2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Traffic Data Imputation with Ensemble Convolutional Autoencoder

Yongchao Ye, Shuyu Zhang, and James J.Q. Yu, *Senior Member, IEEE*

***Abstract*— Intelligent transportation systems and related applications rely on high-quality traffic data. However, the data collected in real-world is often incomplete, which compromises the system performance. Traffic data imputation estimates the missing values by analyzing traffic flow features, therefore can improve the performance of related applications. Traditional imputation methods mainly focus on isolated traffic data sensors or road sections and show their limitations in representing complex spatial-temporal features. In this paper, we propose a novel ensemble model named ensemble convolutional autoencoder for the task. The observed values, together with the missing points are reconstructed into a two-dimensional matrix by the extracted spatial-temporal relation. Convolutional and deconvolutional layers are adopted to encode and decode spatial-temporal features, respectively. Besides, we train autoencoders with different input feature maps and ensemble the outputs by linear combination. Experimental results show that compared with other traffic data imputation methods, the proposed method can achieve better accuracy and has stable performance under various missing data scenarios with different types and rates.**

I. Introduction

Nowadays, a large amount of traffic data has been collected by the intelligent transportation system (ITS) with a massive amount of traffic sensors, such as loop detectors, GPS tracking devices and cameras [1], which monitors the road section automatically and generates real-time traffic data. With high-quality traffic data, many data-driven applications have been developed in ITS [2], e.g., traffic signal control, traffic flow prediction, etc. However, affected by many natural and human factors, traffic data collected in the physical world is not perfect. Missing data is commonly encountered in many traffic datasets. Typical reasons can be power failure, transmission error and beyond. It is reported that the rate of missing data is about 10% in Beijing and it can reach up to 25% in some cases [3]. Missing data compromise the performance of ITS and its related applications. For example, in traffic flow prediction, the performance reduces vividly when the traffic data is incomplete [4]. Traffic data imputation, which estimates the missing values by analyzing the spatial-temporal features of the observed values, can be crucial for improving ITS development.

Traditional traffic data imputation methods can be divided into three categories, namely, prediction, interpolation, and statistical learning [5]. Prediction models realize the imputation by forecasting, where the missing points are estimated by analyzing the previously observed data in time series. Such methods usually include time series modeling, see references [6], [7] for examples. These methods can be efficient when the missing rate is minuscule, i.e., only a few points in a notable time frame are missing. However, an obvious disadvantage of these methods is that the prediction model cannot make full use of the observed values after the missing points, resulting in undermined performance. Interpolation methods can be generally divided into three types, namely, spatial-neighboring, temporal-neighboring, and pattern-similarity [5]. According to the study by Yin *et al.*, spatial-neighboring is proved not as efficient as the other two interpolation types [8]. Temporal-neighboring interpolation fills the missing points by calculating the neighboring observed values in the time series on the same day. Pattern-similarity interpolation estimates missing points by analyzing the observed values from the same sensor on different days where k-nearest neighborhood (k-NN) and local least squares (LLS) are two typical methods [9]. However, interpolation methods are based on the assumption that the traffic flow is highly similar on consecutive days. These methods cannot be adjusted according to the stochastic changes in daily traffic flow. In the meantime, statistical learning methods, which can better capture the stochastic changes in traffic flow, achieve imputation by estimating a probability distribution model and iterating the model parameters. Probabilistic principal component analysis (PPCA) is a typical example in this category [3]. Other tensor-based methods reconstruct the traffic flow into third-order tensor and apply matrix decomposition methods in model training [10], [11].

With the rapid development of deep learning, many models have been proposed, such as autoencoder and generative adversarial network (GAN) for data generation, and there are results on end-to-end time-series data imputation based on generative models [12], [13]. Take [12] as an example, Berglund *et al.* introduced the probabilistic interpretations to a bidirectional recurrent neural network to reconstruct missing samples. In traffic data imputation, autoencoder was first applied in [14]. This study constructs a denoising stacked autoencoder (DSAE) with stacked autoencoders as the hidden layers of a neural network. Subsequent work [15] further improves DSAE by restructuring the model and conducting experiments on both weekdays and non-weekdays. At the same time, k-means clustering was applied to analyze traffic characteristics [16]. Besides, there are results that improve DSAE from the data augmentation perspective. Following this line of research, GAN was first employed to

This work is supported by the Stable Support Plan Program of Shenzhen Natural Science Fund No. 20200925155105002 and by the Department of Education of Guangdong Province, China No. SJJG201901. The authors are with the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. James J.Q. Yu is the corresponding author.

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564839

augment the train data. Which together with the observations are input into DSAE for imputation [17]. Its accuracy had been greatly improved compared to previous work. However, these works mainly focused on a standalone traffic sensor or a road section. Training one model for each sensor or road section may cause a large amount of computation when the investigating region is huge and complex. Similar problems also exist in traffic flow prediction tasks. A convolutional neural network (CNN) was first adopted to embed traffic data into a two-dimensional matrix for CNN training [18]. Zhuang *et al.* extended this work and applied CNN to data imputation by transforming traffic data imputation into an image recovery problem [19]. However, the reported experiment is relatively insufficient.

Despite many results have been reported for traffic data imputation, there is still room to improve. In this paper, we proposed an ensemble convolutional autoencoder. The traffic data is first embedded into a two-dimensional matrix for extracting spatial-temporal correlations among the raw data. Then an ensemble of convolutional autoencoders is applied to perform the data imputation. Each autoencoder has the same network architecture and is fed with different input feature maps. We conducted a series of comprehensive case studies to evaluate the model performance. The main contribution of this work is summarized as follows.

- We propose an ensemble autoencoder trained by heterogeneous input feature maps for traffic data imputation. The feature maps are constructed by filling the missing positions with zero and historical average values, respectively.
- We implement the imputation by CNN-based autoencoder. After embedding traffic data into a two-dimensional matrix, the convolutional layer can extract spatial and temporal characteristics.
- We evaluate the proposed method on the real-world traffic dataset. The massive results show that compared with other methods, the proposed method can achieve high imputation accuracy and maintain robust performance in different missing types and missing rates.

The remainder of the paper is organized as follows. In Section II, we present the proposed imputation method with elaboration on the proposed ensemble convolution autoencoder. In Section III, a series of experimental results and analyses are shown. Finally, the paper is concluded in Section IV.

II. Methodology

In this section, we give a detailed introduction of the proposed ensemble convolutional autoencoder. We first introduce the data pre-processing techniques, including the data embedding mechanism. Then, we propose a CNN-based autoencoder for traffic data imputation. Finally, we construct an ensemble for further performance improvement.

A. Data Preprocessing

In the proposed method, the observed traffic data is first embedded into a two-dimensional matrix. The horizontal axis represents the time stamp and the vertical axis represents the ID number of sensors. Since sensors on the same or adjacent roads are usually numbered consecutively, data from two neighboring rows in the matrix are collected by two sensors which are spatially adjacent. Thus, both spatial and temporal information are represented after transferring by this scheme. Assuming there are N sensors in a specific region and each sensor gets M observed values in a day, the size of the matrix should be $N \times M$. For the missed points in traffic data, the corresponding positions in the matrix are empty.

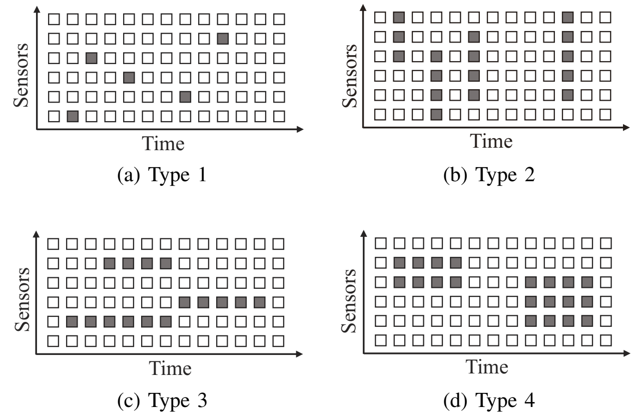


Fig. 1. Diagram of different missing types. (a) Missing completely at random. (b) Missing at temporal random. (c) Missing at spatial random. (d) Missing not at random

Let the ground truth traffic data be $X = \{x_{ij}\} \in \mathbb{R}^{n\times m}$, the observed input feature map with missing points is denoted as $X^l = \{x^l_{ij}\} \in \mathbb{R}^{n\times m}$, where n is the number of sensors, m is the number of time stamps in a day, and x_{ij} denotes the observed of sensor i at time j. A missing matrix $M = \{m_{ij}\} \in \mathbb{B}^{n\times m}$ is defined to record the missing position, where $m_{ij} = 1$ denotes a missing/null x^l_{ij} value, and $m_{ij} = 0$ denotes an observed x^l_{ij}. The output imputation result is denoted as $Y = \{y_{ij}\} \in \mathbb{R}^{n\times m}$.

Traffic data missing can be grouped into three categories: missing completely at random (MCR), missing at random (MR), and not missing at random (NMR) [4]. As the traffic data is reconstructed into a two-dimensional matrix, we divided traffic data missing into four types based on three categories as shown in Fig. 1, where blocks with dark shadow are missing. In MCR, all the missing points are randomly scattered, as shown in Fig. 1a. In MR, missing points are temporally or spatially neighbored as shown in Fig. 1b and Fig.1c, respectively. In NMR, the missing points are gregarious likes blocks as shown in Fig.1d.

B. CNN-based Autoencoder

The missing points and the observed values are often treated separately in traditional traffic data imputation methods. In this paper, we use the missing points together with the observed values as the input, and the output is the complete traffic data. According to this design, the imputation is

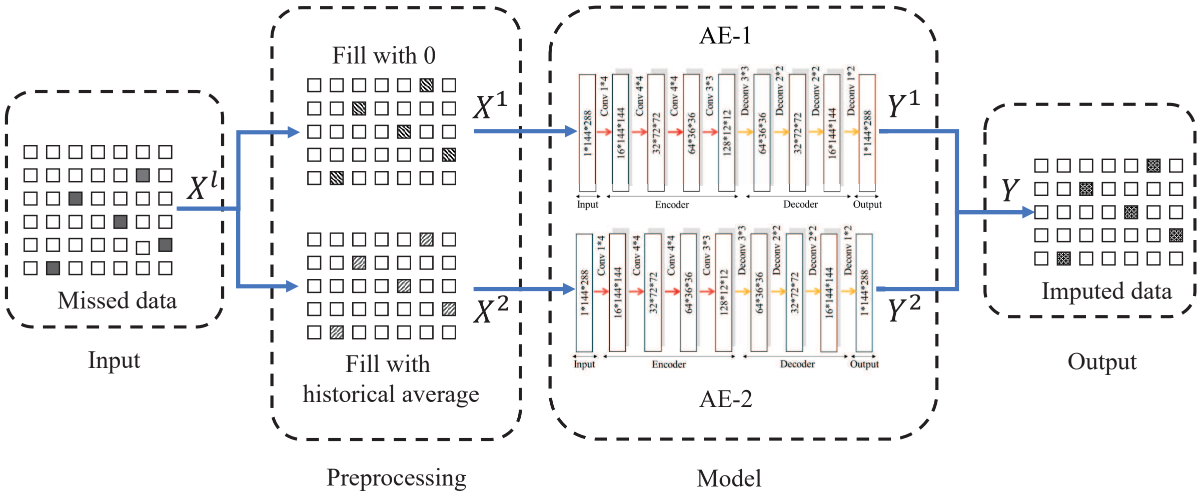


Fig. 2. Structure of the proposed ensemble convolutional autoencoder.

carried out by an autoencoder neural network that has the same input and output feature map size. The reconstruction loss of the autoencoder is the mean squared error between input and output feature map:

$$L(X,Y) = \|X - Y\|^2. \quad (1)$$

The imputation error $E(X,Y)$ is only calculated on the missing points on input and output feature map. $E(X,Y)$ is defined as follows:

$$E(X,Y) = M \odot \|X - Y\|^2, \quad (2)$$

where $\odot$ is the element-wise multiplication operation.

In the proposed CNN-based autoencoder, convolutional layers are applied to downsampling for extracting spatial-temporal features. Deconvolutional layers are applied to upsampling, which restores the traffic flow data. Specifically, the encoder consists of four convolutional layers. With a vector-like convolutional kernel in the first layer, the convolutional operation is only performed on one dimension of the feature map in the respective layer, namely, the temporal dimension, to focus more on temporal characteristics. The subsequent three layers all use canonical square convolution kernels, which extract both spatial and temporal features.

Like the encoder, the decoder consists of four deconvolutional layers. Contrary to convolution, deconvolution performs upsampling, which is essentially a reversed convolution. The size of the input feature map is enlarged by zero-padding according to a specific size, and then the convolution is performed with the kernel size. The convolution kernel size is carefully designed to ensure input and output feature maps have the same size. Symmetrical to the encoder, the last deconvolutional layer uses a vector-like kernel size. Meanwhile, the first three deconvolutions use square kernels. To prevent features from being lost in downsampling, no pooling layers are applied in the proposed autoencoder for imputation. As for the activation function, except that the output layer using $\tanh$, all other hidden layers use LeakyReLU for its relative simplicity in computation and gradient descent. As

TABLE I
HYPERPARAMETERS OF PROPOSED AUTOENCODER

	Layer	Kernel	Stride	Padding	In. Channel
Encoder	Conv 1	(1,4)	(1,1)	(0,1)	1
	Conv 2	(4,4)	(2,2)	(1,1)	16
	Conv 3	(4,4)	(2,2)	(1,1)	32
	Conv 4	(3,3)	(3,3)	-	64
Decoder	Deconv 1	(3,3)	(3,3)	-	128
	Deconv 2	(2,2)	(2,2)	-	64
	Deconv 3	(2,2)	(2,2)	-	32
	Deconv 4	(1,2)	(1,2)	-	16

a variant of ReLU, LeakyReLU prevents "neurons dying" when receiving negative input [20]. The architecture of the proposed autoencoder is concluded in Table I. Let the neural network parameters be θ, the autoencoder is trained by minimizing the reconstruction loss:

$$\arg\min_{\theta} L(X,Y) = \arg\min_{\theta} \sum_{i,j} \|x_{ij} - y_{ij}\|^2. \quad (3)$$

This parameter training objective can be optimized using gradient descent algorithms, e.g., Adam optimizer [21] as we adopted in the case studies to be discussed in Section III.

C. Ensemble Convolutional Autoencoder

As mentioned in the previous section, the missing points in the input data X^l are initially null. Typically, null values in the input feature maps are replaced by zeros before training. In this work, we augment the data richness by additionally using historical average values to pad the missing values. The incentive of using historical average values is based on the hypothesis that traffic flow data have similar characteristics over consecutive days. Hence, historical average values can provide auxiliary information in particular cases, especially under extreme data loss scenarios like block missing with a high missing rate. The proposed ensemble convolutional autoencoder employs the augmented two sets of data as input, which is fed into the proposed CNN-based autoencoder as shown in Fig. 2.

In the proposed ensemble convolutional autoencoder, the input data with missing positions filled with zeros is denoted by X^1, while the one padded with historical average value is denoted by X^2. Two autoencoders are marked as “AE-1” and “AE-2”, respectively. After the input feature maps going through the autoencoders, two matricized outputs Y^1 and Y^2 can be developed, which share the same dimensionality since the two autoencoders have the same structure as in Table I. The outputs are finally ensembled by the following parameterized data aggregation rule:

$$Y = \alpha \times Y_1 + (1 - \alpha) \times Y_2, \tag{4}$$

where α is the self-adapting weight value of the two autoencoders. The weight values, alone with other network parameters, can be updated via backpropagation during training. The imputation error of the ensemble model is defined identical to (3).

III. Case Studies

In this section, comprehensive experiments are conducted to evaluate the performance of the proposed imputation method. We first give a brief description of the datasets and experiment settings. Then, we compare the imputation accuracy of the proposed method with baseline approaches. Additionally, we inspect how each constituting component of the proposed ensemble model influences the model performance.

A. Dataset

PeMSD5[1] is a popular subset of California Performance Measurement System (PeMS) data. The dataset provides real-time and historical traffic data populated by the average traffic speed extracted from District 5 of California. The PeMS system collects raw speed every 30 seconds and performs aggregation every 5 minutes. Therefore, each sensor provides 288 observed values for each day. There are in total 153 sensors in District 5, of which 144 sensors from mainline are adopted in PeMSD5. We test the model performance on the weekday data ranging from January 1, 2013, to August 31, 2013.

B. Experiment Settings

The missing matrices are generated over all data employed in the test by random erasure. As spatiotemporal information loss may arise when missing data are observed, data augmentation is utilized in the training stage to give more information on each day frame. Specifically, for a certain missing type (c.f. Fig. 1) and missing rate, the erasure is repeated ten times per day. This means that there are in total ten input feature maps with different missing points for each day. Four missing types introduced in Section II-A are considered in this paper, each of which has data missing rates ranging from 10% to 50%. The training, validation and test sets are sequentially generated, each of which contains 60%, 20%, and 20% of all data. Note that this augmentation is only conducted on the training set. No changes are made to the validation or testing set.

When training the proposed ensemble convolutional autoencoder, we employ Adam optimizer for network parameter tuning. The learning rate is initially set to 0.001, which is subsequently dynamically adjusted with a decay rate of 0.5. The model is trained for 500 epochs with a batch size of 32. The proposed model is implemented by PyTorch, and all experiments are conducted on an NVIDIA GeForce RTX 2080Ti GPU. We use Mean Absolute Percentage Error (MAPE) as the evaluation metric in this paper, which is defined as follows:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\widehat{x}_i - x_i}{x_i} \right| \times 100\% \tag{5}$$

where $\widehat{x}_i$ is the imputed traffic speed at i, and x_i is the corresponding observed value.

C. Baselines

We employ the following approaches as baselines to evaluate the performance of the proposed model:

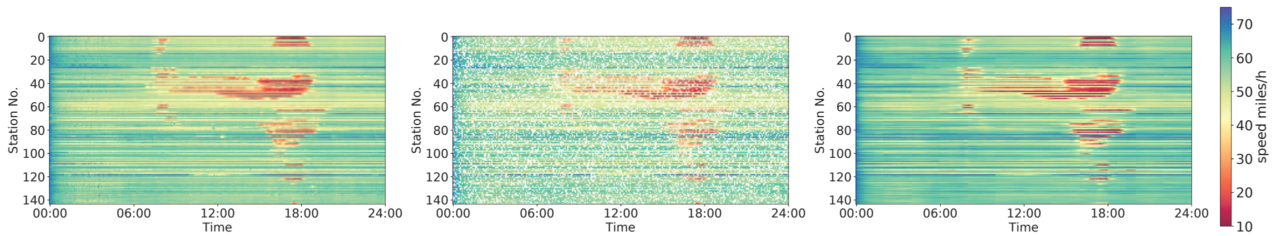
- **Historical Average (HA)**: Missing values are estimated by averaging the observed values in previous days. In this paper, we use the historical average of the past 5 days.
- **k-NN**: k-Nearest Neighborhood performs interpolation by calculating the average value of the nearest k neighboring points and has been applied in traffic data imputation [22]. In this paper, the nearest 4 spatial-temporal neighbors are used for the calculation.
- **Support Vector Regressor (SVR)**: Following the principle of support vector machine, SVR achieves interpolation by finding a hyperplane which minimizes the distance from observed values to the plane [6] .
- **Kriging interpolation**: Kriging is conducted based on a statistical model with the covariance of observed values. Kriging and its improved methods have been applied to traffic data imputation [23].
- **Bayesian Gaussian CP decomposition (BGCP)**: BGCP is a state-of-the-art imputation method based on tensor decomposition. Variational Bayes is adopted to learn the parameters of the model [10].

D. Performance Comparison

We visualize the imputation result on missing Type 1 at 30% missing rate. Fig. 3 shows the general distribution of the input and the output. The blank pixels in Fig. 3b indicate missing points. We can see that the generated output is very close to the ground truth, demonstrating that our model can accurately recover the misses.

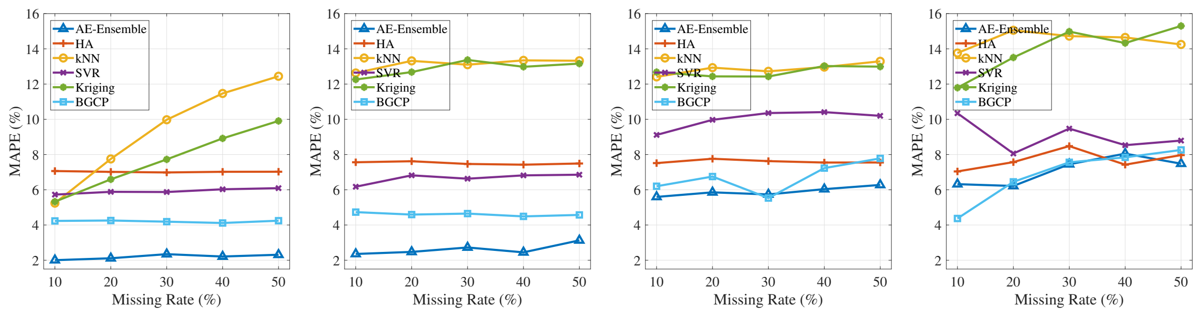
Fig. 4 show the MAPE of five baseline methods and the proposed method evaluated with different missing types on the PeMSD5 dataset, among which the proposed method is labeled by “AE-Ensemble”. From the simulation results and comparisons, the following observations are developed:

[1] http://pems.dot.ca.gov/



(a) Original observations (b) Input with missing points (c) Output

Fig. 3. Imputation results on missing Type 1 at 30% missing rate.



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

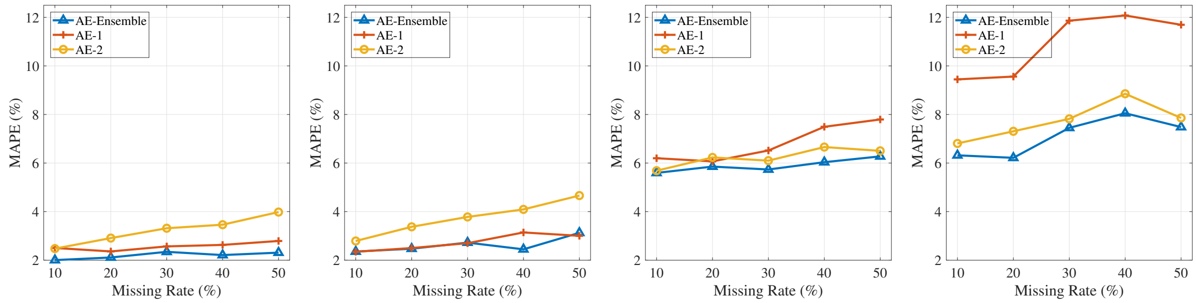
Fig. 4. MAPE of the proposed method compared with baselines for different missing types on PeMSD5 dataset.

- Generally, most methods achieve the highest imputation accuracy in Type 1 missing (c.f. Fig. 1), which is completely random missing and has the lowest accuracy in Type 4 block missing. As an exception, HA performs imputation by averaging previously observed values. Therefore, the performance of HA does not fluctuate with different missing types and rates.
- For the same missing type, the proposed method remains stable performance when the missing rate increases, while other interpolation-based methods decrease rapidly. The reason is that the proposed model aggregates autoencoders by self-adapting weight, which significantly enhances the robustness.
- Among the four different missing types, the proposed method achieves better performance in Types 1 and 2. Specifically, the MAPE of the proposed method outperforms BGCP by nearly 50% (error reduced from approx. 4% to 2%) in PeMSD5. Missing points are randomly scattered or spatially continuous distributed in Types 1 and 2, which means that the missing points can be estimated by temporally neighboring values. This observation indicates that the convolution and deconvolution layers in the proposed method have advantages in presenting temporal adjacency characteristics.
- For Types 3 and 4, the proposed method does not perform as well as the others. Nonetheless, the proposed method still introduces improvement in Type 3 over BGCP (from 7% to 6%) and performs similarly in Type 4. As the missing points are continuously distributed along the time-axis in these types, this observation suggests that the proposed model cannot handle the other types when the observed values can not provide sufficient temporal correlation among the raw data.

E. Model Analysis

In this work, we adopt an ensemble convolutional autoencoder for traffic imputation, which is composed of two standalone convolutional autoencoders. In this subsection, we investigate how each of the neural network impacts imputation accuracy by training and testing them individually. In accordance with the introduction in Section II-C, the two autoencoders are marked as "AE-1" and "AE-2", whose input is X^1 and X^2, respectively. The training and testing in this case study are kept identical to the configurations introduced in Section III-A.

Fig. 5 shows the MAPE of two autoencoders and the proposed ensemble model evaluated with different missing types on the PeMSD5 dataset. We have the following conclusions from the comparison. First and foremost, the proposed ensemble method achieves the best performance in all missing types and missing rates. This observation accords with the intuition that more temporal information is provided to the learning system, leading to better latent information extraction. When comparing the two standalone autoencoders, AE-1 achieves better performance in Types 1 and 2, and AE-2 achieves better performance in Types 3 and 4. These observations indicate that filling the missing points with historical average values gives auxiliary temporal information in Types 3 and 4, in which the missing points are continuously distributed in time series. However, it may bring interference information that compromises imputation accuracy when the missing points are scattered or spatially continuous in Types 1 and 2. The self-adapting weight values combine the advantage of two autoencoders in the proposed



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

Fig. 5. MAPE of two autoencoders and the propose ensemble model for different missing types on PeMSD5 dataset.

ensemble. To sum up, compared with a single autoencoder, the proposed ensemble achieves not only better performance but also robustness in various missing data scenarios.

IV. Conclusions

In this paper, we propose a traffic data imputation method based on a novel ensemble convolutional autoencoder. The method first embeds the traffic data into a two-dimensional matrix according to the spatial-temporal characteristics, and then implements data imputation using an autoencoder consists of convolution and deconvolution layers. In addition, we use historical average values that provide auxiliary information in the data pre-processing. Missing points are respectively filled with zeros and historical average values to construct different input feature maps. At last, the autoencoders trained by different input feature maps are aggregated by a self-adapting weight value. To evaluate the proposed method, we conduct comprehensive experiments on four different missing data scenarios. The results demonstrate that the proposed method can achieve better imputation accuracy and robustness compared with baseline approaches. An ablation test is also carried out to validate the effectiveness of the ensemble over single neural networks.

References

[1] C. Gong and Y. Zhang, "Urban traffic data imputation with detrending and tensor decomposition," *IEEE Access*, vol. 8, pp. 11 124–11 137, 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] Q. Li, H. Jianming, L. Li, and Z. Yi, "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.

[4] 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.

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

[6] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[7] P. Duan, G. Mao, W. Liang, and D. Zhang, "A unified spatio-temporal model for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3212–3223, 2019.

[8] W. Yin, P. Murray-Tuite, and H. Rakha, "Imputing erroneous data of single-station loop detectors for nonincident conditions: Comparison between temporal and spatial methods," *Journal of Intelligent Transportation Systems*, vol. 16, no. 3, pp. 159–176, 2012.

[9] 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.

[10] 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.

[11] 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.

[12] 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, NeurIPS*, Montréal, Canada, 2018, pp. 6776–6786.

[13] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proceedings of the 35th International Conference on Machine Learning, ICML*, Stockholm, Sweden, 2018, pp. 5689–5698.

[14] Y. Duan, Y. Lv, W. Kang, and Y. Zhao, "A deep learning based approach for traffic data imputation," in *Proceedings of 17th International IEEE Conference on Intelligent Transportation Systems, ITSC*, Qingdao, China, 2014, pp. 912–917.

[15] 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.

[16] W. C. Ku, G. R. Jagadeesh, A. Prakash, and T. Srikanthan, "A clustering-based approach for data-driven imputation of missing traffic data," in *Proceedings of 2016 IEEE Forum on Integrated and Sustainable Transportation Systems, FISTS*, Beijing, China, 2016, pp. 1–6.

[17] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–7, 2019.

[18] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

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

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30th International Conference on Machine Learning, ICML*, Atlanta, GA, 2013, p. 3.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR*, San Diego, CA, 2015.

[22] 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.

[23] B. Bae, H. Kim, H. Lim, Y. Liu, L. D. Han, and P. B. Freeze, "Missing data imputation for traffic flow speed using spatio-temporal cokriging," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 124–139, 2018.