matrix. And, the hidden variables of these two LSTMs are abbreviated as v_t and u_t. Taking the forward direction as an example:

$$v_{t-1} = LSTM_X(\tilde{X}^f_{t-1}) \tag{5}$$

$$u_{t-1} = LSTM_M(M_{t-1}) \tag{6}$$

where the $\tilde{X}^f_{t-1}$ is the forward imputation of X_{t-1}. Specially, at the initial moment of calculation, that is, when $t = 1$, we randomly initialize v_t and u_t to v_0 and u_0, and then use them to calculate $\tilde{X}^f_1$.

Fusion Layer: To merge the information of the value and the missingness, we also use a fusion layer to combine two hidden variables as in the previous work [54]. Fig. 5 displays the structure of the fusion layer. The variable u_{t-1} of the mask LSTM is treated as a filter through a sigmoid activation function to preserve the useful information and discard the

useless information of v_{t-1}. The formula of the fusion layer is:

$$m\_g_{t-1} = \sigma(W_g u_{t-1} + b_g) \tag{7}$$

$$h_{t-1} = tanh(W_h[m\_g_{t-1} * v_{t-1}] + b_h) \tag{8}$$

where W and b are weight parameters and bias parameters respectively.

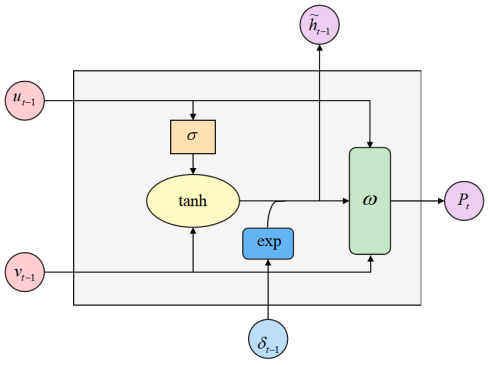


Figure 5: Illustration of the fusion layer with residual for merging information of missingness and values.

Temporal Decay Estimation with Residual: Intuitively, the influences of observations on the reconstruction of missing data will gradually diminish with the time interval between the last observations and missing positions getting longer. Therefore, we introduce a time decay gate λ_{t-1} which is derived from the time interval δ_{t-1} to filter the output hidden variable h_{t-1} as in [39, 55]. The λ_{t-1} is restricted to be between 0 and 1. We generate the hidden variable $\tilde{h}_{t-1}$ as follows:

$$\lambda_{t-1} = exp\{-max(0, W_\lambda \delta_{t-1} + b_\lambda)\} \tag{9}$$

$$\tilde{h}_{t-1} = h_{t-1} * \lambda_{t-1} \tag{10}$$

where W_λ is the weight parameter and b_λ is the bias parameter. The previous regularities of the observation and the missingness is implied in the variable $\tilde{h}_t$. Instead of generating temporal decay estimation directly through the $\tilde{h}_t$, we use a residual connection to preserve the original information in the hope of obtaining a more accurate estimate:

$$p_t = W_p * [\tilde{h}_{t-1}||v_{t-1}||u_{t-1}] + b_p \tag{11}$$

where W_p is the weight parameter and b_p is the bias parameter, and $p_t \in \mathbb{R}^{1\times N}$, and $||$ stands for concatenation operation.

4.4. Spatial Dependencies Estimation

Spatial Graph attention: In a traffic network, the traffic flow on a certain road is not only related to the previous moment but also related to its surrounding roads. This characteristic of traffic is usually established on the spatial dependence of the road network. Inspired by this, we use observation data to infer missing data from a spatial perspective. We use Graph Attention Network (GAT) to capture the spatial dependencies. Note that to allow GAT to capture a more complete spatial relationship, the missing data in X_t is replaced by the temporal decay estimation, and the original observations are retained:

$$p\_c_t = X_t * M_t + p_t * (1 - M_t) \tag{12}$$

where $p\_c_t \in \mathbb{R}^{1\times N}$. GAT incorporates a self-attention mechanism to learn dynamic spatial correlations. The graph self-attention operation is defined as:

$$gat_t^i = \sigma(\sum_{j\in N(i)} \alpha_t^{ij} W p\_c_t^j) \tag{13}$$

where $\sigma()$ is a non-linear activation function, $gat_t^i \in \mathbb{R}$ is the output value on vertex i at the timestamp t, $N(i)$ is the one-hop neighborhood of vertex i, $W \in \mathbb{R}$ is the training parameter. The attention coefficient α_t^{ij} is computed as follows:

$$\alpha_t^{ij} = \frac{exp(LeakyReLU(a^T[W p\_c_t^i||W p\_c_t^j]))}{\sum_{k\in N(i)} exp(LeakyReLU(a^T[W p\_c_t^i||W p\_c_t^k]))} \tag{14}$$

where $a \in \mathbb{R}^2$ is the training parameter.

Spatial Dependency Estimation with Residual: After the graph attention layer capturing the spatial information from adjacent vertex at timestamp t, a standard LSTM is performed to synthesize the spatial information. The shorthand formulas for spatial LSTM is $LSTM_S$. Based on learned spatial dependencies, we apply a fully connected layer to obtain a spatial dependencies estimation q_t via a residual connection as the temporal estimation:

$$h\_gat_t = LSTM_S(gat_t) \tag{15}$$

$$q_t = W_q * [h\_gat_t||gat_t] + b_q \tag{16}$$

where W_q is the weight parameter and b_q is the bias parameter.

4.5. Combination

Imputation Layer: Now, we have the temporal decay estimation p_t and the spatial dependency estimation q_t. How to determine the proportions of them to the final forward imputation result is the upcoming problem. Similar to [39], we transform the mask vector M_t and the temporal decay λ_t to learn a trade-off coefficient β_t. The comprehensive imputation result is designed as follows:

$$\beta_t = \sigma(W_\alpha[\lambda_t||M_t] + b_\alpha) \tag{17}$$

$$c_t = \beta_t * p_t + (1 - \beta_t) * q_t \tag{18}$$

$$\tilde{X}_t^f = X_t * M_t + c_t * (1 - M_t) \tag{19}$$

where W_α is the weight parameter and b_α is the bias parameter, $\sigma()$ is an activation function to scale β_t within the range of $(0, 1)$.

Finally, by iterating the above process, we can get the forward imputation $\tilde{X}^f_{t:t+T}$ and the backward imputation $\tilde{X}^b_{t:t+T}$ of the $X_{t:t+T}$. In order to effectively integrate them, we design a combination method based on the historical observation O to replace the mean method. The proportion of the forward and backward combination is related to the amount of valid information possessed by each position. The specific calculation process is as follows:

$$\gamma = \frac{O^f_{t:t+T}}{O^f_{t:t+T} + O^b_{t:t+T}} \tag{20}$$

$$\hat{X}_{t:t+T} = \gamma * \tilde{X}^f_{t:t+T} + (1 - \gamma) * \tilde{X}^b_{t:t+T} \tag{21}$$

where γ represents the proportion of the forward.

4.6. Loss

To optimize GARNN, we design a two-part loss. The first part is the reconstruction loss and the other is the imputation consistency loss. The former is a mean absolute error between the estimated value (temporal decay estimation, spatial dependency estimation, and combination estimation) and the original observed value, which makes the imputed values as close as possible to the real values. The latter is a mean absolute error acting on the forward imputation and the backward imputation. It enforces the imputation to be consistent in both directions. The formula of loss is as:

$$\begin{aligned} LOSS = \sum_{t=1}^{T} & L_e(X_t, p_t, M_t) + L_e(X_t, q_t, M_t) \\ & + L_e(X_t, c_t, M_t) + L_e(\tilde{X}_t^f, \tilde{X}_t^b) \end{aligned} \tag{22}$$

where L_e represents the mean absolute error. Algorithm 1 demonstrates a more detailed training procedure.

Algorithm 1 Training of GARNN.

Input:
Historial observations with missing **X**, Corresponding mask matrix **M**, adjacent matrix **A**, *LSTM* hidden_size **H**, training epoch **N**.

Output:
Imputed data $\hat{X}$.

Calculate the time-lag matrix δ^f and δ^b between observated time point based on mask matrix **M** accorrding to Equation (3).
While epoch $\leq N$ **do**:
  forward imputation (backward imputation):
  for $t < T$ **do**:
    Calculate the hidden variables of imputation *LSTM* and mask *LSTM* according to Equation (5) - (6);
    Fuse and filter the temporal information according to Equation (7) - (10);
    Generate temporal decay estimation p_t accorrding to Equation (11);
    Capture and synthesize the spatial dependencies according to Equation (12) - (15);
    Generate spatial dependencies estimation q_t accorrding to Equation (16);
    Learn a tradeoff cofficient β_t accorrding to Equation (17);
    Generate the forward(backward) imputation X_t^f (X_t^b) accorrding to Equation (19);
  End for;
  Compute the overall error to optimize the model accorrding to Equation (22);
End While;
Generate the final imputation results accorrding to Equation(20-21);
Return the imputed data $\hat{X}$.

5. Experiment

In this section, we first describe two real-world datasets which are used to evaluate our proposed model. Then we introduce the baseline models, the implementation details, and the evaluation metrics. Finally, we analyze the experiment result.

5.1. Dataset

PeMS 04 and PeMS 08: They are constructed from two districts 04 and 08 in California respectively. The dataset includes flow, speed, and occupy data which are collected by the Caltrans Performance Measurement System (PeMS) and aggregated into 5-minutes intervals that means there are 288 points in one day. In this paper, we only use the traffic flow data. The PeMS04 is collected from January 1, 2018, to February 28, 2018, through 307 sensors, a total of 59 days, we use the first 45 days as the training set and the remaining 14 days as the test set; while the PeMS08 is collected from July 1, 2016, to August 31, 2016, through 170 sensors, a total of 62 days, we use the first 50 days as the training set and the remaining 12 days as the test set. We investigate three missing data patterns as mentioned in Sect. 3.2, each of which has data missing rates ranging from 10% to 50% with 10% interval. The missing positions are randomly chosen according to the missing type and missing rate. Also, both the training and test datasets maintain the same missing rate and missing pattern during the experiments.

5.2. Experimental Settings

Baseline and Implementation Details: We make comparisons with several state-of-the-art methods to demonstrate the outstanding performance of GARNN:

Table 1
Imputation performance terms of MAPE(%) RMSE and MAE in MCAR.

Datasets	PeMS04					PeMS08				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
HA	29.67%	30.68%	31.14%	31.57%	31.93%	33.26%	34.71%	35.12%	35.40%	36.03%
	0.1095	0.1110	0.1122	0.1136	0.1149	0.1195	0.1207	0.1222	0.1242	0.1262
	0.0768	0.0775	0.0782	0.0790	0.0797	0.0809	0.0821	0.0833	0.0849	0.0866
KNN	15.96%	16.46%	17.02%	17.63%	18.39%	12.33%	14.30%	14.80%	16.29%	17.47%
	0.0631	0.0642	0.0665	0.0677	0.0707	0.0536	0.0551	0.0574	0.0611	0.0655
	0.0422	0.0429	0.0437	0.0450	0.0468	0.0359	00365	0.0374	0.0390	0.0415
BRITS	16.03%	17.37%	18.20%	19.31%	20.89%	12.05%	13.46%	14.64%	15.70%	17.35%
	0.0588	0.0610	0.0631	0.0652	0.0678	0.0510	0.0528	0.0554	0.0580	0.0615
	0.0379	0.0395	0.0409	0.0427	0.0448	0.0324	0.0337	0.0355	0.0378	0.0401
BGCP	17.61%	17.74%	17.86%	17.86%	17.79%	15.92%	16.19%	17.08%	16.49%	16.18%
	0.0585	0.0590	0.0589	**0.0588**	**0.0588**	0.0515	0.0516	0.0517	0.0517	**0.0518**
	0.0391	0.0391	0.0392	0.0391	**0.0392**	0.0347	0.0346	0.0347	0.0346	**0.0347**
PC-GAIN	19.69%	19.08%	19.81%	19.37%	20.29%	21.58%	19.97%	19.46%	19.52%	21.47%
	0.0632	0.0618	0.0613	0.0641	0.0673	0.0704	0.0678	0.0670	0.0680	0.0705
	0.0417	0.0412	0.0422	0.0429	0.0450	0.0442	0.0427	0.0421	0.0432	0.0456
ImputeRNN	16.07%	16.45%	17.10%	17.31%	17.78%	12.07%	12.92%	13.17%	13.78%	15.15%
	0.0607	0.0610	0.0628	0.0636	0.0652	0.0529	0.0540	0.0548	0.0552	0.0571
	0.0390	0.0392	0.0405	0.0411	0.0426	0.0334	0.0343	0.0349	0.0355	0.0372
GACN	20.41%	20.54%	21.37%	21.79%	22.83%	18.85%	20.81%	21.60%	23.02%	24.36%
	0.0673	0.0686	0.0695	0.0703	0.0713	0.0656	0.0693	0.0725	0.0757	0.0756
	0.0448	0.0459	0.0470	0.0479	0.0493	0.0439	0.0463	0.0487	0.0510	0.0519
GARNN	**14.39%**	**15.08%**	**15.75%**	**15.90%**	**16.38%**	**10.93%**	**11.71%**	**12.23%**	**12.70%**	**13.41%**
	0.0548	**0.0565**	**0.0582**	0.0599	0.0618	**0.0453**	**0.0467**	**0.0485**	**0.0505**	0.0527
	0.0357	**0.0366**	**0.0378**	**0.0390**	0.0406	**0.0300**	**0.0312**	**0.0325**	**0.0337**	0.0352

Historical Average (HA) [26]: It is a statistical method. We calculate the estimation of the missing values by averaging the previous 5 days.

K-Nearest Neighborhood (KNN) [28]: We select KNN as a representative of machine learning methods. It averages the values of neighboring samples to estimate the missing.

Bidirectional Recurrent ITS (BRITS) [39]: BRITS is a typical example of RNN-based methods. It applies temporal decay and a special linear layer on temporal and spatial dimensions to capture spatial-temporal correlations. In this paper, we use the same hyperparameters in [39].

BGCP [11]: A bayesian maxtrix/tensor factorization method, which uses Markov chain algorithm to learn the latent factor matrices.

PC-GAIN [23]: We choose PC-GAIN as an example of generation-based methods. It generates pseudo-labels for samples as potential category information to further improve the imputation capability of GAIN.The same hyperparameters in [23] are used in this paper.

ImputeRNN [54]: A typical RNN-based method. It utilizes two GRU networks to capture the temporal information, and a special linear layer to learn the spatial correlations of sensors as in [39].

GACN [51]: GACN is a GAT-based method, which uses graph attention network (GAT) and convolution neural network (CNN) to capture spatial dependencies and temporal correlations, respectively. The parameters are the same as in [51].

For our model, we use the Adam optimizer [56] as the optimization method with learning rate of 0.0005 to train the model. The LSTM hidden layer size is 1.5 times the number of sensors. The negative slope of *LeakyReLU* is set to 0.2. The batchsize is set to 64 during the training stage. The model is iterated for 1000 epochs. And the early stopping method is also used to find the best hyperparameter values and prevent the model overfitting. All experiments are implemented with PyTorch.

5.3. Evaluation Metrics

To appraise the imputation performance of the GARNN, we utilize three widely used performance metrics which are defined as follws:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{\hat{x}_i - x_i}{x_i}| \times 100\% \tag{23}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{24}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{25}$$

Table 2
Imputation performance terms of MAPE(%) RMSE and MAE in TCM.

Datasets	PeMS04					PeMS08				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
HA	29.30%	30.65%	31.02%	31.40%	31.75%	35.22%	36.42%	36.34%	36.54%	36.99%
	0.1087	0.1105	0.1121	0.1135	0.1145	0.1186	0.1201	0.1218	0.1236	0.1257
	0.0761	0.0774	0.0782	0.0790	0.0795	0.0804	0.0814	0.0829	0.0844	0.0863
KNN	17.41%	18.09%	18.49%	18.86%	19.67%	16.06%	15.76%	16.20%	17.44%	18.69%
	0.0671	0.0687	0.0700	0.0720	0.0745	0.0593	0.0602	0.0617	0.0655	0.0710
	0.0448	0.0457	0.0465	0.0479	0.0494	0.0385	0.0389	0.0397	0.0414	0.0442
BRITS	16.89%	18.52%	19.57%	20.08%	21.11%	14.40%	15.22%	15.76%	17.02%	18.10%
	0.0622	0.0642	0.0658	0.0679	0.0703	0.0551	0.0567	0.0590	0.0617	0.0648
	0.0397	0.0415	0.0430	0.0447	0.0466	0.0347	0.0363	0.0381	0.0403	0.0425
BGCP	17.80%	18.43%	18.05%	18.22%	18.32%	19.70%	19.23%	19.45%	19.56%	19.21%
	0.0581	**0.0582**	**0.0584**	**0.0586**	**0.0585**	0.0532	0.0531	**0.0530**	**0.0532**	**0.0530**
	0.0390	**0.0392**	**0.0392**	**0.0394**	**0.0394**	0.0353	**0.0352**	**0.0351**	**0.0352**	**0.0351**
PC-GAIN	20.01%	18.95%	19.00%	19.26%	20.93%	22.66%	19.51%	20.33%	19.50%	20.97%
	0.0651	0.0610	0.0633	0.0637	0.0687	0.0729	0.0665	0.0672	0.0684	0.0722
	0.0429	0.0407	0.0421	0.0426	0.0459	0.0458	0.0421	0.0424	0.0432	0.0463
ImputeRNN	16.74%	17.31%	17.65%	17.74%	18.58%	13.97%	14.58%	15.51%	16.01%	16.55%
	0.0626	0.0632	0.0643	0.0652	0.0670	0.0569	0.0569	0.0581	0.0587	0.0601
	0.0404	0.0408	0.0415	0.0424	0.0439	0.0360	0.0366	0.0378	0.0380	0.0391
GACN	19.63%	20.95%	21.16%	21.63%	22.98%	28.14%	28.20%	29.33%	29.65%	30.25%
	0.0664	0.0690	0.0697	0.0707	0.0715	0.0823	0.0850	0.0877	0.0883	0.0882
	0.0441	0.0463	0.0470	0.0485	0.0495	0.0502	0.0526	0.0545	0.0557	0.0565
GARNN	**15.97%**	**16.72%**	**17.18%**	**17.32%**	**17.48%**	**12.85%**	**13.50%**	**13.95%**	**14.78%**	**16.11%**
	0.0607	0.0620	0.0628	0.0638	0.0646	**0.0526**	**0.0528**	0.0539	0.0552	0.0589
	0.0395	0.0406	0.0410	0.0418	0.0425	**0.0349**	0.0355	0.0363	0.0372	0.0396

where n is the total number of missing data,and x_i and $\hat{x}_i$ denote the ground truth and the imputed values.

5.4. Imputation Performance

In this section, we first evaluate the imputation performance of our proposed model on the previously mentioned datasets under three data missing scenarios. Then we conduct ablation experiments to demonstrate the effectiveness of sub-components. Finally, we investigate the influence of hidden layer size of LSTM on model performance.

1)Imputation Results Under MCAR Pattern: We evaluate the imputation performance of GARNN on MCAR scenario for two datasets PeMS04 and PeMS08, respectively. The result is showed in Table 1. The model of GARNN has an advantage in the RM pattern with a low missing rate. GARNN outperforms BGCP, ImputeRNN, and GACN, which proves that GARNN can learn better spatiotemporal correlations. We can see that the performance of method is also different on different datasets. The overall performance of KNN is better than that of PC-GAIN on PeMS08, but it is opposite on PeMS04. We ascribe this to the periodicity of road network traffic flow. ImputeRNN is superior to BRITS which demonstrate the validity of handling value and missingness separately. BGCP achieves the second best baseline which benefits from its various optimization methods. GACN gets poor performance, the possible reason is that GACN fills missing values with zeros, which makes the model learn biased parameters, and CNN cannot learn temporal long-term dependencies well.

2)Imputation Results under MAR pattern: We report the imputation performance of GARNN and other baseline models for the continuous missing scenarios including Temporally continuous missing(TCM) and Spatially continuous missing(SCM) in Table 2 and Table 3. Under TCM pattern, GARNN achieves the second best result after BGCP on RMSE and MAE metrics, which may be due to the fact that we replace the missing positions at each moment with imputed values, and these imputed values have a cumulative effect on subsequent imputations. Moreover, GARNN only uses a short-term sequence data as input to impute the missing, while BGCP uses the entire dataset to infer which can get more information. Contrary to TCM, GARNN remains advanced in SCM, which means that GARNN can better mine the spatial correlation of traffic data. Interestingly, the KNN model performs better than the PC-GAIN model on the PeMS04 dataset, which is different from the other two cases. This may be related to the missing pattern, the continuous missing on the nodes (features) has a negative impact on the clustering result which is very important for PC-GAIN.

3)Evaluation of different components of GARNN: We verify the efficacy of different components in GARNN, testing the following variants:

Table 3
Imputation performance terms of MAPE(%) RMSE and MAE in SCM.

Datasets	PeMS04					PeMS08				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
HA	30.71%	30.52%	31.08%	31.36%	31.84%	34.53%	35.18%	35.76%	36.31%	36.23%
	0.1093	0.1109	0.1121	0.1134	0.1149	0.1195	0.1211	0.1221	0.1246	0.1264
	0.0768	0.0776	0.0782	0.0790	0.0797	0.0809	0.0820	0.0829	0.0849	0.0866
KNN	16.27%	16.51%	16.20%	17.44%	18.69%	12.31%	14.22%	14.29%	16.29%	17.84%
	0.0636	0.0647	0.0658	0.0682	0.0715	0.0541	0.0558	0.0579	0.0614	0.0683
	0.0423	0.0431	0.0438	0.0452	0.0473	0.0357	0.0366	0.0377	0.0393	0.0424
BRITS	18.01%	18.57%	19.78%	19.70%	20.93%	12.48%	13.36%	14.78%	16.27%	17.69%
	0.0608	0.0621	0.0642	0.0659	0.0681	0.0536	0.0541	0.0565	0.0590	0.0623
	0.0397	0.0408	0.0421	0.0432	0.0449	0.0344	0.0351	0.0371	0.0389	0.0408
BGCP	18.52%	18.40%	17.93%	17.99%	18.12%	16.00%	15.72%	15.26%	16.83%	16.93%
	0.0583	0.0583	0.0582	**0.0583**	**0.0583**	0.0518	0.0517	0.0516	0.0517	**0.0519**
	0.0393	0.0393	0.0391	0.0391	**0.0391**	0.0349	0.0346	0.0346	0.0347	**0.0349**
PC-GAIN	23.72%	20.15%	23.54%	23.91%	24.40%	25.49%	24.28%	22.96%	24.51%	24.75%
	0.0674	0.0639	0.0751	0.0752	0.0744	0.0783	0.0778	0.0767	0.0778	0.0785
	0.0448	0.0429	0.0493	0.0495	0.0493	0.0496	0.0495	0.0489	0.0498	0.0506
ImputeRNN	16.71%	16.83%	17.02%	17.57%	17.87%	12.50%	12.74%	12.92%	13.82%	13.67%
	0.0608	0.0618	0.0625	0.0645	0.0655	0.0537	0.0542	0.0549	0.0554	0.0527
	0.0392	0.0400	0.0404	0.0420	0.0402	0.0338	0.0345	0.0352	0.0358	0.0347
GACN	20.63%	21.05%	21.63%	22.72%	24.00%	25.77%	28.43%	29.88%	29.91%	30.81%
	0.0676	0.0692	0.0697	0.0705	0.0727	0.0799	0.0858	0.0856	0.0879	0.0878
	0.0460	0.0470	0.0476	0.0557	0.0564	0.0492	0.0549	0.0551	0.0558	0.0565
GARNN	**15.28%**	**15.24%**	**15.56%**	**15.86%**	**16.47%**	**10.99%**	**11.52%**	**11.91%**	**12.74%**	**13.12%**
	0.0560	**0.0567**	**0.0580**	0.0595	0.0618	**0.0454**	**0.0467**	**0.0486**	**0.0502**	0.0526
	0.0362	**0.0369**	**0.0378**	**0.0389**	0.0402	**0.0299**	**0.0311**	**0.0324**	**0.0337**	0.0352

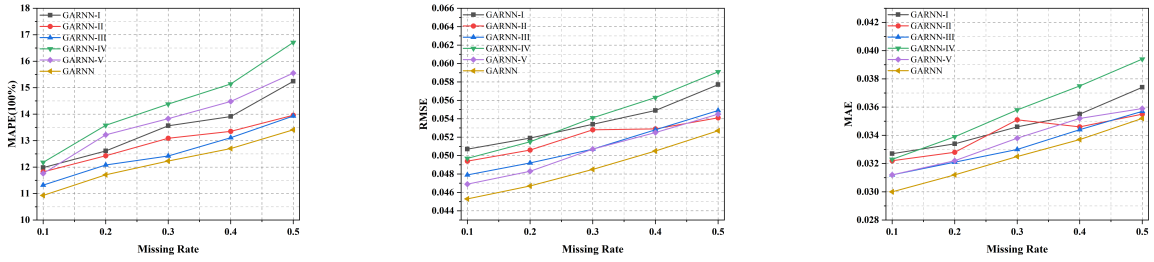
1) GARNN-I: GARNN without $LSTM_M$, which means a single LSTM for the value and missingness.
2) GARNN-II: GARNN without decay mechanism to measure the effect of the time interval.
3) GARNN-III: GARNN without residual connection.
4) GARNN-IV: The variant have no bidirectional imputation, only forward.
5) GARNN-V: The variant uses the mean method when merging the results of forward and backward imputation.

The ablation study results are showed in Fig.6 - Fig.8. From the result, we find that GARNN with all the components performs better than other variants, which proves the necessary of all the modules. Among the variants, GARNN without bidirection has much worse performance than others, suggesting the significance of bidirection for GARNN. Furthermore, the effectiveness of the proposed bidirectional imputation fusion method based on historical information is also demonstrated.

4)Impact of the size of hidden layer: Finally, we investigate the influence of the hidden layer size H. It is a hyperparameter that determines the output size of fusion information. We measure the performance when the missing rate is 10% under MCAR pattern on two datasets. Fig.9 presents the results. The N represents the number of sensors. We observe that the model performance shows a V-shaped trend as H increases. The model achieves the best performance when the H is 1.5 times the N. When the H is too small, the hidden variables can not preserve much useful information. When the H is too large, this results in much redundancy in hidden variables.

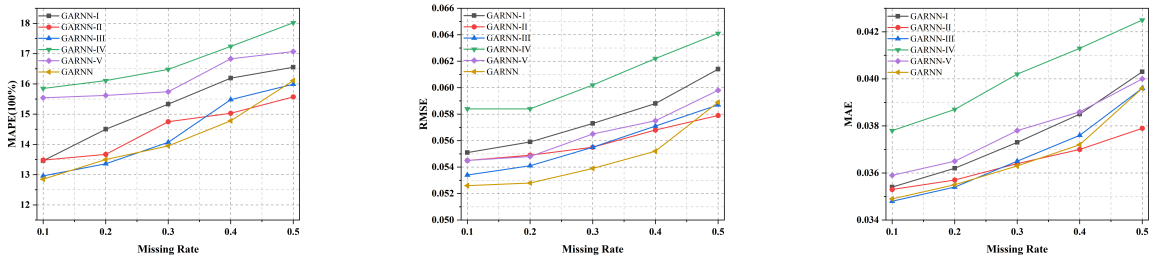
6. Conclusion

In order to address the missing traffic data problem, we propose a novel spatiotemporal imputation model combining LSTM and GAT in this study. We regard the missingness of data as another special information distinct from observations and model it by a mask LSTM. Then we apply a fusion layer to combine the two kinds of information. In order to take advantage of the interdependencies across time steps, a decay mechanism is borrowed to the LSTM structure. Besides, we also capture the spatial dependencies of the road network through GAT and a spatial LSTM. Finally, we design a new bidirectional fusion method to combine the forward and backward imputation. Experimental results on two real-world datasets under different missing scenarios demonstrate the feasibility of the method compared to baseline methods. In the future, we will try to incorporate adversarial learning with some auxiliary information such as weather into the proposed model to improve the imputation accuracy. Additionally, we plan to apply the proposed model to the traffic prediction problem with missing data.



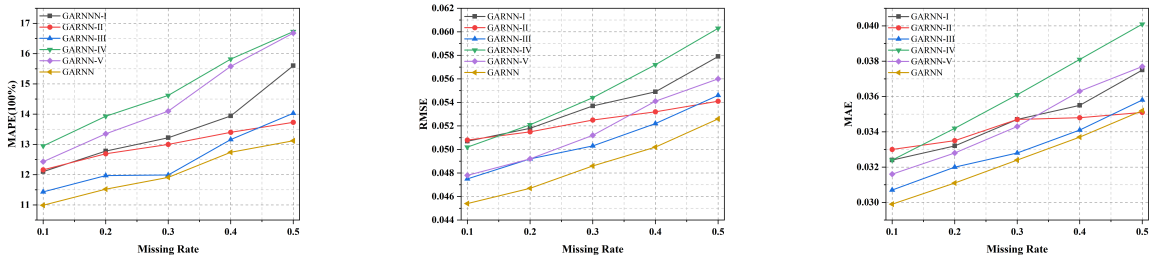
(a) (b) (c)

Figure 6: Analysis of ablation experiment result in MCAR.(a) MAPE of compared models. (b) RMSE of compared models. (c) MAE of compared models



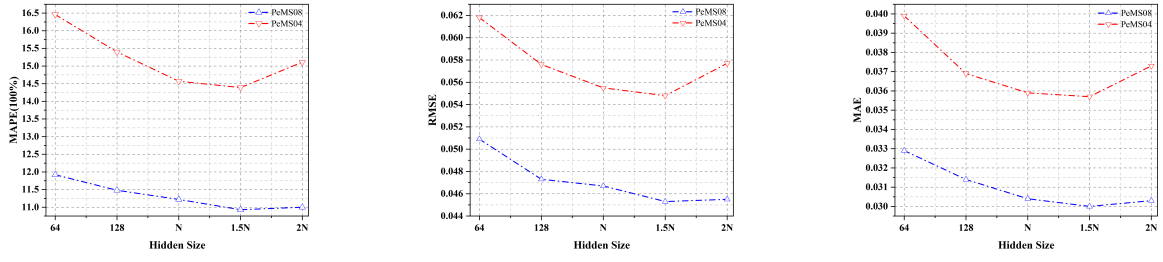
(a) (b) (c)

Figure 7: Analysis of ablation experiment result in TCM.(a) MAPE of compared models. (b) RMSE of compared models. (c) MAE of compared models



(a) (b) (c)

Figure 8: Analysis of ablation experiment result in SCM. (a) MAPE of compared models. (b) RMSE of compared models. (c) MAE of compared models



(a) (b) (c)

Figure 9: Performance of data imputation w.r.t the size of LSTM hidden layer. N represents the number of sensors.

Acknowledgement

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2022C01050, in part by the National Natural Science Foundation of China under Grant 62073295 and Grant 62072409, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, and in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant RF-B2020001.

References

[1] Li Zhu, Fei Richard Yu, Yige Wang, Bin Ning, and Tao Tang. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398, 2019.

[2] Xiao Han, Guojiang Shen, Xi Yang, and Xiangjie Kong. Congestion recognition for hybrid urban road systems via digraph convolutional network. *Transportation Research Part C: Emerging Technologies*, 121:102877, 2020.

[3] Xiangjie Kong, Bing Zhu, Guojiang Shen, Tewabe Chekole Workneh, Zhanhao Ji, Yang Chen, and Zhi Liu. Spatial-temporal-cost combination based taxi driving fraud detection for collaborative internet of vehicles. *IEEE Transactions on Industrial Informatics*, 18(5):3426–3436, 2022.

[4] Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, Salvatore Antonio Biancardo, and Junyi Zhang. Missing data repairs for traffic flow with self-attention generative adversarial imputation net. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[5] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] Muhammad Tayyab Asif, Justin Dauwels, Chong Yang Goh, Ali Oran, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic, and Patrick Jaillet. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):794–804, 2013.

[7] Pinlong Cai, Yunpeng Wang, Guangquan Lu, Peng Chen, Chuan Ding, and Jianping Sun. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, 62:21–34, 2016.

[8] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[9] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems*, 29, 2016.

[10] Xiaoyi Jia, Xiaoyu Dong, Meng Chen, and Xiaohui Yu. Missing data imputation for traffic congestion data based on joint matrix factorization. *Knowledge-Based Systems*, 225:107114, 2021.

[11] Xinyu Chen, Zhaocheng He, and Lijun Sun. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 98:73–84, 2019.

[12] Xinyu Chen and Lijun Sun. Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[14] Y. Duan, Y. Lv, Y. L. Liu, and F. Y. Wang. An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72(nov.):168–181, 2016.

[15] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[16] Danqing Kang, Yisheng Lv, and Yuan-yuan Chen. Short-term traffic flow prediction with lstm recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[17] Xiangjie Kong, Kailai Wang, Shupeng Wang, Xiaojie Wang, Xin Jiang, Yi Guo, Guojiang Shen, Xin Chen, and Qichao Ni. Real-time mask identification for covid-19: An edge-computing-based deep learning framework. *IEEE Internet of Things Journal*, 8(21):15929–15938, 2021.

[18] Yifan Zhuang, Ruimin Ke, and Yinhai Wang. Innovative method for traffic data imputation based on convolutional neural network. *IET Intelligent Transport Systems*, 13(4):605–613, 2019.

[19] Ouafa Benkraouda, Bilal Thonnam Thodi, Hwasoo Yeo, Monica Menendez, and Saif Eddin Jabari. Traffic data imputation using deep convolutional neural networks. *IEEE Access*, 8:104740–104752, 2020.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[21] Xiangjie Kong, Qiao Chen, Mingliang Hou, Azizur Rahim, Kai Ma, and Feng Xia. Rmgen: A tri-layer vehicular trajectory data generation model exploring urban region division and mobility pattern. *IEEE Transactions on Vehicular Technology*, 2022.

[22] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, 2018.

[23] Yufeng Wang, Dan Li, Xiang Li, and Min Yang. Pc-gain: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural Networks*, 141:395–403, 2021.

[24] Amir Kazemi and Hadi Meidani. Igani: Iterative generative adversarial networks for imputation with application to traffic data. *IEEE Access*, 9:112966–112977, 2021.

[25] Dong Li, Linhao Li, Xianling Li, Zhiwu Ke, and Qinghua Hu. Smoothed lstm-ae: A spatio-temporal deep model for multiple time-series missing imputation. *Neurocomputing*, 411:351–363, 2020.

[26] Weihao Yin, Pamela Murray-Tuite, and Hesham Rakha. Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods. *Journal of Intelligent Transportation Systems*, 16(3):159–176, 2012.

[27] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[28] Gang Chang and Tongmin Ge. Comparison of missing data imputation methods for traffic flow. In *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, pages 639–642. IEEE, 2011.

[29] Gang Chang, Yi Zhang, and Danya Yao. Missing data imputation for traffic flow based on improved local least squares. *Tsinghua Science and Technology*, 17(3):304–309, 2012.

[30] Yongchao Ye, Shuyu Zhang, and JQ James. Traffic data imputation with ensemble convolutional autoencoder. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1340–1345. IEEE, 2021.

[31] L Qu, L Li, Y Zhang, and J Hu. Ppca-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):512–522, 2009.

[32] L. Li, Y. Li, and Z. Li. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transportation Research Part C Emerging Technologies*, 34(sep.):108–120, 2013.

[33] Jeng Min Chiou, Yi Chen Zhang, Wan Hui Chen, and Chiung Wen Chang. A functional data approach to missing value imputation and outlier detection for traffic flow data. *Transportmetrica B*, 2(2):106–129, 2014.

[34] Vincent Audigier, Franois Husson, and Julie Josse. Multiple imputation for continuous variables using a bayesian principal component analysis. *Journal of Statistical Computation and Simulation*, 2016.

[35] Pan Wu, Lunhui Xu, and Zilin Huang. Imputation methods used in missing traffic data: a literature review. In *International Symposium on Intelligence Computation and Applications*, pages 662–677. Springer, 2019.

[36] Yoshua Bengio and Francois Gingras. Recurrent neural networks for missing or asynchronous data. *Advances in neural information processing systems*, 8, 1995.

[37] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Multi-directional recurrent neural networks: A novel method for estimating missing data. In *Time Series Workshop in International Conference on Machine Learning*, 2017.

[38] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

[39] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[40] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[41] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118:102674, 2020.

[42] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[43] Xiangjie Kong, Shiqin Tong, Haoran Gao, Guojiang Shen, Kailai Wang, Mario Collotta, Ilsun You, and Sajal K. Das. Mobile edge cooperation optimization for wearable internet of things: A network representation-based framework. *IEEE Transactions on Industrial Informatics*, 17(7):5050–5058, 2021.

[44] Guojiang Shen, Xiao Han, KwaiSang Chin, and Xiangjie Kong. An attention-based digraph convolution network enabled framework for congestion recognition in three-dimensional road networks. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14, 2021.

[45] Senzhang Wang, Meiyue Zhang, Hao Miao, and Philip S Yu. Mtstnets: Multi-task spatial-temporal networks for multi-scale traffic prediction. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 504–512. SIAM, 2021.

[46] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. Inductive graph neural networks for spatiotemporal kriging. *arXiv preprint arXiv:2006.07527*, 2020.

[47] Yuebing Liang, Zhan Zhao, and Lijun Sun. Dynamic spatiotemporal graph convolutional neural networks for traffic data imputation with complex missing patterns. *arXiv preprint arXiv:2109.08357*, 2021.

[48] Weida Zhong, Qiuling Suo, Xiaowei Jia, Aidong Zhang, and Lu Su. Heterogeneous spatio-temporal graph convolution network for traffic forecasting with missing values. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 707–717. IEEE, 2021.

[49] Difeng Zhu, Guojiang Shen, Jingjing Chen, Wenfeng Zhou, and Xiangjie Kong. A higher-order motif-based spatiotemporal graph imputation approach for transportation networks. *Wireless Communications and Mobile Computing*, 2022, 2022.

[50] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[51] Yongchao Ye, Shiyao Zhang, and James JQ Yu. Spatial-temporal traffic data imputation via graph attention convolutional network. In *International Conference on Artificial Neural Networks*, pages 241–252. Springer, 2021.

[52] Yichen Zhu, Mengtian Zhang, Bo Jiang, Haiming Jin, Jianqiang Huang, and Xinbing Wang. Networked time series prediction with incomplete data. *arXiv preprint arXiv:2110.02271*, 2021.

[53] Shuo Yang, Minjing Dong, Yunhe Wang, and Chang Xu. Adversarial recurrent time series imputation. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[54] Jiawei Ouyang, Yuhao Zhang, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. Imputernn: Imputing missing values in electronic medical records. In *International Conference on Database Systems for Advanced Applications*, pages 413–428. Springer, 2021.

[55] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pages 3094–3100. AAAI Press, 2019.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

View publication stats