IEEE Open Journal of Intelligent Transportation Systems

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJITS.2022.1234567

Attention-Driven Recurrent Imputation for Traffic Speed

Shuyu Zhang*, Chenhan Zhang†*, STUDENT MEMBER, IEEE Shiyao Zhang‡, MEMBER, IEEE AND James J.Q. Yu.*, SENIOR MEMBER, IEEE

[1]Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China.
[2]Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China.
[3]School of Computer Science, University of Technology Sydney, Sydney 2007, Australia

CORRESPONDING AUTHOR: James J.Q. Yu (e-mail: yujq3@sustech.edu.cn).

This work was supported by the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation No. 2020B121201001, by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2021KQNCX078, and by the Stable Support Plan Program of Shenzhen Natural Science Fund No. 20200925155105002.

ABSTRACT In practice, traffic data collection is often warned by missing data due to communication errors, sensor failures, storage loss, among other factors, leading to impaired data collection and hampering the effectiveness of downstream applications. However, existing imputation approaches focus exclusively on estimating the lost value from incomplete observations and ignore historical data. In this paper, we propose a novel neural network model, namely, Attention-Driven Recurrent Imputation Network (ADRIN), to address the problem of missing traffic data. Specifically, in ADRIN, we devise an Imputation-targeted Long Short-Term Memory (LSTM-I) module for filling in missing data. Meanwhile, we consider the periodicity of historical data and design a historical average calculation module in ADRIN. On this basis, we employ the multi-head self-attention mechanism for further extracting latent temporal features from the output of the two modules. ADRIN is capable of modeling both incomplete observation inputs and historical averages independently to estimate the missing values. We conducted comprehensive experiments on three real-world traffic datasets, to demonstrate that ADRIN consistently outperforms other baselines in a variety of scenarios. Furthermore, ablation experiments are conducted on the various modules of the model, and it is concluded that historical data can significantly enhance the imputation effect.

INDEX TERMS Traffic speed imputation, deep learning, long term short memory, self-attention, intelligent transportation systems.

I. INTRODUCTION

TRAFFIC data is a collection of historical road observations (e.g., flow and speed) acquired over a period and is considered as a critical component of Intelligent Transportation Systems (ITS) [1], [2]. On the basis of these data, the transportation department can exercise reasonable and effective traffic control, and businesses can provide more accurate and reliable service. Recently, there has been a surge in the development of deep learning-based algorithms for a variety of problems, including traffic speed prediction, origin-destination prediction, and travel time estimation, etc. [3], [4]. However, the majority of deep learning-based methods are highly reliant on high-quality data [5], [6]. In practice, traffic datasets frequently contain missing data due to sensor failures, regional power outages, extreme weather, among other reasons [7]. For example, more than 5% of the PeMS traffic data is missing [8]. Ni *et al.* [9] noted that data from the Texas Transportation Institute contained missing rates ranging from 16% to 93%. As a result, the issue of missing data needs to be addressed urgently.

To overcome the issue, the most straightforward approaches are to delete all data with the same 1) timestep or 2) sensor/road as unobserved. Both methods, however, imply loss of temporal or spatial information. To address this concern, data imputation is used to estimate missing values through the analysis of traffic data dependencies or distributions. Appropriate data imputation methods can accurately restore missing data, thereby avoiding the performance

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

degradation of various downstream data mining algorithms in intelligent transportation systems.

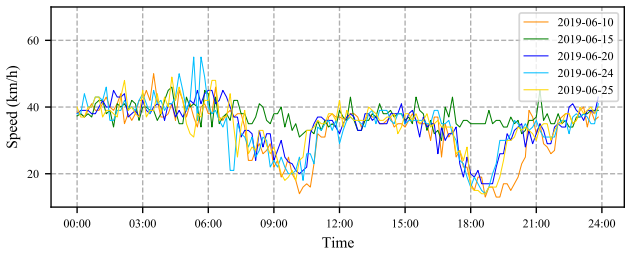


FIGURE 1. The traffic speed of East San Li He Avenue (Beijing) per 10 minutes on different days.

In addition, traffic data are highly cyclical and volatile in nature, compared with typical time-series data (e.g., stock indices and medical device data). Figure 1 shows the road speed of East San Li He Avenue in Beijing, China, over a five-day period. In general, the traffic speeds of successive days are similar. For example, the road speed is low during the morning and evening rush hours every day. Besides, traffic speeds vary throughout the day due to a variety of factors (e.g., weather, accidents, and dates.) and are highly correlated with the road speeds immediately preceding and following. Therefore, understanding how to combine the historical data and observations prior to and following different points in time is crucial to accomplish the missing data imputation task.

In the literature, researchers designed algorithms for data imputation based on statistical methods such as KNN [10], ARIMA [11]. These methods, however, are only effective for data with relatively simple distributions [12]. While some researchers have presented probabilistic models [13] and Gaussian distributions [14] using matrix decomposition [15], these methods are typically limited to statistical and low-rank data [16]. In recent years, deep learning has been widely utilized in the data imputation problem and achieved outstanding performance. For example, Cao *et al.* [17] proposed BRITS with a bidirectional recurrent neural network structure to combine the data prior to and after the residual location for imputation. Luo *et al.* [18] proposed E2GAN based on the adversarial generative network [19] structure to reconstruct the missing data. These studies, however, are undermined by slow computation or convergence on large datasets. Ye *et al.* [20] proposed GACN that incorporates conventional neural network and graph attention network. The GACN can estimate the missing value based on spatial and temporal dependencies. Nonetheless, the convolutional neural network-based structure results in its performance being unremarkable when facing massive missing. While current approaches have produced acceptable results for the imputation of traffic data, some challenges remain. For instance, parts of existing methods are overly complicated and challenging to train [18]. Additionally, most approaches consider only the temporal dependency within the incomplete time series data, failing to take advantage of the inherent periodicity of the traffic data. Furthermore, the question of how to better extract features from residual data continues thought-provoking.

To address the aforementioned issues and close the research gaps, we propose a novel data imputation model called Attention-Driven Recurrent Imputation Network (ADRIN). Unlike previous models, we extract features from both missing input and recent road speed to account for the volatility and periodicity of traffic data. ADRIN employs the long short-term memory network for imputation and the multi-head self-attention [21] network to extract features from missing data. Additionally, we apply the multi-head self-attention mechanism in conjunction with the recent traffic state to extract features associated with the historical information. The outputs of both modules are then routed through a fusion module that incorporates a self-attention layer and a linear layer to obtain the imputation result. Additionally, considering the spatial correlations is also essential for the missing data imputation, we designed spatial information powered ADRIN that contains graph convolutional network to extract spatial dependency in the road networks. We also design a new loss function to aid in training model. The major contributions and efforts are listed as following:

- We propose a novel network structure, Attention-driven Recurrent Neural Imputation Network (ADRIN) for reconstructing the incomplete input and historical average to improve the missing traffic data.
- We devise the long short-term memory for imputation (LSTM-I) that is designed for intermittent time-series data. In ADRIN, this LSTM-I network is integrated with multi-head self-attention mechanism to extract temporal features from the incomplete input.
- We propose a GCN empowered ADRIN that can impute the incomplete data based on the spatial-temporal information, the performance of the enhanced model demonstrates that the outstanding scalability of ADRIN.
- We conduct comprehensive experiments and analysis on three real-world traffic speed datasets, and the results demonstrate that our approach outperforms existing models by a significant margin. In addition, we investigate the role of the constituent components in ADRIN through ablation experiments and parameter tests.

The rest of this paper is organized as follows. In Section II, we briefly summarize the existing data imputation research. In Section III, we introduce the formulation of the proposed ADRIN, and give details to its constituting components and training process. Then, we present the experiments and discussions in section IV. Section V concludes this paper.

II. RELATED WORK

The subject of missing data imputation has gathered considerable interest from researchers with the growth of traffic big data research. In this section, we provide a systematic

review of related literature and classify these approaches into three categories: statistical learning-based methods, matrix factorization-based methods, and deep learning-based methods.

A. STATISTICAL LEARNING-BASED METHODS

Conventional statistical learning-based methods perform simple statistical operations for the missing data imputation task, such as linear interpolation [22] and complementing the missing values with history average or the last observation [23], which can address the problem only for simple data distributions. Some researchers proposed computing the missing values using data surrounding the missing positions in the feature matrix, such as the classical K-Nearest Neighbors Algorithm (KNN) algorithm [10], which imputes the missing by averaging the K nearest neighbors around. ARIMA [11] and its variants [24] impute the missing values by prediction based on historical data. Nevertheless, unlike the other methods for imputation, these approaches are unable to make effective use of the feature collected after the missing occurs. The constraint method [25] establishes rules for completing the missing value based on the overall data characteristics in the dataset; however, this approach is applicable only to univariate data and is less effective in the majority of the practiced scenarios involving multivariate data.

B. MATRIX FACTORIZATION-BASED METHODS

Matrix Factorization [15] discovers correlations within the data and imputes missing values by decomposing and reconstructing the traffic data matrix. Temporal Regularized Matrix Factorization (TRMF) [26] is a time-series imputation method that makes use of regularization schemes and a scalable matrix factorization approach. Additionally, Aude *et al.* [14] introduced Probabilistic Principal Component Analysis (PPCA) to matrix factorization, which presumes that the latent features of the observed data conform to a Gaussian distribution. Chen *et al.* [13] employed a more sophisticated low-rank tensor complementation algorithm to recover missing data. The proposed Bayesian Gaussia Candecomp/Parafac (BGCP) tensor decomposition method converts the original data matrix to a high-dimensional tensor and then describes and recovers the incomplete matrix. However, the matrix factorization-based methods need input with specific shape, which limit their application.

C. DEEP LEARNING-BASED METHODS

In the last decade, an increasing number of researchers have used deep learning techniques to extract the spatio-temporal dependencies for missing data imputation. Based on the Recurrent Neural Network (RNN) [27] for its time-series data modeling capability, GRUD [28] smooths the input using the historical average and the most recent observation before inputting into gated recurrent units [29]. Cao *et al.* [17] proposed BRITS, a bi-directional RNN structure for imputation that takes into account both the forward and backward directions of time-series data. However, the stepwise computation of RNNs renders their slow speed and high memory usage. Referring to the image coloring problem, Denoising Stacked Autoencoders (DSAE) [30] combines denoising and stacked encoders to estimate the unobserved values in the traffic data matrix. DSAE begins by encoding the data to extract implicit features and then decoding them to perform fitting and completion. However, it has certain limitations in terms of introducing significant fluctuations into the data.

Generative Adversarial Networks (GANs) based models [19] aim to generate the missing value by learning the general distribution of the training data. In Generative Adversarial Imputation Nets (GAIN) [31], the generator develops the complete data from observations and noise input, while the discriminator verifies the authenticity of the generated data. Additionally, Luo *et al.* [18] proposed E2GAN as an extension of GAN by incorporating an encoder-decoder RNN structure to the generator and discriminator to better model the temporal dependency. However, there are computation speed and convergence issues on large datasets. Mao *et al.* [32] proposed a novel data complementation model SSGAN by combining GAN and BRITS. The model, on the other hand, requires labeled categories in the input temporal data, limiting its applicability to the real-world scenarios. Recently, due to the superiority ability of attention mechanism to model the inter-feature dependencies, scholars introduced it to the text to aid in the missing data imputation task [33], [34], [16]. In [33], Yang *et al.* utilized the graph attention neural network [35] to learn the spatial dependence of data. In [34], Wu *et al.* applied the attention mechanism to model the correlation between features and solve the problem of missing data in database scenarios, limiting his application. In [16], based on the GAIN, Zhang *et al.* introduced the self-attention mechanism to better model the temporal dependence, but the problem of training difficulty persists. The GACN [20] applies convolutional neural network and graph attention network to extract the temporal and spatial dependencies, respectively. Nevertheless, when estimating the continuous missing, the CNN-based structure is limited due to the size of the receptive field. In this paper, the proposed ADRIN exploits incomplete input and historical average to impute the missing traffic data. Unlike the other models that are difficult to interpret and train, our model have more intuitive structure and straight forward training process.

Another related research problem is filling the missing vehicle trajectory data points. Shi *et al.* proposed a Monte-Carlo-based lane marking identification approach [36] to extract the vehicle trajectory data. In [37], the authors proposed a car-following-based (CF-based) vehicle trajectory connection method that can fill missing data points caused by detection errors. However, since these methods focus on the processing of individual vehicle data, the missing traffic data imputation concentrates on the road state under the city

scale. Therefore, the respective solutions cannot be directly adopted for the missing data imputation task.

III. ATTENTION-DRIVEN RECURRENT IMPUTATION NETWORK

In this section, we elaborate on the proposed Attention-Driven Recurrent Imputation Network (ADRIN). To begin, we define the traffic missing data imputation problem in detail. Subsequently, the framework of ADRIN is introduced, including the technical components of ADRIN, and the loss function of ADRIN is defined. Additionally, we extend the model to graph domain by proposing a GCN-based variant of ADRIN, which is presented in the end of this section.

A. PROBLEM DEFINITION

Traffic speed data imputation aims to predict the missing data points with known observed traffic speed data. Given the ground truth road speed data $\mathbf{Y} \in \mathbb{R}^{n\times T}$, we have the observed input feature map with missing data points denoted by $\mathbf{X}^{(?)} = (x_{ij}) \in \mathbb{R}^{n\times T}$, where n represents the number of spatial nodes (e.g., sensor stations or road segments), T represents the number of timesteps in a day, and x_{ij} represents the observed data point of node i at the j-th timestep. To extrude the missing data points in neural computing, we additionally define a mask matrix (also known as binary flag matrix) $\mathbf{M} = (m_{ij}) \in \mathbb{R}^{n\times T}$ as

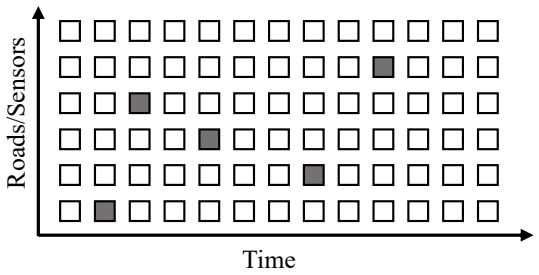
$$m_{ij} = \begin{cases} 0, & \text{if data point } x_{ij} \text{ is recorded;} \\ 1, & \text{if data point } x_{ij} \text{ is missing.} \end{cases} \tag{1}$$

For conveniently understanding, a matrix forms of the incomplete traffic data and the corresponding mask matrix is illustrated below:

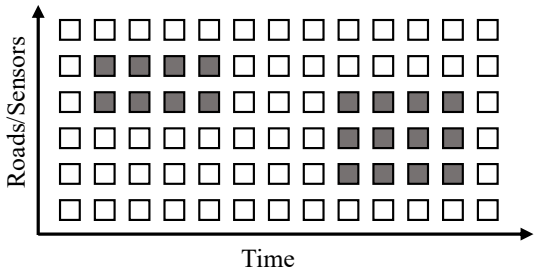
$$\mathbf{X}^{(?)} = \begin{bmatrix} 36 & 40 & ? & 45 & 41 \\ 40 & 44 & 46 & 48 & ? \\ 28 & ? & 40 & 35 & 46 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

It can be observed that data values at positions $(1,3)$,$(2,5)$,$(3,2)$ are missing in the feature matrix with missing data denoted by question marks. Their corresponding binary values are 1 while the values of other observed data points are 0 in their respective mask matrix. The objective of missing data imputation is to restore the missing data points by interpolating synthesized data values. The error between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ should be minimized, where $\hat{\mathbf{Y}}$ is the imputation result.

There are two common patterns of missing data in the literatures, namely, missing completely at random (MCAR) and missing not at random (MNAR) [38], [39], which are illustrated in Figures 2(a) and 2(b), respectively. In MCAR, the distribution of missing data is dispersed and random in time series, while the missing data points appear in continuous time points in MNAR. Comparatively, MNAR is a more challenging problem to solve due to the lack of neighboring information essential for recovering a single missing point. To evaluate the capacity of ADRIN on different missing data with different missing patterns, both patterns are investigated in this work.



(a) Missing completely at random (MCAR).



(b) Missing not at random (MNAR).

FIGURE 2. Diagram of different missing types. The shaded blocks represent the missing data points.

B. ATTENTION-DRIVEN RECURRENT IMPUTATION NETWORK (ADRIN)

Figure 3 depicts the framework of ADRIN. We construct two main data processing flows, as illustrated in the left and right parts of Figure 3 based on the time-series features of traffic speed data. The left flow focuses on extracting typical temporal feature from the incomplete input, i.e., $\mathbf{X}^{(?)} = (\vec{X}_1^{(?)}, ..., \vec{X}_T^{(?)})$. Considering the strong periodical correlations in the traffic data within contiguous days as shown in Figure 1, we additionally construct the right flow, which accepts the historical average data matrix $\mathbf{X}^{(a)} = (\vec{X}_1^{(a)}, ..., \vec{X}_T^{(a)})$ as input by averaging the data of the recent seven days before the incomplete input. Considering that urban sensors are more likely to have similar missing conditions on adjoining days in the real world, so we set the historical data to have the same missing pattern (i.e., missing pattern and missing rate) with the target day. And this impact of historical data will be discussed in Section F. To avoid the influence of extreme value, we first fill the missing position by the average speed of each day in these days. The output of these two flows are two hidden feature matrices, denoted by $\mathbf{H} = (\vec{H}_1, ..., \vec{H}_T)$ and $\mathbf{H}^* = (\vec{H}_1^*, ..., \vec{H}_T^*)$, respectively. The former contains extracted temporal information between timesteps, while the latter incorporates the periodic time series information. In the end, a Fusion Module is devised to aggregate the outputs of two flows (i.e., $\mathbf{H}$ and $\mathbf{H}^*$) and develop the imputed data $\hat{\mathbf{Y}} \in \mathbb{R}^{n\times T}$.

We tailor a few advanced neural network-based approaches according to the time-series traffic speed data and

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2022.3215621

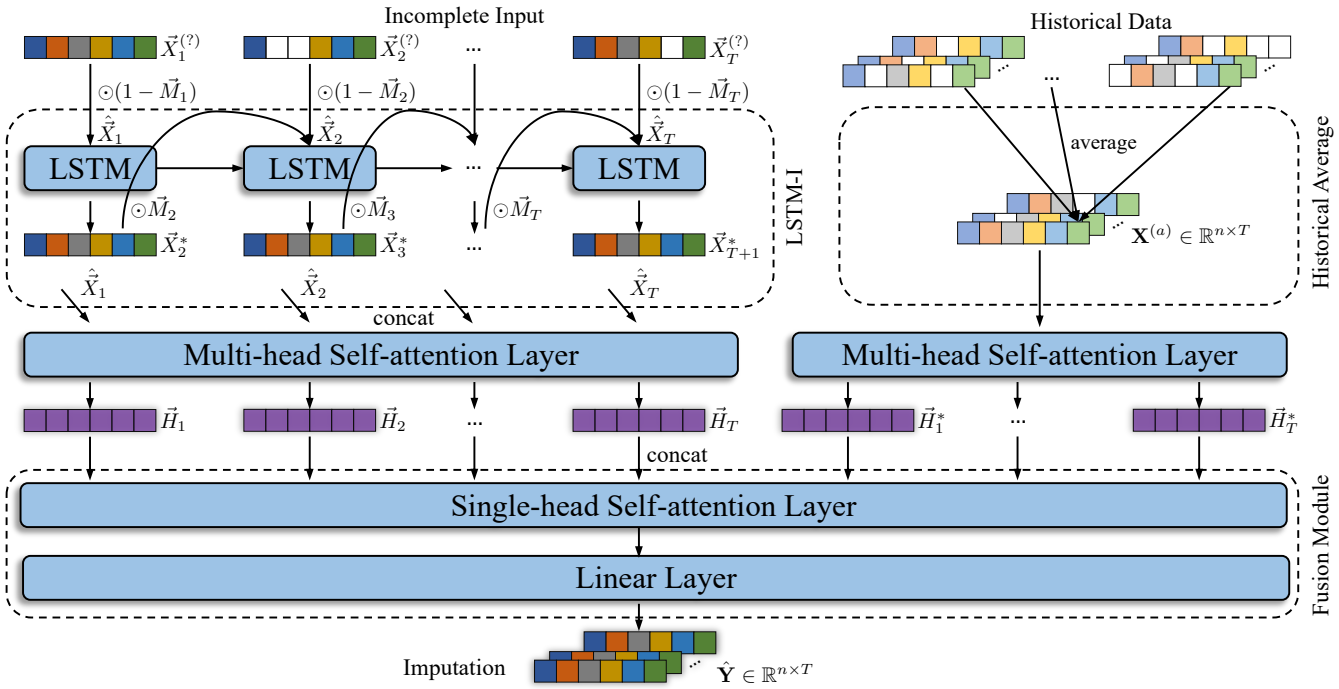


FIGURE 3. **The overall architecture of the proposed ADRIN for traffic data imputation.**

integrated them into ADRIN. Particularly, we first recast the vanilla LSTM and propose an LSTM for Imputation (LSTM-I), which accepts inputs with missing data. The LSTM-I estimates missing values by forward prediction to develop the predicted feature map $\hat{\mathbf{X}}$. Additionally, we employ the multi-head self-attention mechanism [21], which has been demonstrated to be effective in handling time-series problems [16], [40], in the data processing flows as Multi-head Self-attention Layer (MSL) to respectively extract temporal dependency in $\hat{\mathbf{X}}$ and historical features $\mathbf{X}^{(a)}$, and obtain the hidden feature matrices $\mathbf{H}$ and $\mathbf{H}^*$. In the following subsections, we first formulate LSTM-I and multi-head self-attention into layers in the data processing flow as illustrated in Figure 3. Then, we elaborate on the fusion module that combines the time-series and historical feature to generate the imputation.

1) LONG SHORT-TERM MEMORY FOR IMPUTATION LAYER (LSTM-I)

In recent years, LSTM has made numerous achievements in temporal data modeling tasks, particularly in time-series predictions [41], [42]. In comparison to the vanilla RNN, LSTM can prevent gradient explosion during the learning process. Nevertheless, for the majority of existing LSTM networks, the input time-series data must be complete. This requirement, however, is impractical in the investigated traffic data scenarios. Therefore, we refactor the existing LSTM networks and propose LSTM-I dedicated to process the input with missing data points.

As shown in Figure 3, at timestep t, $\vec{X}_t^{(?)}$ denotes the incomplete observation and $\vec{X}_t^*$ is the prediction. When the input $\vec{X}_t$ contains missing data, we combine the $\vec{X}_t^{(?)}$ and $\vec{X}_t^*$ as current input. Specifically, the remodeled LSTM-I uses the predicted value from the previous timestep to impute the incomplete value at the current timestep, which can better utilize the observed values. For each timestep, the LSTM-I employs the estimation and observed values to restore the feature as following.

$$\hat{\vec{X}}_t = \vec{X}_t^* \odot \vec{M}_t + \vec{X}_t^{(?)} \odot (1 - \vec{M}_t), \tag{2}$$

where the $\vec{M}_t \in \mathbf{M}$ represents the missing positions in t-th timestep as defined in A, and $\odot$ denotes the Hadamard product.

For each layer of LSTM-I, the LSTM cell with shared parameters is used for the computation. For the t-th LSTM Cell, the input includes the cell state c_{t-1} at the previous timestep, the hidden feature h_{t-1} and the input $\hat{\vec{X}}_t$. There are three types of gating units in an LSTM cell, namely, input gate i_t, forget gate f_t, and output gate o_t, which are used to decide whether to add/remove information to/from a cell state. These gates adaptively save the input information to the current memory state and develop the hidden feature $h_t \in \mathbb{R}^d$, where d is the dimension size of the hidden feature of the LSTM cell output, which is defined here as the size of LSTM. The entire computation process of LSTM cell is

shown below:

$$i_t = \sigma(\mathbf{W}_i \cdot [\hat{\vec{X}}_t; h_{t-1}] + b_i), \tag{3}$$

$$f_t = \sigma(\mathbf{W}_f \cdot [\hat{\vec{X}}_t; h_{t-1}] + b_f), \tag{4}$$

$$g_t = \tanh(\mathbf{W}_g \cdot [\hat{\vec{X}}_t; h_{t-1}] + b_g), \tag{5}$$

$$o_t = \sigma(\mathbf{W}_o \cdot [\hat{\vec{X}}_t; h_{t-1}] + b_o), \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{7}$$

$$h_t = o_t \odot \tanh(c_t), \tag{8}$$

where g_t is the cell input activation vector, $\mathbf{W}_g \in \mathbb{R}^{d\times 2n}$ and $b_g \in \mathbb{R}^d$ are the weight matrix and bias parameters of the cell, respectively; $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o \in \mathbb{R}^{d\times 2n}$ denote the weight matrices of the input, forget and output gates, respectively; $b_i, b_f, b_o \in \mathbb{R}^d$ are the bias parameters of the corresponding gates; $\mathbb{R}^{d\times 2n}$ and $b_o \in \mathbb{R}^d$ are the weight matrix and bias matrix of the memory cell, respectively; σ represents the sigmoid activation function. Additionally, prediction $\vec{X}^*_{t+1}$ at the next timestep is calculated based on the hidden feature h_t by $\vec{X}^*_{t+1} = \mathbf{W}_h \cdot h_t + \mathbf{b}_h$, where $\mathbf{W}_h \in \mathbb{R}^{n\times d}$ and $b_h \in \mathbb{R}^n$ are the weight and bias matrix, respectively.

Currently, there are some works that apply RNN-based structure to deal with the missing data problem. GRU-D [28] first employs the historical average to fill the missing values and accepts the time-series input. Additionally, the LSTM with a mask can also handle the missing data. However, the former introduces too many noises from the historical data, especially when facing non-recurrent traffic patterns, like traffic accidents. If the missing rate is high, the latter will be hard to learn the temporal dependency in the sparse data within the direct mask operation. In this way, we propose the LSTM-I to avoid the influence of missing position, which can better extract the temporal features.

2) MULTI-HEAD SELF-ATTENTION LAYER (MSL)

Although the enhanced LSTM-I is able to estimate the missing data step-by-step; however, it has limitation to model the dependency of long time series data, especially traffic data often have hundreds of timesteps in one day. Therefore, we apply the multi-head self-attention mechanism that has been demonstrated to be effective in extracting temporal dependencies [40], [43] for further feature extraction of output of LSTM-I $\hat{\mathbf{X}}$ and the historical average $\mathbf{X}^{(a)}$. The multi-head operation can perform the self-attention mechanism in several sub-spaces separately, and then combines all the results after obtaining them. The output matrix with the same shape as the input time series can be developed by recasting with a fully-connected layer.

For self-attention computation, we define the time-series input as $\mathbf{X} \in \mathbb{R}^{T\times n}$ (corresponding to the transposition of $\hat{\mathbf{X}}$ and $\mathbf{X}^{(a)}$). There are three defined components in the computation of the self-attention mechanism, namely, the query matrix $\mathbf{Q} = \mathbf{X} \cdot \mathbf{W}_q \in \mathbb{R}^{T\times T}$, the key matrix $\mathbf{K} = \mathbf{X} \cdot \mathbf{W}_k \in \mathbb{R}^{T\times T}$ and the value matrix $\mathbf{V} = \mathbf{X} \cdot \mathbf{W}_v \in \mathbb{R}^{T\times n}$, where $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{n\times T}$ and $\mathbf{W}_v \in \mathbb{R}^{n\times n}$ are the parameter matrices of the corresponding parts, respectively. The attention matrix $\mathbf{E}$ is subsequently computed by [1]

$$\mathbf{E} = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{T}}) \in \mathbb{R}^{T\times T}. \tag{9}$$

Here, the $\mathrm{softmax}$ function is used to convert the attention matrix into a probability matrix, and the probabilities of all columns sum to 1. Thus, E_{ij} represents the influence of the i-th time point on the j-th time point. The dynamic impact of traffic speed data on different time points on the j-th time point can be captured by multiplying the weights of the j-th column with the value matrix $\mathbf{V}$ and calculating the accumulation. The output of the self-attention mechanism is denoted by $\mathbf{Z}$, which can be computed by

$$\mathbf{Z} = \mathrm{attention}(\mathbf{X}) = \mathbf{E}\mathbf{V} = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{T}})\mathbf{V}. \tag{10}$$

For multi-head self-attention computation, we follow the paradigm in [21] that involves multiple attention mechanisms to compute their respective output $\mathbf{Z}_i, i..H$ separately, where H is the number of attention heads. This allows learning in different attentional subspaces derived by Eq. (10) separately, which is capable of capturing richer feature relationships. Finally, we concatenate all $\mathbf{Z}_i$ inputs to the linear layer to get the final output, which can be formulated as

$$Output = \mathrm{concat}(\mathbf{Z}_1, ..., \mathbf{Z}_h)\mathbf{W}_c, \tag{11}$$

where $Output$ is the final output of MSL that corresponding to the hidden states $\mathbf{H}$ and $\mathbf{H}^*$, and $\mathbf{W}_c$ is the parameter weight of the linear layer.

3) FUSION MODULE

To aggregate the output of the two flows and develop the final imputed data, a Fusion Module is devised that incorporates a Single-head Self-attention Layer and a Linear Layer. The Single-head Self-attention Layer is the one-head version of MSL in which the input is the concatenation of the output of the two flows. After the attentional computation, we use a fully-connected neural network in the Linear Layer to develop the final imputed results, which is defined by

$$\hat{\mathbf{Y}} = \mathrm{attention}(\mathrm{concat}(\mathbf{H}, \mathbf{H}^*))\mathbf{W}_l + \mathbf{b}_l, \tag{12}$$

where $\mathrm{attention}(\cdot)$ represents the attention computation as introduced in Eq. (10), $\mathbf{W}_l$ and $\mathbf{b}_l$ are the parameters of the linear layer, and $\hat{\mathbf{Y}}$ is the final imputed results of ADRIN.

4) LOSS FUNCTION

To restrict the output of LSTM-I to obey the distribution of the speed matrix while accelerating the model convergence, we define respective loss functions for LSTM-I and the final output. Given the ground truth road speed data $\mathbf{Y} \in \mathbb{R}^{n\times T}$

[1] The T in $\mathbf{K}^T$ denotes the transposition in Eq. (9) and (10)

and the imputed output $\hat{\mathbf{Y}}$, we define the masked loss function $\mathcal{L}_1$, which is formulated as

$$\mathcal{L}_1(\mathbf{Y}, \hat{\mathbf{Y}}) = |\mathbf{M} \odot \mathbf{Y} - \mathbf{M} \odot \hat{\mathbf{Y}}|, \tag{13}$$

where $\mathbf{M} \in \mathbb{R}^{n \times T}$ is the mask matrix defined in Section A to indicate the missing data points.

To ensure $\hat{\mathbf{X}}$ (i.e., the output of LSTM-I) is similar with $\mathbf{X}^{(?)}$ (i.e., the input of LSTM-I), and accelerate the convergence speed of LSTM-I, we define the loss function $\mathcal{L}_2$ as

$$\mathcal{L}_2(\mathbf{Y}, \hat{\mathbf{X}}) = |\mathbf{M} \odot \mathbf{Y} - \mathbf{M} \odot \hat{\mathbf{X}}|. \tag{14}$$

Combining the above two sub-loss functions, we have the final loss function $\mathcal{L}$, which is formulated as

$$\mathcal{L} = \mathcal{L}_1(\mathbf{Y}, \hat{\mathbf{Y}}) + \mathcal{L}_2(\mathbf{Y}, \hat{\mathbf{X}}). \tag{15}$$

C. GRAPH CONVOLUTIONAL NETWORK-EMPOWERED ADRIN

Plenty of works [44], [20] has shown that spatial information is essential for road-based tasks like traffic state prediction and traffic data imputation. However, over-reliance on spatial information will limit the application scenarios of the proposed model. For example, geographic information of some roads or sensors may be unretrievable due to security or privacy reasons. In this way, we only propose a vanilla extension version of ADRIN, named GCN-ADRIN, which incorporates the ADRIN and graph conventional network (GCN) [45]. Except for the merit as introduced in ADRIN, the GCN-ADRIN is able to estimate the missing considering the spatial information.

The road network can be represented by a directional graph $G = (V, E)$, where V is the set of road segments (i.e., nodes) and E is the set of road intersections (i.e., edges). The adjacency matrix $A = \{I_{ij}\}$ is generated by a thresholded Gaussian kernel method:

$$I_{ij} = \begin{cases} 1, \text{ when } i \neq j \text{ and } \exp\left(-\dfrac{\text{dist}(v_i, v_j)}{\kappa^2}\right) \geqslant \mu \\ 0, \text{ otherwise.} \end{cases}, \tag{16}$$

where I_{ij} denotes the connectivity between nodes v_i and v_j, which depends on their Euclidean space distance defined by $\text{dist}(v_i, v_j)$; μ and κ are the user-defined parameters which are to control the sparsity of graph, and their values are set in accordance with [46].

Figure 4 depicts the structure of the extension part that contains two vanilla GCN layers. To avoid overfitting and debilitating performance, we utilize the skip-connection [47] to calculate the final output of the extension part.

$$\hat{\mathbf{X}}^{(out)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widehat{A} \widetilde{D}^{-\frac{1}{2}} \hat{\mathbf{X}} \mathbf{W}_g\right) + \hat{\mathbf{X}}, \tag{17}$$

where $\widehat{A} = A + I_n$ denotes the sum of adjacent matrix and self-connection, I_n is the identity matrix, $\widetilde{D} = \sum_j \widehat{A}_{ij}$ is the degree matrix, $\mathbf{W}_g \in \mathbb{R}^{n \times n}$ is the weighted matrix, $\hat{\mathbf{X}}^{(out)}$ is the output of this module, and $\sigma(.)$ is the sigmoid function.

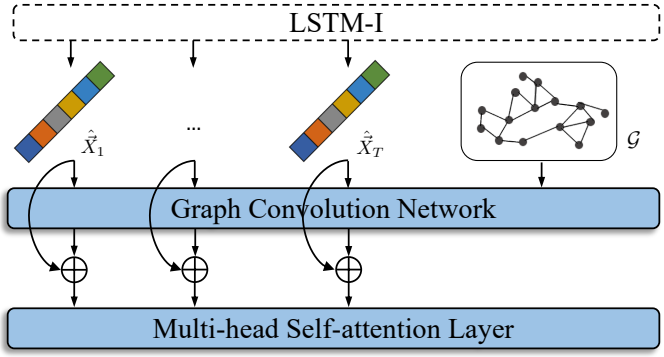


FIGURE 4. The architecture of GCN module in ADRIN.

We consider extracting the spatial dependency from both left and right halves respectively in Figure 3. As shown in Figure 4, the module that contains GCN and skip-connection is between the LSTM-I and the MSL. Similarly, the module with the same structure is applied to the historical average part to learn the spatial information between the input and MSL. Currently, there are a variety of advanced graph neural network structures applied in the traffic domain, and the senior design will be discussed in future work.

IV. CASE STUDIES

In this work, we propose ADRIN for missing traffic data imputation. To evaluate and explore its performance, we conduct a series of comprehensive case studies on three real-world traffic speed datasets. This section first introduces the experimental setup. Subsequently, we compare the imputation accuracy of ADRIN with existing state-of-the-art algorithms and visualize the imputation effects. Ablation experiments and hyper-parameter tests are carried out to demonstrate and elaborate on the necessity of various parts in the model. Additionally, we investigate the performance of models on incomplete data with different sampling noise.

A. EXPERIMENT CONFIGURATION

1) DATASET

In this work, we employ three real-world traffic speed datasets, namely, NavInfo-Beijing (BJ)[2], PeMS: District 5 (PeMSD5)[3], and Hong Kong Traffic Speed Map (HK)[4]. Specifically, the BJ dataset is provided by NavInfo Traffic Index Platform, which contains the average speed of 1368 roads in Beijing from 00:00 Jan. 1, 2019 to 23:55 Jun. 30, 2019. The sampling interval of records is 5 minutes. To minimize the impact of missing data while retaining the dataset complexity, we only use the road speed data with an overall missing data rate of less than 5% for the experiment, i.e., a total of 168 roads. According with the literature [46], [48], we utilise linear interpolation to complement the

[2]http://www.nitrafficindex.com
[3]https://PeMS.dot.ca.gov/
[4]https://data.gov.hk/en-data/dataset/hk-td-sm_1-traffic-speed-map

missing traffic data, record their positions, and remove the interpolation during the evaluating stage.

The PeMSD5 dataset contains traffic speed data collected from Caltrans Performance Measurement System. This dataset includes 144 sensor stations in District 5 of California, and the recording duration is from 00:00 Jan. 1, 2013 to 23:55 Jun. 30, 2013. It is worth mentioning that the records in PeMSD5 are collected from sensors in highways, which differ from the urban roads in BJ. The HK dataset includes the average road speed of arterial roads in Hong Kong from 00:00 Mar. 10, 2021 to 23:50 Jul. 31, 2021. We note that the records from outbacks like Tuen Mun and Shatin are static in the long term. Hence, we employ the road data in Hong Kong island that consists of 84 roads. Particularly, the actual sampling interval is 2 minutes in HK dataset; however, considering the large computational consumption and the occurrence of original missing in dataset, the time interval of traffic speed data is set to 10 minutes. One to note that the BJ an HK are from dual-loop detectors, and the D5 is from a single loop detectors. A summary of these datasets is presented in Table 1. Figure 5 depicts the sampling locations of roads or sensors in the three datasets[5].

TABLE 1. Summary of BJ, PeMSD5, and HK Datasets

	BJ	PeMSD5	HK
No. roads	168	144	84
No. days	181	181	144
Time interval	5min	5min	10min
Avg. speed	36.70 km/h	54.47 mi/h	51.24 km/h
Std. speed	11.52 km/h	7.32 mi/h	20.19 km/h

2) CONFIGURATIONS

In all case studies, we employ the Z-score normalization to preprocess the data. For cross-validation, we follow the prior work [39], [49] and split the datasets into two non-overlapping subsets based on the chronological order, namely, the training and test sets: the first 80% of all samples are training data, while the remaining 20% are the test data. In addition, a data augmentation method is applied during the training stage: for each sample $\mathbf{Y}$ in the training set, we generate the missing input $\mathbf{X}^{(?)}$ ten times randomly. We use Adam [50] as the optimizer with an initial learning rate of 0.001. The batch size is 20, and the number of training epoch is set to 200. The number of heads H in the multi-head self-attention mechanism is set to 8, and the hidden layer size d is set to 168; these two parameters will be discussed in the following section. PyTorch is used to conduct the experiments, and the hardware configuration includes nVidia RTX 2080Ti GPUs and a Xeon Silver 4210 CPU.

[5]To better show the collection locations, we have simplified the roads in the BJ and HK datasets to a point representation.

3) METRIC

Following the previous works [6], [39] on traffic data imputation, we adapt mean absolute error (MAE) and mean absolute percentage error (MAPE) as the metrics for this experiment, which are defines as

$$\mathrm{MAE} = \frac{1}{c}\sum_{j=1}^{T}\sum_{i=1}^{n}\left|m_{ij}(y_{ij}-\hat{y}_{ij})\right|, \tag{18}$$

$$\mathrm{MAPE} = \frac{100\%}{c}\sum_{j=1}^{T}\sum_{i=1}^{n}\left|\frac{m_{ij}(y_{ij}-\hat{y}_{ij})}{y_{ij}}\right|, \tag{19}$$

where m_{ij} indicates the traffic data in same position is missing or not as introduced in A, $c = \sum_{j=1}^{T}\sum_{i=1}^{n} m_{ij}$ represents the number of corrupted records, $y_{ij} \in \mathbf{Y}$ is the ground truth, and $\hat{y}_{ij} \in \hat{\mathbf{Y}}$ is the imputation value.

B. ACCURACY OF IMPUTATION

To assess the model comprehensively, we conducted experiments considering the two missing patterns, i.e., MCAR and MNAR, in this case study. Additionally, to verify the effectiveness of model in a variety of scenarios, we compare ADRIN to existing methods considering a wide range of missing rates from 10% to 90%. The applied baseline approaches are selected from a variety of imputation methods as reviewed in Section II, which are the current state-of-the-art or most widely adopted ones for missing data imputation:

- *Historical Average (HA)* [23]: HA takes the complete data of the most recent seven days and averages the corresponding timestep in a day for each road segment to fill in the missing values.
- *Bayesian Gaussian CP decomposition (BGCP)* [13]: BGCP is a Bayesian tensor factorization model that employs Markov chain Monte Carlo to model the latent factor (i.e., low-rank structure).
- *Bayesian Temporal Matrix Factorization (BTMF)* [51]: BTMF employs a Gaussian vector autoregressive process to model the temporal dependence to impute the time series data.
- *Parallel Data and Generative Adversarial Networks for Imputation (PGAN)* [39]: PGAN a GAN-based data imputation approach for missing traffic data, and both the generator and discriminator are made of linear layers.
- *Graph Attention Convolutional Network (GACN)* [20]: GACN incorporates convolutional neural network and graph attention network to estimate the missing values and follows an encoder-decoder structure.
- *Bidirectional Recurrent Imputation (BRITS)* [17]: BRITS accepts the missing timing data into two RNNs in forward and reverse directions, respectively, and combines the outputs of the two RNNs to compensate for the missing time-series data.

Among these baselines, the matrix factorization-based methods (e.g., TRMF and BTMF) must strictly ensure that

(a) Road segment locations in BJ dataset. (b) Sensor locations in PeMSD5 dataset. (c) Road segment locations in HK dataset.

FIGURE 5. Three real-world datasets investigated in experiments.

the input format shape is $days \times time \times road$. For comparison purposes, we uniformly set the inputs of deep learning methods such as PGAN, BRITS, GACN, ADRIN, and GCN-ADRIN to be the missing data of one day, i.e., $time \times road$. Additionally, the input of matrix factorization-based methods is the concatenation of complete training data and incomplete testing data. Furthermore, due to it is hard to ensure the convergence of PGAN, we set the learning rates of generator and discriminator among $\{0.00001, 0.0001, 0.001, 0.01\}$ and apply grid search to obtain the best result. The hyper-parameters of the other models remain unaltered.

Tables 2, 3, and 4 summarize the imputation result of MCAR and MNAR, respectively, with the missing rate between 10% and 90% on the three datasets. The experimental results demonstrate that the proposed ADRIN and GCN-ADRIN outperforms all other baselines by achieving the lowest MAE and MAPE values under most scenarios. The results are outstanding on complicated datasets like urban roads speed data BJ and HK, while on the simpler highway speed dataset PeMSD5, ADRIN can achieve results close to SOTA on MCAR. This is due to the superiority of the proposed model to extract time-series dependency from missing input and historical features of ADRIN. In particular, statistical learning-based method, such as HA produce significantly worse imputation roads speed values than ADRIN. This is because such approach capture only the sample distribution within the dataset to complement the missing value but ignore the temporal correlation, which make the inferior performance.

Considering the compared deep learning-based methods, the proposed ADRIN outperforms the state-of-the-art PGAN, GACN, and BRITS. Due to the superior ability to model time-series dependencies from the missing input and the historical data separately, ADRIN achieves a significant improvement over the other approaches when the missing rate is low, regardless of whether the missing data is completely random or not. For example, compared to other deep learning-based methods on HK dataset, e.g., the BRITS who won third place, in both cases, ADRIN reduces the MAPE by 9.82% and 8.53% at 10% missing. At the same time, at 90% missing, the MAPE results in reductions of 5.39% and 7.30%, respectively. The result demonstrates that ADRIN can achieve more significant gains when the missing rate is low, which is contributed by incorporating more complete history features. The same phenomena can be observed on the D5, HK dataset. In addition, since the PGAN use only linear modules to model features, they can only impute acceptable results on the PeMSD5 dataset with simple data distributions. However, the proposed ADRIN, which additionally takes into account the temporal dependency, can cope with both missing data of urban roads and highways.

Unlike matrix factorization-based methods, deep learning ones have more advantages for modeling complicated data such as the urban road speed. However, deep learning models are highly dependent on the quality of data, and high missing rates make it difficult to capture the temporal feature correlations. For instance, in the case of MCAR and 90% missing on PeMSD5 dataset, ADRIN achieves an inferior performance due to the data distribution of highway speed is simpler than urban roads. Additionally, the results of matrix factorization-based approach are much stable across all missing rates. This is because that these methods estimate the missing values from the overall data distribution. For example, the absolute differences of ADRIN's MAPE results in all missing rate cases are 1.07% (MCAR) and 0.57% (MNAR) on BJ dataset, but the differences of BGCP are 0.21% (MCAR) and 0.26% (MNAR). However, contributed by the outstanding ability for modeling the temporal dependency, ADRIN outperforms all other models on the complicated urban datasets (BJ and HK) with various missing cases.

Comparing the imputation results between MCAR and MNAR, it is clear that all approaches have lower MAPE under MCAR than MNAR. Due to the continuous large batch of missing blocks, it is challenging to model the feature dependency of data in MNAR, leading to the degraded performance. However, on the BJ and HK datasets, ADRIN can achieve more significant improvement compared with GACN, BRITS, and PGAN in MNAR than MCAR,

TABLE 2. Performance Comparison (in MAE/MAPE(%)) for Imputation Tasks on BJ Dataset

Pattern	Missing Rate	Methods							
		HA	BGCP	BTMF	PGAN	GCAN	BRITS	ADRIN	GCN-ADRIN
MCAR	10%	3.72/11.90	3.20/9.98	3.12/9.64	6.22/21.41	3.15/9.93	3.04/9.37	**2.78/8.45**	2.99/8.63
	30%	3.70/11.94	3.20/10.04	3.13/9.72	6.22/21.34	3.14/9.93	3.05/9.42	**2.80/8.59**	2.94/8.67
	50%	3.70/11.90	3.22/10.07	3.15/9.76	6.22/21.35	3.15/9.94	3.10/9.58	**2.87/8.85**	2.95/8.93
	70%	3.71/11.92	3.22/10.09	3.19/9.89	6.22/21.37	3.15/9.99	3.13/9.70	**2.95**/9.15	2.99/**9.06**
	90%	3.70/11.92	3.25/10.19	3.28/10.18	6.23/21.38	3.16/9.98	3.24/10.06	3.05/9.52	**3.03/9.41**
MNAR	10%	3.80/12.24	3.28/10.24	3.26/10.15	6.70/23.74	3.20/10.31	3.28/10.20	**3.03**/9.33	3.03/**9.27**
	30%	3.81/12.36	3.32/10.48	3.33/10.46	6.66/22.90	3.25/10.44	3.30/10.35	**3.08/9.46**	3.10/9.49
	50%	3.81/12.22	3.31/10.36	3.34/10.44	6.63/22.56	3.24/10.34	3.30/10.34	**3.11/9.53**	3.13/9.59
	70%	3.81/12.31	3.32/10.46	3.39/10.66	6.58/22.48	3.21/10.28	3.35/10.54	3.11/9.67	**3.10/9.63**
	90%	3.78/12.23	3.30/10.45	3.38/10.63	6.53/22.43	3.23/10.34	3.39/10.68	**3.15/9.90**	3.16/9.92
Time Consumption :		-	0.15h	2.61h	2.86h	1.24h	5.51h	2.97h	3.16h

TABLE 3. Performance Comparison (in MAE/MAPE(%)) for Imputation Tasks on PeMSD5 Dataset

Pattern	Missing Rate	Methods							
		HA	BGCP	BTMF	PGAN	GCAN	BRITS	ADRIN	GCN-ADRIN
MCAR	10%	2.56/7.25	2.04/5.08	1.65/4.04	3.46/10.51	2.14/6.09	1.15/2.66	1.12/2.57	**1.12/2.51**
	30%	2.53/7.13	2.09/5.19	1.66/4.03	3.47/10.59	2.11/5.89	1.22/2.80	1.18/2.77	**1.13/2.62**
	50%	2.53/7.11	2.11/5.24	1.66/4.02	3.49/10.63	2.14/6.05	1.35/3.17	1.32/3.15	**1.30/3.06**
	70%	2.53/7.12	2.17/5.37	1.67/4.06	3.53/10.69	2.14/6.01	1.58/3.80	1.50/3.63	**1.48/3.55**
	90%	2.53/7.12	2.22/5.52	**1.76/4.27**	3.60/10.83	2.15/6.09	1.93/4.96	1.80/4.53	1.78/4.44
MNAR	10%	2.87/8.13	2.34/5.98	2.49/6.21	4.15/13.79	2.42/6.81	2.76/7.33	2.04/5.23	**1.99/5.07**
	30%	2.95/8.67	2.41/6.90	3.92/9.51	4.44/14.34	2.49/7.73	2.73/7.68	2.34/5.98	**2.33/5.93**
	50%	3.01/9.09	2.70/7.33	3.45/9.14	4.17/13.16	2.42/6.99	2.67/7.32	2.25/6.47	**2.19/6.26**
	70%	2.91/8.62	2.66/7.10	3.42/9.24	4.12/13.01	2.36/6.84	2.60/7.26	2.28/6.53	**2.21/6.32**
	90%	2.86/8.41	2.62/6.86	3.57/9.49	4.14/13.14	2.35/6.77	2.77/7.59	2.26/6.51	**2.21/6.34**
Time Consumption :		-	0.06h	0.60h	2.40h	0.92h	4.83h	1.89h	1.92h

TABLE 4. Performance Comparison (in MAE/MAPE(%)) for Imputation Tasks on HK Dataset

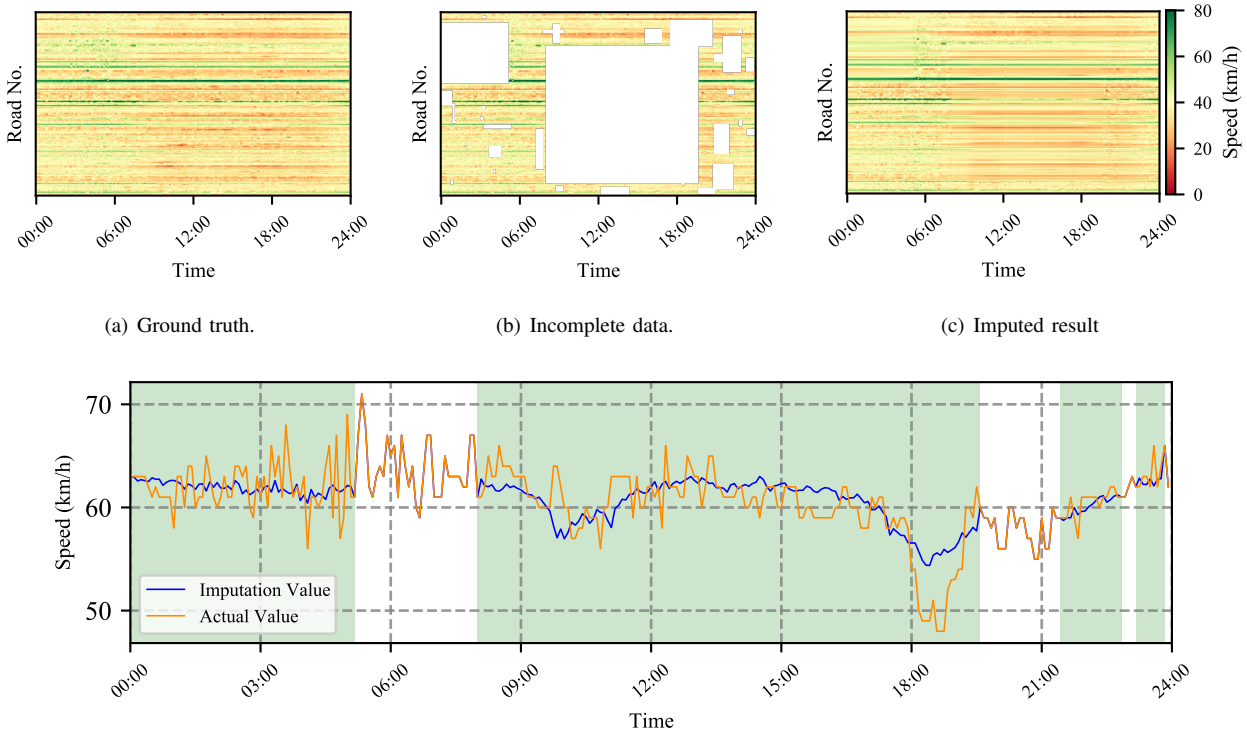
Pattern	Missing Rate	Methods							
		HA	BGCP	BTMF	PGAN	GCAN	BRITS	ADRIN	GCN-ADRIN
MCAR	10%	4.63/15.18	4.08/13.02	2.91/9.14	10.43/68.58	6.45/22.48	2.14/6.23	1.60/4.59	**1.58/4.44**
	30%	4.61/15.15	4.09/12.92	3.07/9.76	10.43/68.52	6.92/23.39	2.19/6.42	1.69/4.80	**1.66/4.74**
	50%	4.59/15.14	4.09/13.15	3.16/10.10	10.48/69.53	6.89/23.81	2.40/7.13	1.95/5.58	**1.85/5.47**
	70%	4.61/15.07	4.09/12.96	3.32/10.55	10.50/69.75	7.07/24.21	2.83/8.48	2.34/6.73	**2.23/6.63**
	90%	4.60/15.11	4.14/13.34	3.72/11.77	10.58/69.68	7.56/24.90	3.90/12.29	3.15/9.51	**3.04/9.31**
MNAR	10%	4.64/15.47	4.07/13.45	3.28/11.24	11.61/86.02	6.16/20.30	2.73/9.27	1.99/5.84	**1.95/5.57**
	30%	4.66/15.93	4.16/14.04	3.42/11.77	11.61/86.75	6.73/25.06	2.88/8.88	2.29/7.24	**2.25/7.20**
	50%	4.70/16.18	4.24/14.29	3.70/12.87	11.50/83.61	6.55/25.35	3.14/10.60	2.54/7.89	**2.50/7.63**
	70%	4.66/15.72	4.13/13.72	3.84/13.48	11.32/80.93	6.83/24.50	3.30/10.90	2.79/8.44	**2.78/8.36**
	90%	4.67/15.72	4.25/14.25	3.99/13.38	11.32/79.99	7.16/25.31	3.48/11.52	**2.96/9.10**	2.98/9.19
Time Consumption :		-	0.08h	1.07h	1.24h	0.52h	2.42h	0.96h	1.01h

contributed by the aid of historical information. Additionally, resembling the performance of ADRIN and GCN-ADRIN, the latter achieves better performance on the D5 and HK datasets with the aid of GCN. The difference is because the distance between road segments is small as shown in Figure 5(b) and 5(c), and the spatial information is of great significance on the two datasets. On the contrary, the road segments in BJ are far from each other. The vanilla GCN structure can not fully extract the spatial dependency, resulting in no noticeable improvement.

To better evaluate the efficiency of different methods, we count the average time overhead of the above experiments, and the results are shown in Table 2, 3, and 4. Among these methods, BGCP achieves the shortest computation time on all datasets, thanks to that BGCP only employs Markov chains to factorize the residual data and complement it. Additionally, among the deep learning methods, our proposed ADRIN and GCN-ADRIN have a significant improvement in computation time over BRITS thanks to their end-to-end structure. In addition, the computation time is less than that of PGAN, which requires pre-training. GCAN achieves a lower time overhead due to feature extraction only from a spatial perspective and its simple structure, but its simple structure leads to a lower imputation accuracy than other methods.

Finally, to illustrate the imputation of ADRIN more clearly, we visualize the ground truth data, the missing input and the imputation results on June 30, 2019 on dataset BJ in Figures 6(a), 6(b), and 6(c), respectively. It is noticeable that the model prefers to describe the traffic speed using

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2022.3215621



(a) Ground truth. (b) Incomplete data. (c) Imputed result

(d) The imputed result of West Zhong Shan Avenue. Green shaded areas indicate that traffic data in the corresponding period are missing. The no-shade areas express the data is observed during these periods.

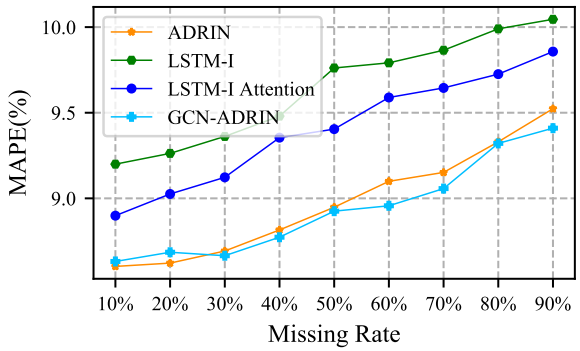
FIGURE 6. The visualization road speed on 2019-06-30 in Beijing, including the ground true value, the incomplete observation, imputed results, and an individual road speed.

smooth curves for different road segments rather than to more dispersed actual values. Furthermore, in the presence of consecutive missing values, ADRIN does an outstanding job of estimating the unobserved values. Additionally, we choose the No. 45 road, West Zhong Shan Avenue, which severely deficits observation values, as shown in Figure 6(d). In Figure 6(d), the shaded area indicates that the data are missing period, the orange and black curves represent the actual and imputed speeds, respectively. While ADRIN's imputation results are smoother than the real speed variation, they can accurately capture changing road speed trends. For example, even when data are missing, ADRIN can accurately reflect the daily trend of increasing and then decreasing road speed between 9:00 and 18:00.

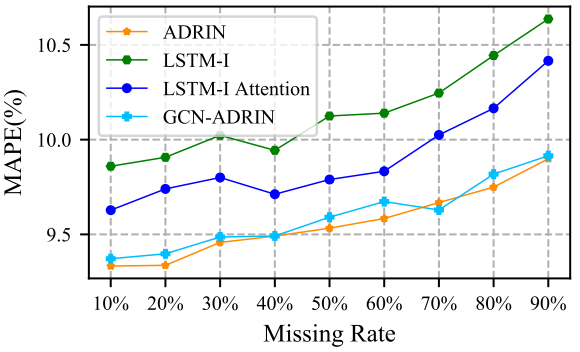
C. ABLATION TEST

In this subsection, we carry out comprehensive ablation tests to assess the contribution of each sub-networks to the overall model performance. To maintain objectivity, The training and testing configurations used in these case studies are identical to those described in Section 2. Specifically, the variants LSTM-I Attention is defined by: The left half as shown in Figure 3 that incorporates LSTM-I and MSL for modeling the temporal dependence within the incomplete input. Besides, we also compare the performance of LSTM-I and GCN-ADRIN.

Figure 7 shows the MAPE result of the two sub-networks, ADRIN, and GCN-ADRIN evaluated with different missing types on the BJ dataset. The comparison leads to the following conclusions: To begin, the proposed ADRIN achieves the best performance across most missing rates and patterns. This observation conforms to the intuition that ADRIN models the temporal dependency from both the missing data and the historical average, resulting in improved information extraction and imputation result. Especially, thanks to the aid of historical data, the improvement of the imputation effects are more evident in high missing rates. Furthermore, unlike the LSTM attention network, ADRIN utilizes the recent five-day traffic data to enhance the imputation. Comparing the two result curves, ADRIN indicates that historical data has a non-neglectable positive effect on the imputation. Additionally, the better result of LSTM-I Attention than LSTM-I shows the superior ability of the multi-head self-attention mechanism to model time-series features. Besides, with the help of spatial information, GCN-ADRIN has achieved improvements in some cases like MCAR-60%, MCAR-70%, etc. However, the graph is generated based on the distance between roads as defined in Eq. (16). It can not express the connectivity of road network while applied on the BJ



(a) MCAR.



(b) MNAR.

FIGURE 7. The MAPE results of ablation tests on BJ dataset.

dataset whose roads are far from each other, leading to the unremarkable improvement of imputation. We will design an advanced graph structure to address this issue in future work.

D. HYPER-PARAMETERS

TABLE 5. ADRIN Hyper-Parameter Models And Imputation Accuracy (MCAR)

Model	Hidden size	Heads number	MAE	MAPE(%)
ADRIN	84	4	2.90	8.99
		8	2.88	8.94
		12	2.88	8.95
	168	4	2.88	8.90
		8	2.87	8.85
		12	2.87	8.86
	336	4	2.88	8.89
		8	**2.85**	**8.83**
		12	2.85	8.83

Proper selection of hyper-parameters, such as the size of the hidden layer, the number of hidden layers, and number of heads, is critical in determining the model's performance in the context of deep learning. In this subsection, we study the

TABLE 6. ADRIN Hyper-Parameter Models And Imputation Accuracy (MNAR)

Model	Hidden size	Heads number	MAE	MAPE(%)
ADRIN	84	4	3.15	9.58
		8	3.12	9.55
		12	3.12	9.54
	168	4	3.12	9.55
		8	3.11	9.53
		12	3.11	9.52
	336	4	3.13	9.51
		8	**3.10**	9.51
		12	3.11	**9.49**

performance of ADRIN across a range of hyper-parameters and attempt to determine the optimal neural network architecture. In particular, we employ the hidden size d of LSTM-I among $\{84, 168, 336\}$[6], and the number of heads H among $\{4, 8, 12\}$[7] to test the imputation performance on BJ with 50% missing rate on MCAR and MNAR.

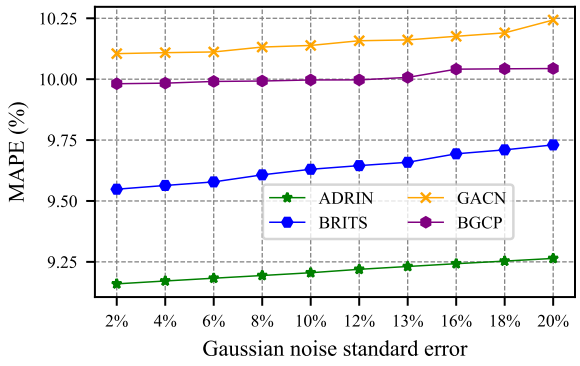
Tables 5 and 6 demonstrate a variety of well-performing structures and their imputation results. From the results, we can draw the following conclusions. First, increasing the number of hidden states in the model can improve performance (compare the performance of ADRIN with $d = 84$ and ADRIN with $d = 168$). This is consistent with the widely hold belief that increasing the number of neural network layers increases the efficacy of learning latent data features. However, the excessive number of neurons can degrade the performance due to over-fitting. Second, involving more attention heads can slightly improve the system performance. This is consistent with the concept of multi-head self-attention mechanism, which states that more features can be learned through the multi sub-space learning process. However, too many heads may lead to excessive consumption of computing resources.

E. DATA NOISE

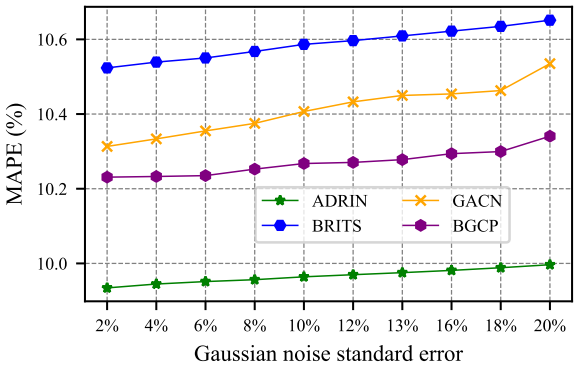
In the previous case studies, we utilized the ground truth of three real-world traffic speed datasets to evaluate and explore the proposed model. However, noise during the sampling process inevitably influences the data collected. In this subsection, we manually introduce various noises to the input of the model to demonstrate their robustness on the BJ dataset. Specifically, following the previous work [52], we generate the Gaussian noise that follows the mean equals to 1 and the standard error among $\{2\%, 4\%, ..., 20\%\}$ and multiply original input by the noise. It means the average deviations are 1%, 2%, ..., and 10% of the ground truth, respectively.

[6]Considering the BJ dataset contains 168 road segments.
[7]Referring to [21].

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2022.3215621

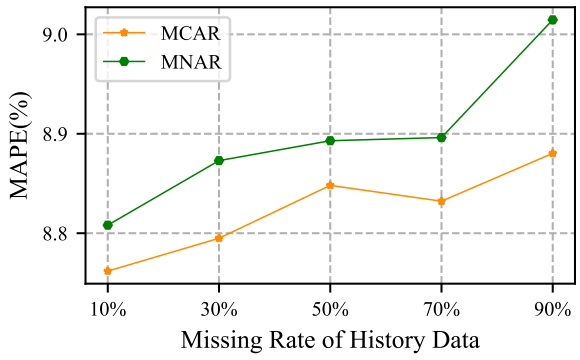


(a) MCAR.

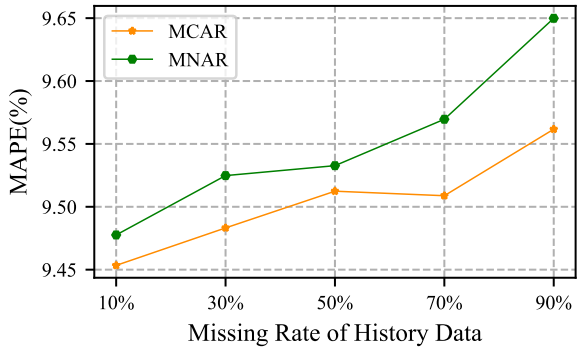


(b) MNAR.

FIGURE 8. The MAPE results of noise experiment on BJ dataset.



(a) MCAR-50%.



(b) MNAR-50%.

FIGURE 9. The MAPE results of MCAR-50% and MNAR-50% with various historical data missing pattern on BJ dataset.

Figure 8 shows the imputation performance of four models who achieved outstanding results in previous case studies, including BGCP, GACN, BRITS, and the proposed ADRIN. The results indicate that increasing noise level generally leads to larger MAPE, while the influenced data distribution patterns are harder to learn. Additionally, because the performance of deep learning-based methods highly depends on the quality of data, it is challenging to learn the data dependency with the intensive noise. In contrast, the matrix factorization-based method learns from the low-rank space, and noise is attenuated in the process of factorization, making it better noise robustness.

F. THE IMPACT OF HISTORICAL DATA

In Section C, through the ablation test, we can see that the historical data module has a significant role in the missing data imputation. However, the influence of historical data with different missing patterns and rates still needs to be explored. Therefore, in this part, we set different missing rates, from 10% to 90%, and missing patterns, including MNAR and MCAR, of historical data to observe their imputation performance for our proposed ADRIN model at 50% of MCAR and MNAR, respectively.

Figure 9 demonstrates the influence of different historical data missing cases on the imputation performance. One to note that the vertical axis indicates the missing rate of historical data, the orange line indicates that the missing pattern of historical data is MCAR, and the green line means the MNAR. As can be observed from the above figure, for both MCAR-50% and MNAR-50%, when the missing pattern of historical data is MCAR, the overall imputation accuracy is better than MNAR. This is due to the fact that in MCAR, the data are more dispersed, and more information is available for the data matrix than in MNAR, which is more evident in the case of a high missing rate. In addition, along with the increase of the historical data missing rate, the available information decreases, the improvement of the imputation performance is smaller, and the error of imputation, i.e., MAPE, is larger, which is intuitive. As we can see from the presentation of the completion results, the historical data with different missing cases have a positive impact on the imputation effect. However, at the same time, the enhancement effect on the imputation effect is different according to the distribution and proportion of information.

V. CONCLUSION

In this paper, we propose a novel network structure named Attention-Driven Recurrent Imputation Network (ADRIN)

for the missing traffic data imputation problem. Compared with existing deep learning-based imputation approaches, ADRIN exploits the unique periodicity and volatility of traffic data to extract features and complement missing values from the incomplete data input and historical average. In ADRIN, we first propose a Long Short-Term Memory for Imputation (LSTM-I) model to process the missing inputs. Following that, we apply a multi-head self-attention mechanism to extract temporal features from the historical averages and LSTM-I outputs, respectively. The outputs are passed through self-attention and fully connected neural network layers to fuse and obtain the imputed results. Based on the ADRIN, we devise spatial information enhanced version named GCN-ADRIN, which can extract the spatial-temporal dependency. In addition, we design a tailored loss function for each module separately during the training process.

To evaluate the performance of the proposed ADRIN, we conducted comprehensive case studies on three real-world traffic speed datasets. Compared with baseline methods, ADRIN achieves superior imputation performance for both MCAR and MNAR with varying missing rates. Meanwhile, the effectiveness of different modules in ADRIN are investigated and analyzed using ablation experiments and parameters tests. The results show that historical information is crucial for the imputation of traffic data. The performance of GCN-ADRIN demonstrates that the outstanding scalability of ADRIN. In the future, we set about to integrate more advanced deep learning and data mining approaches into ADRIN to better incorporate the topological relationships between various road segments. Furthermore, on the time axis, we will explore the utilization methods in terms of different periods' data to further improve the imputation performance.

REFERENCES

[1] X. Chen, J. Yang, and L. Sun, "A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102673, 2020.

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.

[3] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 372–385, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X1831043X

[4] D. Zhang, F. Xiao, M. Shen, and S. Zhong, "Dneat: A novel dynamic node-edge attention network for origin-destination demand prediction," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102851, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307518

[5] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018.

[6] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2933–2943, 2019.

[7] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1762–1771, 2016.

[8] H. Tan, Y. Wu, B. Cheng, W. Wang, and B. Ran, "Robust missing traffic flow imputation considering nonnegativity and road capacity," *Mathematical Problems in Engineering*, vol. 2014, p. 763469, 2014. [Online]. Available: https://doi.org/10.1155/2014/763469

[9] D. Ni, J. D. Leonard, A. Guin, and C. Feng, "Multiple imputation scheme for overcoming the missing values and variability issues in its data," *Journal of transportation engineering*, vol. 131, no. 12, pp. 931–938, 2005.

[10] M. R. Malarvizhi and A. S. Thanamani, "K-nearest neighbor in missing data imputation," *International Journal of Engineering Research and Development*, vol. 5, no. 1, pp. 5–7, 2012.

[11] W. F. Velicer and S. M. Colby, "A comparison of missing-data procedures for arima time-series analysis," *Educational and Psychological Measurement*, vol. 65, no. 4, pp. 596–615, 2005.

[12] F. Rodrigues, K. Henrickson, and F. C. Pereira, "Multi-output gaussian processes for crowdsourced traffic data imputation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 594–603, 2019.

[13] X. Chen, Z. He, and L. Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 73–84, 2019.

[14] A. Sportisse, C. Boyer, and J. Josse, "Estimation and imputation in probabilistic principal component analysis with missing not at random data," *arXiv preprint arXiv:1906.02493*, 2019.

[15] X. Luo, M. Zhou, H. Leung, Y. Xia, Q. Zhu, Z. You, and S. Li, "An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 333–343, 2016.

[16] W. Zhang, P. Zhang, Y. Yu, X. Li, S. A. Biancardo, and J. Zhang, "Missing data repairs for traffic flow with self-attention generative adversarial imputation net," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[17] W. Cao, D. Wang, J. Li, H. Zhou, Y. Li, and L. Li, "Brits: Bidirectional recurrent imputation for time series," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Curran Associates Inc., 2018, pp. 6776–6786.

[18] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E²gan: End-to-end generative adversarial network for multivariate time series imputation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, Conference Proceedings, pp. 3094–3100. [Online]. Available: https://doi.org/10.24963/ijcai.2019/429

[19] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[20] Y. Ye, S. Zhang, and J. J. Q. Yu, "Spatial-temporal traffic data imputation via graph attention convolutional network," in *Artificial Neural Networks and Machine Learning – ICANN 2021*, I. Farkaš, P. Masulli, S. Otte, and S. Wermter, Eds. Cham: Springer International Publishing, 2021, pp. 241–252.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[22] J. Twisk and W. de Vente, "Attrition in longitudinal studies: How to deal with missing data," *Journal of Clinical Epidemiology*, vol. 55, no. 4, pp. 329–337, 2002.

[23] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[24] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Information Sciences*, vol. 178, no. 23, pp. 4550–4559, 2008.

[25] S. Song, Y. Sun, A. Zhang, L. Chen, and J. Wang, "Enriching data imputation under similarity rule constraints," *IEEE transactions on knowledge and data engineering*, vol. 32, no. 2, pp. 275–287, 2018.

[26] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Curran Associates Inc., 2016, pp. 847–855.

[27] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[28] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[30] 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*, vol. 72, pp. 168–181, 2016.

[31] J. Yoon, J. Jordon, and M. Van Der Schaar, "Gain: Missing data imputation using generative adversarial nets," *arXiv preprint arXiv:1806.02920*, 2018.

[32] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao, and J. Yin, "Generative semi-supervised learning for multivariate time series imputation," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 8983–8991.

[33] B. Yang, Y. Kang, Y. Yuan, X. Huang, and H. Li, "St-lbagan: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation," *Knowledge-Based Systems*, vol. 215, p. 106705, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120308340

[34] R. Wu, A. Zhang, I. F. Ilyas, and T. Rekatsinas, "Attention-based learning for missing data imputation in holoclean," in *Proceedings of Machine Learning and Systems 2020, MLSys 2020*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds. mlsys.org, 2020. [Online]. Available: https://proceedings.mlsys.org/book/307.pdf

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[36] X. Shi, D. Zhao, H. Yao, X. Li, D. K. Hale, and A. Ghiasi, "Video-based trajectory extraction with deep learning for high-granularity highway simulation (high-sim)," *Communications in Transportation Research*, vol. 1, p. 100014, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772424721000147

[37] X. Shi, D. Zhao, and X. Li, "A car following-based method for vehicle trajectory connection," 07 2021.

[38] S. Tak, S. Woo, and H. Yeo, "Data-driven imputation method for traffic data in sectional units of road links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1762–1771, 2016.

[39] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624–1630, 2020.

[40] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5243–5253, 2019.

[41] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[42] J. Mackenzie, J. F. Roddick, and R. Zito, "An evaluation of htm and lstm for short-term arterial traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1847–1857, 2018.

[43] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial-temporal attention approach for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[44] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," *CoRR*, vol. abs/2108.09091, 2021. [Online]. Available: https://arxiv.org/abs/2108.09091

[45] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[46] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, ser. IJCAI'18. AAAI Press, 2018, pp. 3634—3640.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[48] C. Zhang, J. J. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.

[49] J. J. Q. Yu, C. Markos, and S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[51] X. Chen and L. Sun, "Bayesian temporal factorization for multidimensional time series prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[52] J. J. Yu, "Citywide traffic speed prediction: A geometric deep learning approach," *Knowledge-Based Systems*, vol. 212, p. 106592, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120307218

Shuyu Zhang received his B.Eng. degree in computer science from the Southern University of Science and Technology, Shenzhen, China, in 2021. He is currently working toward the master degree in Urban Informatics and Smart Cities with the Hong Kong Polytechnic University. His research interests include smart cities, urban computing, and deep learning.

Chenhan Zhang(S'19) received the B.Eng. degrees in Telecommunication Engineering from University of Wollongong, Wollongong, Australia, and Zhengzhou University, Zhengzhou, China in 2017 and 2018, respectively. He received the M.S degree in Engineering Management from City University of Hong Kong in 2019. He is currently a PhD student at Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His research interests include security and privacy of graph neural networks and trustworthy intelligent transportation systems.

Shiyao Zhang(S'18–M'20) received the B.S. degree (Hons.) in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2014, the M. S. degree in Electrical Engineering (Electric Power) from University of Southern California, Los Angeles, CA, USA, in 2016, and the Ph.D. degree from the University of Hong Kong, Hong Kong, China. He was a Post-Doctoral Research Fellow with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology from 2020 to 2022. He is currently a Research Assistant Professor with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology. His research interests include smart energy systems, intelligent transportation systems, optimization theory and algorithms, and deep learning applications.

James J. Q. Yu(S'11–M'15–SM'20) is an assistant professor at the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an honorary assistant professor at the Department of Electrical and Electronic Engineering, the University of Hong Kong. He received the B.Eng. and Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow at the University of

Hong Kong from 2015 to 2018. His general research interests are in smart city and privacy computing, deep learning, intelligent transportation systems, and smart energy systems. His work is now mainly on forecasting and decision making of future transportation systems and artificial intelligence techniques for industrial applications. He was ranked World's Top 2% Scientists of 2019 and 2020 by Stanford University. He is an Editor of the IET SMART CITIES journal and a Senior Member of IEEE.