2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922425

# Towards missing traffic data imputation using attention-based temporal convolutional networks*

Weiqiang Chen, Jianlong Zhao, Wenwen Wang, and Huijun Dai

*Abstract*— **Measurements from traffic sensors usually go missing at unanticipated moments as a result of detector malfunctioning, communication error, or erratic sampling. These missing data may weaken or even imperil the validity and effectiveness of data-driven traffic applications. The paper proposes a novel deep learning-based missing traffic imputation framework using self-attention based temporal convolutional network (ATCN) to achieve fast extraction of spatio-temporal traffic patterns. To be specific, our proposed ATCNImp model uses an encoder-decoder architecture with one-dimensional convolutional (Conv1D) layer to obtain spatial representations. Additionally, the ATCN module is utilized between the encoder and decoder to capture long-range spatial-temporal dynamics, further enhancing feature presentations and highlighting degradation information under traffic missing settings. Finally, the performance of our ATCNImp is verified using the public PeMS-BAY dataset. Experimental results reveal that ATCNImp outperforms the other four imputation models and provides stable imputation performance.**

## I. Introduction

Thanks to the advanced sensor technology, intelligent transportation systems (ITS) can detect a significant amount of traffic data generated on the traffic network. In order to lessen traffic congestion and increase traffic safety, extensive researches have been done to develop effective ways for extracting traffic information from the vast volumes of data [1]. However, due to traffic detector failure, communication fault, or erratic sampling, measurements from certain sensor may be lost at some unanticipated moments. Furthermore, missing traffic may reduce or even imperil the validity and performance of applications that rely on such data, such as classification, clustering, and forecasting tasks, resulting in biased inference [2]. Therefore, learning how to quickly estimate and impute the missing traffic values is essential.

Researchers have created a variety of imputed techniques over the past few decades, which may broadly be divided into two groups: statistical and machine learning (ML) approaches [1–5]. The early statistical model is to use an autoregressive integrated moving average (ARIMA) [6], which works well under the linear assumption. The dimensional reduction model for missing imputation also achieves fairly results, e.g., a probabilistic principle component analysis (PPCA) method [7]. However, this kind of methods may lose detailed features while acquiring holistic patterns. K-Nearest Neighbors (KNN), which estimates missing data from the K closest data points [8], is the most extensively used method for impute missing values. Incrementally growing in the scale of traffic data impels the researchers to exploit more scalable imputation technique. Deep neural networks have recently excelled in solving complex problems, and their flexible design for feature representation and degradation modeling demonstrate their superiority. For the imputation of missing traffic data, a novel denoising stacked autoencoder (DSAE) has been researched in [9,10]. In [11], a deep overcomplete denoising autoencoder multiple imputation model is confirmed. It is suggested to use a context autoencoder built on a convolutional neural network (CNN) to tackle the missing traffic imputation problem [12], which can use convolutional layers to capture local spatial and temporal relationships.

Although deep learning models have been widely developed to predict traffic data, it is still challenging for large-scale traffic imputation problems. The reason is that these models do not fully consider long-range spatio-temporal attentive patterns. Recent research has shown that a more sophisticated temporal convolutional network (TCN) [13] for sequential modeling performs well in speech synthesis and machine translation. TCN consists of causal convolutions and one-dimensional fully convolutional network (1D-FCN). The 1D-FCN ensures the length of TCN's input and output are constant. The receptive field of TCN can be greatly increased by using dilated causal convolutions and residual modules, making it easier to perceive long-range memories [13]. TCN has the ability to process features in parallel and model temporal components. The general TCN framework, however, gives equal weights to all of the distinct features and may worsen the imputation accuracy when particular important time-steps of the input data need to be focused. The attention mechanism [14-15] used in the prediction models was created to generate attentive weights for different parts of input, allowing the model to extract more significant features. Additionally, the attention mechanism can directly model the relationship between each two distant time steps without relying on the previous calculated results, so it can be processed in parallel with TCN, enabling the model to improve performance without consuming too much time.

To learn long-range enhanced spatio-temporal patterns from traffic data, this paper proposes ATCNImp, an integrated deep learning framework based on attention-based TCN combined with encoder-decoder architecture. ATCNImp can extract attentive spatio-temporal traffic patterns quickly while facilitating training and parallel deployment. This study makes two contributions. First, ATCNImp uses an encoder-decoder

*Research supported by the National Key Research and Development Program of China (Grant No. 2021YFA1000128), the Shandong Province Postdoctoral Program for Innovative Talents (Grant No. SDBX2021020) and the Shaanxi Province Industrial Research Program (Grant No. 2019GY-005).

Weiqiang Chen, Jianlong Zhao, and Wenwen Wang are with the Research and Development Center, Qingdao Hisense TransTech Co., Ltd., Qingdao 266100 China (e-mail: zhaojianlong@hisense.com).

Huijun Dai, is with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049 China (e-mail: daihuijun@ mail.xjtu.edu.cn).

architecture with Conv1D layers to obtain spatial features. Besides, between the encoder and decoder, a self-attention based TCN (ATCN) module is employed to capture the long-range spatiotemporal patterns, further enhancing feature presentations and highlighting degradation information under traffic missing settings. Finally, we test the ATCNImp model to see how well it can impute missing traffic speed values under various missing ratios. The results show that the proposed model outperforms the other four imputation models and provides stable imputation results.

The rest of paper is formed as follows: Section II presents the related works. Section III describes the missing traffic imputation problem. Section IV explains the proposed imputation model. Section V shows the experimental results. Section VI summarizes the paper and gives the future work.

## II. Related Works

Several statistical and machine learning (ML)-based imputation techniques have been utilized to impute missing traffic values. Early statistical imputations, such as mean imputation, linear, spline and cubic imputations suffer from the assumption of potential data distribution, which makes them unable to model complex nonlinear dependencies. In addition, the matrix completion technique has also been applied to impute missing data [16]; however, it imposes restrictions on low-rankness. As AI technology has advanced quickly, ML-based imputation models have been proposed to solve shortcomings of statistical imputation methods, such as SVR [17], KNN [8] and PPCA [7]. However, SVR and KNN models are unable to integrate the link between various parameters. PPCA method is created on traditional principal component analysis by incorporating prior traffic distribution. However, this kind of methods may lose detailed features while acquiring holistic patterns due to the high dependence on the assumed probability distribution model.

Recently, some researchers used deep learning methods to model complex traffic patterns. These strategies have been demonstrated to outperform ML and statistical methods when it comes to spatio-temporal mining on traffic datasets. Duan et al. [9,10] employed deep neural network in traffic missing imputation with DSAE model. The main idea is to put raw data with missing parts into an encoder that can obtain semantically meaningful features. The decoder then estimates the missing input based on its hidden features, and its output represents the imputation results. Zhuang et al. [12] proposed an innovative approach for traffic imputation on the basis of CNN and autoencoder framework. The practicality of CNN in solving missing traffic imputation is discussed in this work. The CNN-based imputation approach has a similar structure to the DSAE method. However, the ConvPooling layer, which can assess more complicated spatial-temporal information, makes a significant impact. A modified GRU-D recurrent neural network model which incorporates masking vector and time gap directly into the GRU unit, is developed to impute missing time-series data [18], but the spatial pattern is not considered. The above works all consider that data of each sensor at the whole time stamps contributes fairly to missing traffic data, which may cause negative effect because the more informative temporal feature of the collected data should be paid more attention than other time-steps.

The most important features for the missing traffic imputation task have not been adequately investigated due to the complexity of the dynamic spatiotemporal features. The majority of existing imputation methods for missing traffic suffer from enhanced spatiotemporal feature extraction and effective methods. To this end, this paper proposes a deep learning based ATCNImp framework, an encoder-decoder architecture based on TCN with self-attention to extract spatiotemporal traffic patterns quickly and generate enhanced representations for missing traffic imputation.

## III. Problem Definition

Suppose the missing traffic data matrix $\boldsymbol{X}^m = \boldsymbol{X} \cdot \boldsymbol{M}$ for a particular type of data such as traffic speed. Here, $\boldsymbol{M} \in \mathbb{R}^{N \times T}$ denotes the missing indicator matrix in which each entry is defined as

$$m_{n,\tilde{t}} = \begin{cases} 1, & if\ x_{n,\tilde{t}}\ is\ observed \\ 0, & if\ x_{n,\tilde{t}}\ is\ missing \end{cases} \tag{1}$$

where $n = 1,...,N$ denotes the number of traffic sensors, and $\tilde{t} = 1,...,T$ denotes the number of time steps; $\boldsymbol{X}$ represents the complete traffic flow matrix, in which $\boldsymbol{x}_{\tilde{t}} \in \mathbb{R}^{N}$ represents the observed data vector at the $\tilde{t}$ -th time-step. The missing traffic imputation problem aims to obtain the reconstructed input $\hat{\boldsymbol{X}}$, thus filling the missing values as close to the real data as possible by minimizing a loss function $Loss(\hat{\boldsymbol{X}}, \boldsymbol{X}^m)$, e.g., mean squared error.

This study focuses on creating a deep multiple imputation framework for missing traffic data since multiple imputation mechanisms can increase the robustness of imputation models [3], by which each missing entry is replaced with the weighted average of multiple imputation values. In order to use the suggested model for multiple missing traffic imputation, a sliding window method must first create a data point $\boldsymbol{x}_t^m \in \mathbb{R}^{N \times W}$ with time window size $W$ and $t \in [0, T\text{-}W+1]$. The objective is to reconstruct missing data for $\boldsymbol{x}_t^m$ based on the spatial and temporal patterns. Fig.1 shows a schematic example of applying a sliding window on traffic data with random missing entries. An estimated traffic matrix $\hat{\boldsymbol{X}}$ will be obtained following the imputation. Consequently, the missing data in $\boldsymbol{X}^m$ may be recreated using the feed-forward output of the ATCNImp model.

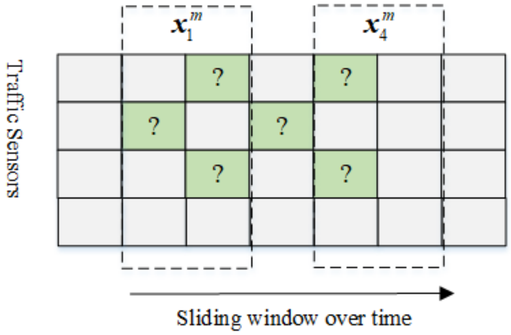


Fig.1 A sliding window generates data sample. The missing values are presented in small green square with '?' symbol

## IV. The ATCNImp Model

Fig.2 depicts the ATCNImp framework for missing traffic imputation problem, which employs an encoder-decoder architecture to learn the enhanced spatiotemporal traffic patterns. The ATCN module between the encoder and decoder forms the core of the imputation framework, which aims to learn the long-range weighted features from the encoded series. The Conv1D operation is used in the encoder and decoder layers to perform nonlinear transformations using ReLU and Linear activation functions, respectively. The encoder captures the local spatial relationship of traffic series among various sensors by encoding the input sequence. The ATCN module is then given the encoded series to continue learning long-range weighted temporal relationships. Finally, by utilizing the long-range spatio-temporal traffic patterns, the decoder recovers a feature map with the same dimension as the missing traffic data. As a result, a more thorough and precise learning of the sequential regularities found in the missing traffic matrix was possible. During training, the ATCNImp model receives inputs with a size of $W$ from a sliding window mechanism. For each time window, there are W anticipated values. The average of these reconstructed values represents the outcome of the imputation model. The subsection provides a detailed explanation of the ATCNImp framework's structure and optimization strategy.

### A. *ATCNImp Model*

Firstly, an encoder consisting of Conv1D with ReLU activation is to extract the local spatial patterns among different sensors. The kernel in the Conv1D layer slides over traffic series data $\boldsymbol{x}_t^m$. Note that the kernel only spans over the time dimension. Suppose the size of kernel is ($N$, $q$), where $q < W$, and the stride size is ($N$, 1). After multiple convolutions, all outputs will be concatenated and denoted with $\boldsymbol{H}^{enc} \in \mathbb{R}^{F\times W}$, where $F$ denotes the total size of all encoder filters.

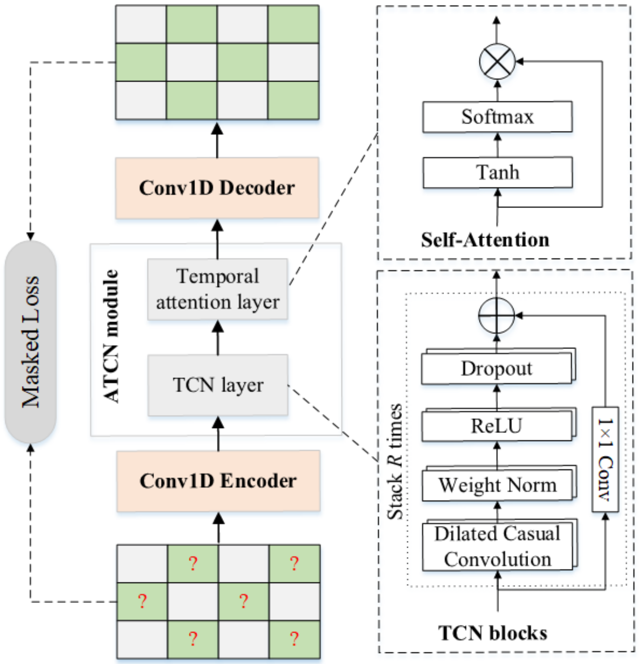


Fig. 2 The proposed ATCNImp framework

Next, the ATCN module receives the outputs of Conv1D encoder to learn long-range temporal patterns. ATCN module consists of self-attention and a TCN layer comprised of several TCN blocks. The TCN layer adopted in ATCN retains the similar architecture in [13], which applies hierarchical temporal convolutions across its inputs, thus extracting its typical features from different temporal scales. As shown in Fig.2, the TCN layer is comprised of $R$ TCN residual blocks, each of which is mainly based on the 1-D dilated casual convolution. This operation means that a filter $f$ with size $K$ ($\boldsymbol{f} \in \mathbb{R}^K$) and dilation factor $d$ is convolved with an encoded sequence $\boldsymbol{h}_i^{enc} \in \mathbb{R}^W, i = 1,..,F$. Specifically, the dilated causal convolution operation on the $t$-th element in the sequence $\boldsymbol{h}_i^{enc}$ is denoted as

$$s_t = \left(\boldsymbol{f} *_d \boldsymbol{h}_i^{enc}\right)_t = \sum_{k=0}^{K-1} f_k h_{i,t-k\cdot d}^{enc} \tag{2}$$

where $t - k\cdot d$ denotes the direction of the past, $d$ increases as the network goes deeper, computed by $d_l = 2^l$ at level $l$ of the network. Stacking multiple dilated causal convolutions allow TCN to possess very wide receptive fields and to obtain long-range temporal patterns with a smaller number of layers. TCN further adopts residual connections to facilitate gradients flow. Let $\boldsymbol{H}_0$=$\boldsymbol{H}^{enc}$, the set of operations at each layer can be represented as follows,

$$\tilde{\boldsymbol{H}}_l = \varphi(\boldsymbol{W}_d * \boldsymbol{H}_{l-1} + \boldsymbol{b}_d) \tag{3}$$

$$\boldsymbol{H}_l = \boldsymbol{H}_{l-1} + \boldsymbol{W}_r * \tilde{\boldsymbol{H}}_l + \boldsymbol{b}_r \tag{4}$$

where $\varphi(\cdot)$ denotes the ReLU activation function, * denotes the convolution operator, $\boldsymbol{H}_l$ denotes the output of layer $l$, $\boldsymbol{W}_d$ is the weights of the dilated convolution filters, $\boldsymbol{W}_r$ is the weights of 1×1 convolution, and $\boldsymbol{b}_d, \boldsymbol{b}_r$ denotes bias vectors. The outputs of the last residual block on the TCN layer is then passed to the temporal attention layer.

The temporal attention module that we use above the TCN layer to generate the estimated imputation is based on the self-attention [15]. In the process, the outputs of TCN layer are weighted by a self-attention mechanism across time axis. This procedure intends to focus more important time steps which may devote more to missing traffic estimation. Note that the inputs and outputs in ATCN module have the same size. Assume that the learned features by ATCN module are expressed as $\boldsymbol{X}' \in \mathbb{R}^{F\times W}$. Here, $\boldsymbol{x}_i' \in \mathbb{R}^W$, where $W$ is the size of sliding window. According to the self-attention mechanism, the importance for distinct time steps of $i$-th input can be defined as

$$\boldsymbol{q}_i = \phi(\boldsymbol{W}_a \boldsymbol{x}_i' + \boldsymbol{b}_a) \tag{5}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{W\times W}$ and $\boldsymbol{b}_a \in \mathbb{R}^W$ denote the weight matrix and the bias vector for $i$-th feature, respectively, $\phi(\cdot)$ is the score function represented by *Tanh* activation function. After acquiring the score of the $i$-th feature vector, it can be normalized using *softmax* function, as shown below:

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{q}_i) = \frac{\exp(\boldsymbol{q}_i)}{\sum_t \exp(\boldsymbol{q}_i)} \tag{6}$$

Consequently, the ultimate outputs $\tilde{\boldsymbol{X}}$ of the temporal attention layer can be denoted as

$$\tilde{\boldsymbol{X}} = \boldsymbol{X}' \otimes \boldsymbol{A} = \{\boldsymbol{\alpha}_1 \boldsymbol{x}'_1, ..., \boldsymbol{\alpha}_F \boldsymbol{x}'_F\} \tag{7}$$

where $\boldsymbol{A} = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_F\}$ , $\otimes$ represents the element-wise multiplication operation.

Finally, the decoder consisting of a Conv1D layer with Linear activation uses the long-range spatiotemporal features $\tilde{\boldsymbol{X}}$ to reconstruct the missing input data. Therefore, the reconstructed outputs $\hat{\boldsymbol{X}}$ can automatically impute the missing values on the basis of the spatial-temporal correlations among traffic series. Given a time sliding window *W*, each time-step $\boldsymbol{x}_t, t = 1, ..., T$ will be reconstructed *W* times and the average results can be used for missing imputation results.

### B. Optimization Strategy

After the imputation process, an estimated matrix $\hat{\boldsymbol{X}}$ is acquired. Considering that only the observed elements can be used as inputs, the masked loss function tailored for missing traffic imputation problem is denoted as,

$$L(\theta) = \underset{\theta}{\text{minimize}} \left\| (\hat{\boldsymbol{X}} - \boldsymbol{X}^m) \cdot \boldsymbol{M} \right\|_F^2 \tag{8}$$

where $\theta$ is the parameter sets of ATCNImp model introduced in all layers, and $\|\cdot\|_F$ denotes the Frobenius norm. The training process is based on Adam optimization method [19]. By abiding by the optimization process described in (8), the difference between the observed parts of $\boldsymbol{X}^m$ and the corresponding part of $\hat{\boldsymbol{X}}$ can be decreased. And then using the feed-forward outputs of the ATCNImp model, the missing data in $\boldsymbol{X}^m$ can be reconstructed.

## V. Experimental Results

### A. Datasets and Preprocessing

We undertake experiments on one public traffic dataset (PeMS-BAY) [20], which contains 6 months of traffic speed values on 325 sensors located in the Bay Area. We adopt the same data pre-processing procedures as [20]. The observations of the sensors are aggregated into 5-minute windows. In this study, we take one month's data from January 1st to January 31st, and use the first three weeks for training and the rest for testing. The input time-series data is generated using a sliding window technique. We discovered that a time window of size 24, 2 hours worked effectively during the experiments. In this study we attempt to reconstruct missing data with MCAR (missing completely at random) and MNAR (missing not at random) patterns [11] using the following detailed steps.

- A uniform random vector *v* with *p* observations is attached to a data set with values between 0 and 1, where *p* is the number of observations in the data.
- MCAR: Uniformly set all attributes to own missing data where $v_i \le \lambda, i = 1:p$ , $\lambda$ is the missingness threshold ranging from 10% to 50% with 10% gain.
- MNAR: Randomly sample two attributes $s_1$ and $s_2$ from the dataset and compute their median value $m_1$ and $m_2$. Uniformly set all attributes to own missing data where $v_i \le \lambda, i = 1:p$ and($s_1 \le m_1$ or $s_2 \ge m_2$).

### B. Evaluation Metrics

We use three commonly used metrics to quantify the quality of missing traffic imputation to evaluate the imputation performance of ATCNImp model: mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination ($R^2$). The imputation error is measured by MAE, and the sample standard deviation of the differences between imputed and observed values is measured by RMSE: the smaller the value, the better the imputation effect. The correlation coefficient $R^2$ is a measure of the ability of the estimated results to match the real data: the higher the value, the greater the prediction effect.

$$MAE = \frac{1}{N'} \sum_{i=1}^{N'} |x_i - \hat{x}_i| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N'} \sum_{i=1}^{N'} (x_i - \hat{x}_i)^2} \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N'} (x_i - \hat{x}_i)^2}{\sum_{i=1}^{N'} (x_i - \bar{x})^2} \tag{11}$$

where $N'$ denotes the number of missing data entries, $x_i$ and $\hat{x}_i$ denotes the real traffic value and the missing imputed value respectively, $\bar{x}$ denotes the average of imputed set.

### C. Results Analysis

We compare the imputation performance of the ATCNImp model with the four baseline methods:

- Mean imputation (MeanImput), which is one of the most-naive and easiest methods for imputing missing values. The missing entries are reconstructed using the mean of the non-missing entries of each variable.
- KNN [8], is a non-parametric method. The output is the average value of its K nearest neighbors. The missing values are calculated using the K nearest spatiotemporal neighbors (K = 30) in this study.
- PPCA [7], which iteratively calculates the greatest likelihood estimates of an incomplete data set using the expectation maximization procedure.
- Convolutional neural network based autoencoder (CNNAE) [12], which creates a context autoencoder based on convolutional neural networks to estimate the complete traffic image from the missing data.

Next we describe the implementation details of the proposed method. The batch size is set to 128 and the epoch is set to 200 while training the ATCNImp model. The model is trained by an Adam optimizer, and the learning rate is 0.001. The encoder's Conv1D layer is made up of 32 filters of size (*N*, 3) and a ReLU activation function. The decoder's Conv1D layer is made up of 32 filters of size (*N*, 3) and a Linear activation function. All models are developed and tested (75%/25% train/test split) with Pytorch1.7.0.

TABLE I. Experimental results by imputation methods: MAE, RMSE and $R^2$ is revealed about each condition with different missing mechanism and missing ratios. The best results for each condition are marked in bold and the second-best results are marked in italics.

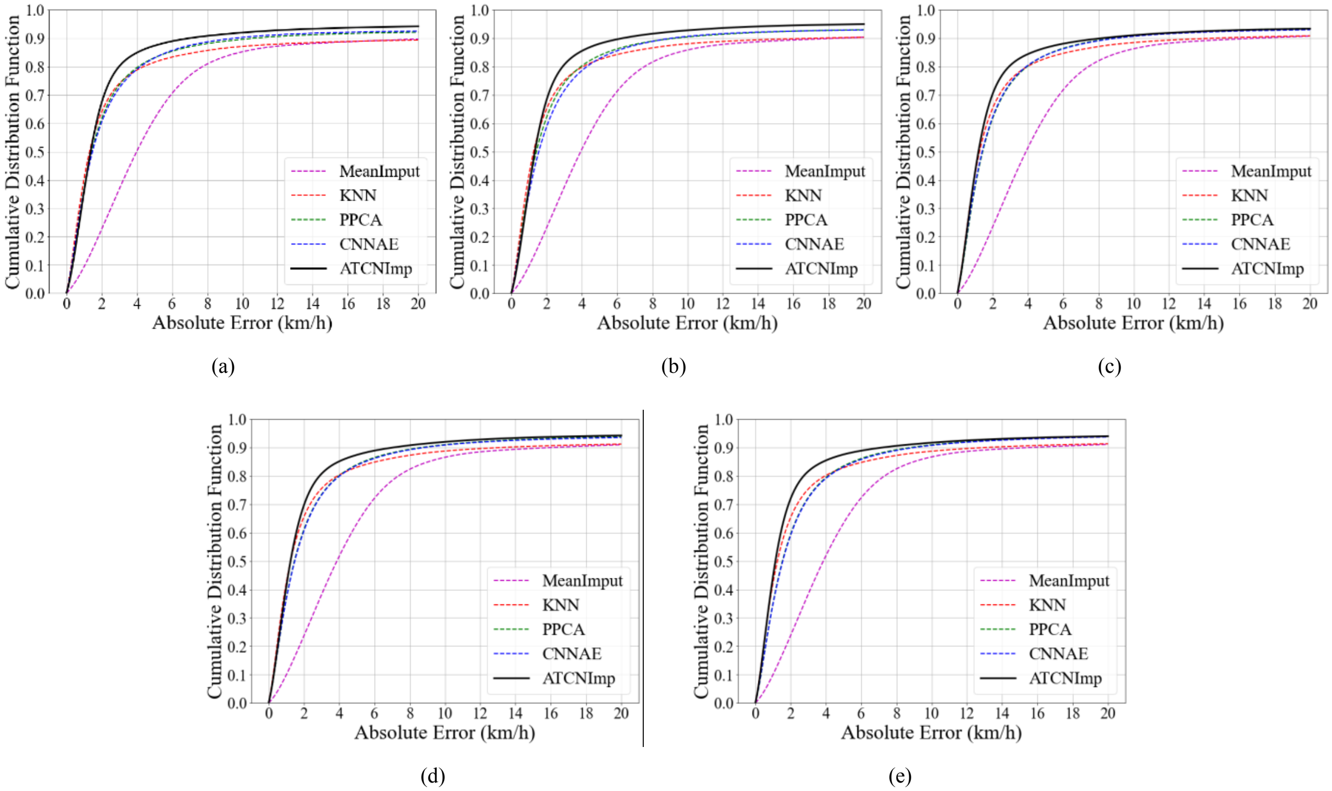
| Missing ratio | Methods | MCAR | | | MNAR | | |
|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | $R^2$ | MAE | RMSE | $R^2$ |
| 10% | MeanImput | 5.771 | 9.880 | 0.064 | 5.960 | 10.397 | 0.066 |
| | KNN | **1.985** | 4.054 | 0.842 | 2.230 | 4.433 | 0.842 |
| | PPCA | 2.123 | 3.872 | 0.855 | 2.359 | 4.208 | *0.857* |
| | CNNAE | 2.145 | *3.787* | *0.862* | *2.119* | *3.843* | 0.855 |
| | **ATCNImp** | *1.993* | **3.644** | **0.870** | **2.018** | **3.762** | **0.868** |
| 20% | MeanImput | 5.767 | 9.849 | 0.062 | 5.939 | 10.369 | 0.065 |
| | KNN | *2.003* | 4.115 | 0.836 | *2.253* | 4.499 | 0.836 |
| | PPCA | 2.164 | 3.947 | 0.849 | 2.401 | 4.303 | 0.850 |
| | CNNAE | 2.218 | *3.826* | *0.858* | 2.334 | *4.013* | *0.869* |
| | **ATCNImp** | **1.990** | **3.612** | **0.874** | **2.032** | **3.753** | **0.870** |
| 30% | MeanImput | 5.733 | 9.776 | 0.063 | 5.878 | 10.277 | 0.067 |
| | KNN | *2.036* | 4.189 | 0.827 | *2.288* | 4.572 | 0.828 |
| | PPCA | 2.215 | 4.025 | 0.841 | 2.451 | 4.383 | 0.842 |
| | CNNAE | 2.119 | *3.843* | *0.850* | 2.561 | *4.365* | *0.843* |
| | **ATCNImp** | **1.869** | **3.662** | **0.868** | **2.107** | **3.762** | **0.870** |
| 40% | MeanImput | 5.721 | 9.756 | 0.062 | 5.847 | 10.270 | 0.066 |
| | KNN | *2.061* | 4.238 | 0.823 | *2.326* | 4.642 | 0.823 |
| | PPCA | 2.280 | 4.119 | 0.832 | 2.518 | 4.498 | 0.833 |
| | CNNAE | 2.244 | *4.001* | *0.842* | 2.616 | *4.491* | *0.834* |
| | **ATCNImp** | **1.915** | **3.647** | **0.869** | **1.962** | **3.726** | **0.870** |
| 50% | MeanImput | 5.732 | 9.777 | 0.062 | 5.832 | 10.298 | 0.065 |
| | KNN | *2.118* | 4.379 | 0.811 | *2.398* | 4.807 | 0.810 |
| | PPCA | 2.406 | 4.329 | 0.810 | 2.649 | 4.757 | 0.815 |
| | CNNAE | 2.396 | *4.277* | *0.820* | 2.702 | *4.666* | *0.821* |
| | **ATCNImp** | **1.855** | **3.701** | **0.865** | **1.947** | **3.777** | **0.866** |

The imputation results of the PeMS-BAY dataset are shown in Table 1. The ATCNImp model outperforms the other baseline models in general. MeanImput is one of the most naive and simplest ways for imputing missing values among the four baseline methods, as it does not take advantage of any correlation between the variables and hence performs poorly in terms of MAE, RMSE, and $R^2$ metrics. Furthermore, in most circumstances, the PPCA model outperforms the KNN model on three metrics. Deep learning imputation methods based on CNNAE and ATCNImp, on the other hand, outperform statistical and machine learning based imputation models. Deep neural networks can detect complicated spatial and temporal patterns in traffic sequences, which explains this occurrence. With the exception of a lower missing ratio (10%), ATCNImp is the best technique when encountering the MCAR mechanism. KNN has a smaller MAE metric than ATCNImp but a slightly higher RMSE metric when the missing ratio is 10%. This indirectly proves that imputation results of ATCNImp model are less likely to have substantial errors. Regardless of the missing ratios, ATCNImp is always the optimum imputation approach when considering the MNAR mechanism. Furthermore, the proposed ATCNImp model can always maintain stable imputation results and is not susceptible to changes in missing ratios under both MCAR and MNAR missing patterns. This implies that ATCNImp can effectively extract enhanced spatial and temporal traffic patterns suitable for missing traffic imputation, resulting in a stable imputation model.

Furthermore, the results from Table 1 are confirmed by calculating the cumulative distribution function (CDF) of the imputation errors under different missing ratios to analyze the performance from a statistical standpoint. All three evaluation metrics (MAE, RMSE, $R^2$) focus on the average distance between predicted data and real data while either distance can be expressed in absolute error. As a result, we just need to care about the distribution of absolute error like in [10]. Fig. 3 and 4 demonstrate the absolute error distribution of MCAR and MNAR missing patterns with five different missing ratios. The absolute errors CDF curve of ATCNImp is always on the left of MeanImput, KNN, PPCA, and CNNAE, as shown in Figure 3. Furthermore, we can see that under 10%, 20%, 30%, 4%, and 50% missing ratios, about 88%, 90%, 87%, 88%, 88% imputed values have absolute errors of less than 6 km/h. The absolute errors' CDF curve of ATCNImp is always to the left of the other baseline models under the MNAR missing condition, as shown in Fig.4. Furthermore, with our deep learning technology, there are around 86%, 88%, 89%, 88%, 88% absolute errors under 6 km/h. In comparison to other models, the majority of errors are accumulating at a low level. This confirms the above-mentioned findings. To summarize, all of the above comparative results show that our ATCNImp model is superior for missing traffic imputation task.

## VI. Conclusion

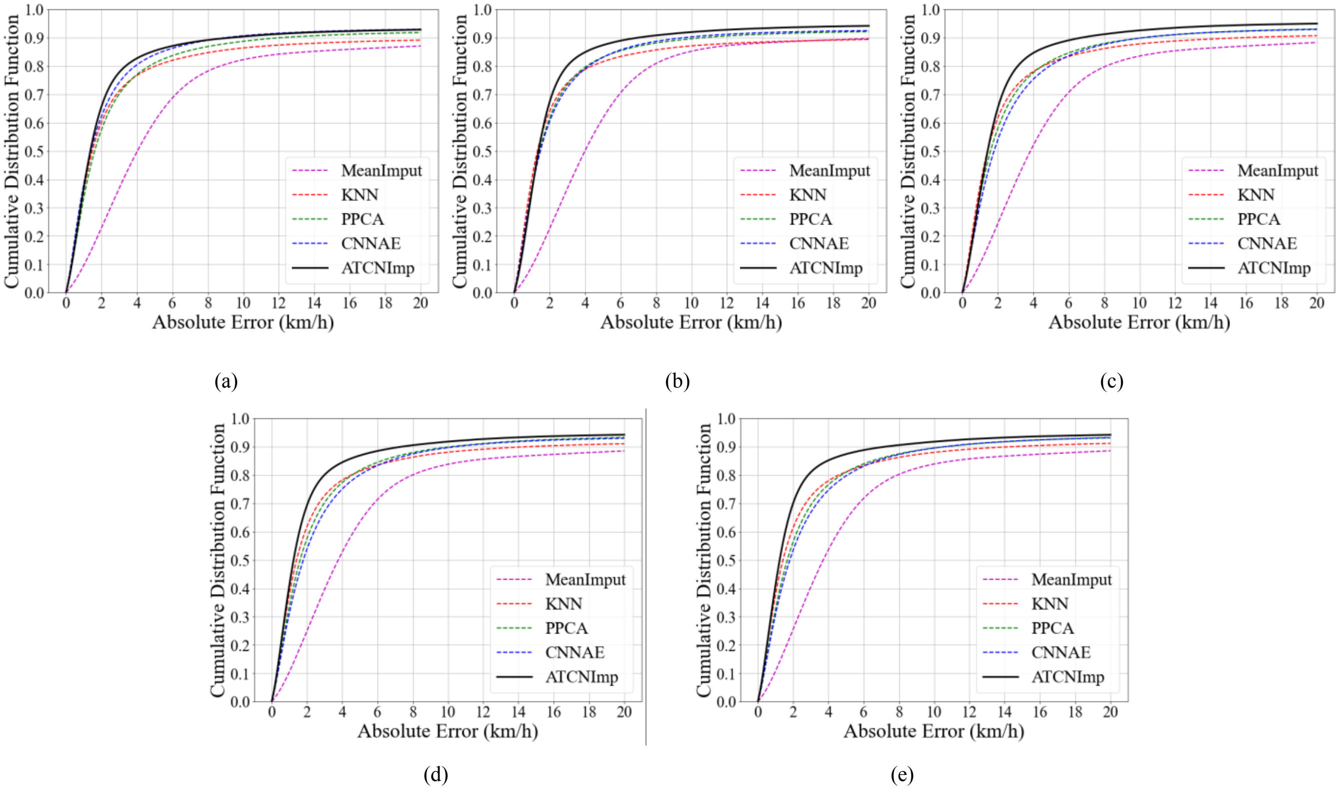
This paper regarded missing traffic data imputation as the corrupted data reconstruction problem and developed a deep learning based ATCNImp model to deal with it. The proposed ATCNImp model extracts spatial and temporal patterns using an attention-based TCN coupled with an encoder-decoder architecture, improving feature presentations and emphasizing degradation information under traffic missing operating settings, leading to more accurate imputation results. In terms of imputation accuracy, our ATCNImp model outperforms mean imputation, KNN, PPCA, and CNN-based autoencoder technique with missing ratios ranging from 10% to 50% with a 10% gain. The findings demonstrated that the ATCNImp technique yields the most consistent outcomes with the smallest estimated error. In the future, exploring the effects of spatio-temporal variables and looking into more sophisticated imputation architecture may lead to better performance for imputation of missing traffic data, which will help a number of ITS applications.



(a) (b) (c)

(d) (e)

Fig. 3 The distribution of the absolute errors under MCAR missing pattern with different missing ratios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, and (e) 50%



(a) (b) (c)

(d) (e)

Fig. 4 The distribution of the absolute errors under MNAR missing pattern with different missing ratios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, and (e) 50%

# REFERENCES

[1] Li L, Du B, Wang Y, et al., “Estimation of missing values in heterogeneous traffic data: application of multimodal deep learning model,” *Knowledge-based Systems.*, vol. 194, pp. 105592, 2020.

[2] Li L, Zhang J, Wang Y, et al., “Missing value imputation for traffic-related time series data based on a multi-view learning method,” *IEEE Transations on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2933-2943, 2018.

[3] Asadi R, Regan A., “A convolution recurrent autoencoder for spatio-temporal missing data imputation,” arXiv preprint arXiv:1904.12413, 2019.

[4] Duan Y, Lv Y, Liu Y L, et al., “An efficient realization of deep learning for traffic data imputation.” *Transportation research part C: emerging technologies*, vol. 72, pp. 168-181, 2016.

[5] Li Y, Li Z, Li L., “Missing traffic data: comparison of imputation methods,” *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 51-57, 2014.

[6] Elshenawy M, El-Darieby M, Abdulhai B, “Automatic imputation of missing highway traffic volume data,” in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2018, pp. 373-378.

[7] Qu L, Li L, Zhang Y, et al., “PPCA-based missing data imputation for traffic flow volume: A systematical approach,” *IEEE Transactions on intelligent transportation systems*, vol. 10, no. 3, pp. 512-522, 2009.

[8] Tak S, Woo S, Yeo H, “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.

[9] Duan Y, Lv Y, Kang W, et al., “A deep learning based approach for traffic data imputation,” in *Proceedings of 17th International IEEE conference on intelligent transportation systems (ITSC)*. IEEE, Qingdao 2014, pp. 912-917.

[10] Duan Y, Lv Y, Liu Y L, et al., “An efficient realization of deep learning for traffic data imputation,” *Transportation research part C: emerging technologies*, vol. 72, pp. 168-181, 2016.

[11] Gondara L, Wang K, “Mida Multiple imputation using denoising autoencoders,” in *Proceeding of Pacific-Asia conference on knowledge discovery and data mining*, Springer, Cham, 2018, pp. 260-272.

[12] Zhuang Y, Ke R, Wang Y, “Innovative method for traffic data imputation based on convolutional neural network,” *IET Intelligent Transport Systems*, vol. 13, no. 4, pp. 605-613, 2019.

[13] Bai S, Kolter J Z, Koltun V, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” arXiv preprint arXiv:1803.01271, 2018.

[14] Shi X, Qi H, Shen Y, et al., “A Spatial-Temporal Attention Approach for Traffic Prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4909-4918, 2020.

[15] Cho K, Van Merriënboer B, Gulcehre C, et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.

[16] Luo X, Meng X, Gan W, et al., “Traffic data imputation algorithm based on improved low-rank matrix decomposition,” *Journal of Sensors*, vol. 2019, 2019.

[17] Shang Q, Yang Z, Gao S, et al., “An imputation method for missing traffic data based on FCM optimized by PSO-SVR,” *Journal of Advanced Transportation*, vol. 2018, 2018.

[18] Che Z, Purushotham S, Cho K, et al., “Recurrent neural networks for multivariate time series with missing values,” *Scientific reports*, vol. 8, no. 1, pp. 1-12, 2018.

[19] Kingma D P, Ba J, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[20] Li Y, Yu R, Shahabi C, et al., “Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting,” in *Proceedings of International Conference on Learning Representations (ICLR)*. IEEE, 2018.