A Clustering-Based Approach for Data-Driven Imputation of Missing Traffic Data

Wei Chiet Ku, George R. Jagadeesh, Alok Prakash, Thambipillai Srikanthan
School of Computer Science and Engineering
Nanyang Technological University
Singapore
[ku0003et, asgeorge, alok, astsrikan] @ ntu.edu.sg

Abstract— **The problem of missing samples in road traffic data undermines the performance of intelligent transportation applications. This paper proposes a data-driven imputation method that exploits the spatial and temporal relationships existing between the traffic flows of multiple road segments that are correlated with each other. The K-means clustering technique is used to group together road segments with similar traffic flow patterns. Next, a deep-learning model based on stacked denoising autoencoders is constructed for each group of road segments to extract their spatial-temporal relationships and use them for imputing the missing data points. Experiments conducted with real traffic data demonstrate that the imputation accuracy of the proposed method is robust under different missing data rates.**

Keywords—Traffic data, imputation, clustering, deep learning, intelligent transportation systems.

I. Introduction

In recent times, advances in Intelligent Transportation Systems (ITS) technologies have allowed transport authorities to collect vast amounts of traffic surveillance data for use in applications such as congestion monitoring, incident detection, traveler information and adaptive traffic control. Traffic data, such as traffic volume and speed, is collected through a number of diverse technologies such as inductive loop detectors, GPS probes and video image detection systems. However, regardless of the technology used, traffic data is often incomplete due to hardware or software malfunctions, power failures and transmission errors. The proportion of missing data is not trivial. For example, in Alberta, Canada, more than half of the highway traffic counts have missing values [1] , while in Beijing, China, the average missing ratio of traffic data is around 10% [2]. The problem of missing traffic data significantly degrades the performance of ITS applications. For instance, traffic forecasting models tend to be less effective or even useless with an incomplete dataset. Similarly, advanced traffic signal control systems need sufficient traffic flow data to generate optimal control schemes. Consequently, it is imperative to fill the gaps in the data with appropriate values through a suitable and effective imputation process.

The conventional approach towards imputing missing traffic data for a road segment is to rely on its historical data or current data from neighboring road segments. This approach has some fundamental limitations. Imputation based on historical data performs poorly in the presence of unusual traffic conditions that deviate from historical norms. Another drawback is that, due to the high degree of variability in the relationships between the traffic parameters of neighboring road segments, it is generally necessary to construct a large number of location-specific models for traffic data imputation.

This paper aims to overcome the above limitations by performing missing data imputation collectively for a group of road segments using a data-driven model. This is achieved by first organizing the road segments in a road network into a number of clusters such that the traffic flows of the road segments within each cluster are highly correlated. The K-means clustering technique is used for this purpose. Subsequently, a model for traffic data imputation is created for each cluster. For this, a deep-learning based neural network model is used. The models are trained to mine the underlying spatial and temporal relationships between the traffic data of road segments within each cluster and utilize these relationships to estimate the missing values.

The rest of this paper is organized as follows: Section II presents related work on methods for imputing missing data. Section III presents the proposed data-driven approach for traffic data imputation. Section IV discusses the experimental results. Concluding remarks are given in Section V.

II. Related work

Many research efforts have been undertaken to address the problem of missing traffic data. The historical data based imputation method fills a missing data point for a location based on the average value of historical data corresponding to the same time interval [3-5]. By utilizing temporal features such as daily periodicity and weekly periodicity, these approaches are easy to implement and perform reasonably well under normal conditions. However, their performance is based on the assumption that similar traffic flow patterns exist over the subsequent days or weeks. This assumption ignores the stochastic fluctuations of traffic flow that vary from day to day. Another related approach involves using time series methods to identify meaningful statistical patterns that can be used to estimate missing data. Autoregressive Integrated Moving Average (ARIMA) and seasonal ARIMA [6, 7] are two such popular techniques. Based on the assumption that historical data of a variable provides an indication of its value in the future [8], time series models are trained by historical observations to estimate missing data in future time intervals.

978-1-5090-2930-3/16/$31.00 ©2016 IEEE

To improve imputation performance, many researchers also consider the spatial–temporal correlations between traffic measurements, since traffic parameters at neighboring locations and time periods generally exhibit similar distribution patterns. These similarities in traffic patterns are exploited by developing a regression function based on nearby (spatially and temporally) observed data. Chen et al. [9] developed a linear regression model to capture the relationship between neighboring sensors, such as loop detectors, in order to impute missing traffic volumes and occupancies. Linear regression models are easy to compute and comprehend, but their performance depends on the availability of neighboring data. Hence, they perform poorly as the data-missing ratio increases. Al Deek et al. [10] proposed a pairwise quadratic model to estimate the missing data. Since their proposed regression expression has to be established for each pair of neighboring sensors, their approach may require considerable effort to impute missing values for a large number of sensors.

The above studies mainly focus on filling the gaps of a single sensor. The models used in them are mostly location-specific and suffer from scalability issues. A method based on Kernel Probabilistic Principle Component Analysis (KPPCA) [11] and a tensor-based method [12] have been proposed to fuse the traffic flow data from multiple locations and impute the missing data in these locations collectively. However, these locations are limited to adjacent, upstream and downstream sensors along a freeway corridor.

III. Proposed method

In the work presented in this paper, a single model is used to impute missing traffic volume data for a group of road segments. Intuitively, for best results, the traffic flows of all the road segments within the group should be correlated with each other. It should be noted that the road segments belonging to the same group need not necessary be connected neighbors that are geographically close to each other. Studies have shown that traffic data is correlated not only in short-distance [17], but also in a large area [18, 19]. By mining the relationships between traffic flows of multiple road segments, including those distant from each other, the imputation is made more robust when data in the immediate neighborhood is unavailable.

To fully exploit the spatial-temporal correlation of traffic flows, K-means clustering is used to group the most correlated road segments in a road network. Subsequently, for each group of road segments, a stacked denoising autoencoders (SDAE) model is built to model their spatial-temporal relationships. After the model is trained in a layer-wise greedy fashion, it is able to collectively estimate the missing data at multiple locations under a unified framework.

A. K-means Clustering

Given a set of objects, the goal of clustering is to classify the data into groups or clusters based on the similarity of objects, so that the intra-cluster object dissimilarity is minimized. K-means is one of the well-known clustering techniques in data mining and pattern discovery.

Let the road network consists of D road segments. For each road segment, traffic volume is reported for a total of T intervals in a given data collection period (e.g. one day). Let $y_{d,t}, 1 \leq d \leq D, 1 \leq t \leq T$, represent the traffic volume of road segment d at time interval t. Let $\boldsymbol{y}_d = \{y_{d,t} | t = 1, \ldots, T\}$ be the set of traffic volume readings corresponding to all T intervals for road segment d.

Using the K-means clustering method, the entire dataset $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_D,\}$ can be classified into K clusters, and each cluster centroid is represented by the mean value in the corresponding cluster. Let $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K,\}$ be the set of K cluster centroids, where $\boldsymbol{c}_k, 1 \leq k \leq K$ represents the centroid of the cluster k. In order to cluster the road segments with similar traffic flow patterns into the same group, a distance measure based on the Pearson correlation coefficient [13] is used. Let $d(\boldsymbol{y}_d, \boldsymbol{c}_k)$ denote the distance measure between $\boldsymbol{y}_d$ and the cluster centroid $\boldsymbol{c}_k$. It is given by

$$d(\boldsymbol{y}_d, \boldsymbol{c}_k) = 1 - \frac{\sum_{t=1}^{T}\left(y_{d,t} - \mu_{y_d}\right)\left(c_{k,t} - \mu_{c_k}\right)}{\sqrt{\sum_{t=1}^{T}\left(y_{d,t} - \mu_{y_d}\right)}^2 \sqrt{\sum_{t=1}^{T}\left(c_{k,t} - \mu_{c_k}\right)}^2} \qquad (1)$$

where $\mu_{y_d}$and $\mu_{c_k}$are the means of all the readings in $\boldsymbol{y}_d$ and $\boldsymbol{c}_k$, respectively, computed as below:

$$\mu_{y_d} = \frac{1}{T}\sum_{t=1}^{T} y_{d,t} \text{ and } \mu_{c_k} = \frac{1}{T}\sum_{t=1}^{T} c_{k,t} \qquad (2)$$

In order to determine the optimal number of clusters K, the silhouette value is used as the clustering validity index. The silhouette value measures how similar a point is to points in its own cluster, compared with points in other clusters [14]. A high silhouette value indicates that a point is well matched to its own cluster, and poorly-matched to neighboring clusters. The optimal number of clusters will be the number for which a clustering structure with the highest mean silhouette value of all data points is obtained.

B. Stacked Denoising Autoencoder (SDAE)

Deep learning models have been proved to be successful in many areas including transportation problems [15, 16]. In this paper, a deep multilayered neural network based on SDAE [17] is used for traffic data imputation. An autoencoder is a neural network that has the same number of output neurons as the number of input neurons. The model is trained to reconstruct the input vector, i.e., the target vector is the input of the model. A denoising autoencoder (DAE) is a variant of autoencoder, which is supplied with corrupted input during the training process. It is trained so that clean data is reconstructed from a corrupted version of it, with the hope that the hidden layer learns the robust features underlying the input data.

Let us assume that after the clustering process, there are p road segments in a cluster k. The traffic volume readings corresponding to all T intervals for all p road segments are concatenated into a training sample $\boldsymbol{x}_i$, which is a set $p \times T$ traffic volume readings. Training samples corresponding to different data collection periods together constitute the complete training dataset $\boldsymbol{X}$, defined as $\boldsymbol{X} = \{\boldsymbol{x}_i | i = 1, \ldots, N\}$, where N is the total number of data samples. For a training sample $\boldsymbol{x}_i$, a random fraction of the entries of the input are masked (set as zero) to generate the corrupted version of the input, $\tilde{\boldsymbol{x}}_i$. Hence,

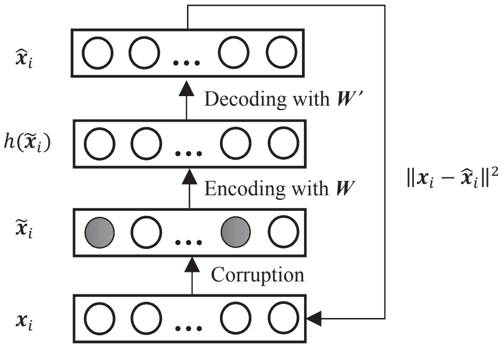
$\widehat{\boldsymbol{x}}_i$
Decoding with $\boldsymbol{W}'$
$h(\widetilde{\boldsymbol{x}}_i)$
Encoding with $\boldsymbol{W}$
$\widetilde{\boldsymbol{x}}_i$
Corruption
$\boldsymbol{x}_i$
$\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2$

Fig. 1. Single-layer denoising autoencoder

$\widetilde{\boldsymbol{x}}_i$ can be seen as the incomplete traffic data with some values missing.

Fig. 1 illustrates the structure of a single-layer DAE. The DAE architecture consists of an encoder and a decoder, which are defined, respectively, by the following mapping:

$$h(\widetilde{\boldsymbol{x}}_i) = f(\boldsymbol{W}\widetilde{\boldsymbol{x}}_i + \boldsymbol{b}_1) \tag{3}$$

$$\widehat{\boldsymbol{x}}_i = f(\boldsymbol{W}'h(\widetilde{\boldsymbol{x}}_i) + \boldsymbol{b}_2) \tag{4}$$

where $f(z) = \frac{1}{1+e^{(-z)}}$ is the nonlinear activation function, $\boldsymbol{W}$ and $\boldsymbol{W}'$are the weights of the encoder and decoder, respectively. Similarly, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the vectors of biases of input and output layers, respectively. With $\widehat{\boldsymbol{x}}_i$ as an approximate reconstruction of input $\boldsymbol{x}_i$, DAE is trained to minimize the reconstruction loss:

$$L_\theta(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_i) = arg\ min_\theta \sum_{i=1}^{N} \|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2 \tag{5}$$

In DAE training, if the number of hidden neurons is the same size or larger than the dimension of the input layer, there is a potential that the DAE will just learn the input identity function without discovering more interesting structure hidden in the data [18]. To prevent this from occurring, sparsity constraints are imposed on the hidden neurons, thereby leading to the following optimization function,

$$minimize\left(\sum_{i=1}^{N} \|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|^2 + \beta \sum_{j=1}^{m} KL(\rho\|\widehat{\rho}_j)\right) \tag{6}$$

where N is the input sample size, m is the hidden layer size, β is the sparsity penalty weight, and ρ is the constant sparsity level. $KL(\cdot)$ is the Kullback-Leibler Divergence metric given as,

$$KL(\rho\|\widehat{\rho}_j) = \rho \log \frac{\rho}{\widehat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\widehat{\rho}_j} \tag{7}$$

$$\widehat{\rho}_j = \frac{1}{N} \sum_{i=1}^{N} h_j(\boldsymbol{x}_i) \tag{8}$$

To learn the complex spatial-temporal relationship of the road segments in the cluster, a deep network is realized by concatenating multiple DAEs to form an SDAE, as shown in Fig. 2. The greedy layer-wise training strategy [19] is applied to train the SDAE. It involves a pre-training step and a fine-tuning step. In the pre-training step, the first layer is trained with raw input and the second layer is trained by taking the first layer's output as the input. The subsequent layers are repeated such that hidden layer of the current DAE is the input layer of the next DAE. Once the pre-training phase is complete, then the whole network is fine-tuned using the classical backpropagation algorithm to refine the parameters. After training, the SDAE model is capable of imputing the missing data of all the road segments in the cluster based on the multidimensional relationships between their traffic volume readings.

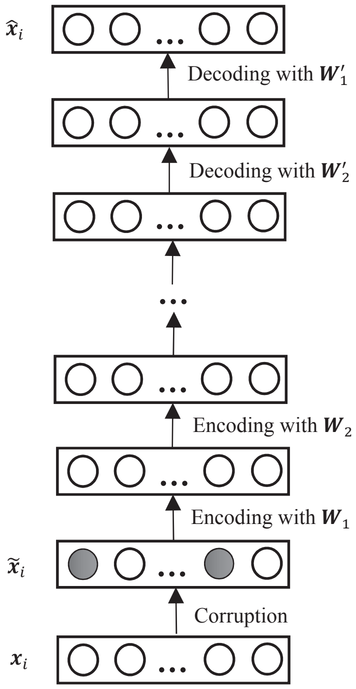
$\widehat{\boldsymbol{x}}_i$
Decoding with $\boldsymbol{W}_1'$
Decoding with $\boldsymbol{W}_2'$
Encoding with $\boldsymbol{W}_2$
Encoding with $\boldsymbol{W}_1$
$\widetilde{\boldsymbol{x}}_i$
Corruption
$\boldsymbol{x}_i$

Fig. 2. Stacked denoising auto-encoder architecture

Fig. 3. The part of the Singapore road network used for the evaluation

IV. Evaluation

A. Dataset

The proposed solution for imputing missing traffic data is evaluated on a dataset obtained from Quantum Inventions [20] a provider of traffic data services in Singapore. This dataset contains traffic flow information collected from 41 road segments in the Jurong West area of Singapore (see Fig. 3) in the first 29 days of November 2014. The traffic flow data is aggregated at 5-minute intervals and results in T = 288 (12 samples per hour in a 24-hour period) data points for the daily flow series of one road segment. The first 23 days of data from this dataset is used for training the models while the remaining data is used as a test set.

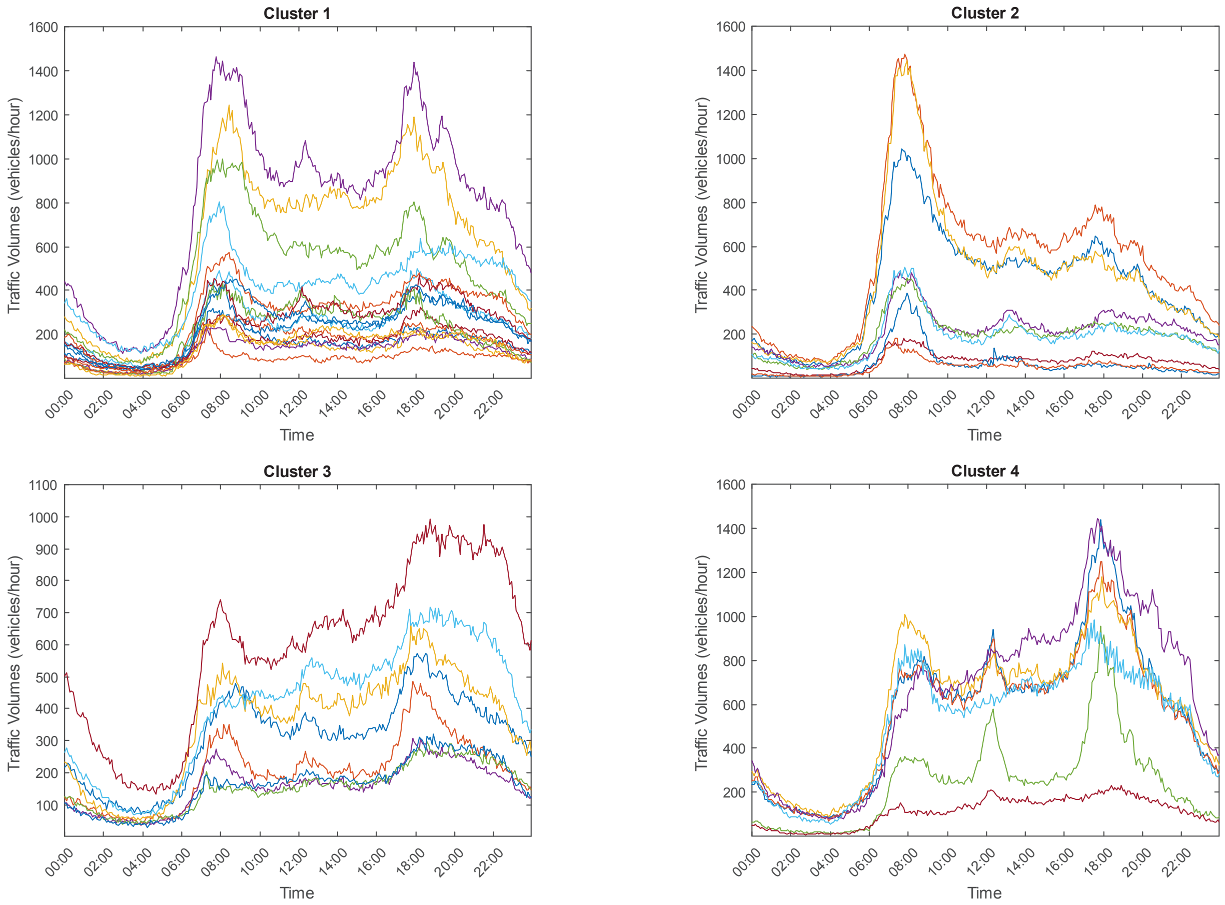

Fig. 4. Daily traffic flow patterns of road segments in each cluster

B. *K-means Clustering Results*

The road segments are grouped into clusters based on their individual average daily traffic flow patterns, which are obtained by taking the average of readings corresponding to each time interval across the training data set. The K-means clustering method uses the distance measure given in (1) to quantify the similarity between the average daily traffic flow patterns of road segments. The optimal number of clusters for the given dataset that maximizes the mean silhouette value is found to be 4. The numbers of road segments in each cluster are 17, 9, 8 and 7, respectively.

The daily traffic flow patterns for each cluster are shown in Fig. 4. As can be observed from the figures, each of the clusters has a distinctive traffic flow pattern. For instance, the road segments in cluster 1 experience a traffic surge during the morning and evening peak hours, while for road segments in cluster 2, the surge occurs only in the morning. On the other hand, road segments in cluster 4 are characterized by heavy traffic flow during the evening peak hours and a distinct spike in traffic around noon. After grouping the road segments that are highly correlated with each other, SDAE model is used to extract the underlying spatial and temporal correlations of their traffic flow patterns and use them to impute the missing traffic data.

C. *Parameter Configuration for the SDAE Models*

Before the training process, a few parameters need to be defined for the SDAE models, such as the number of hidden units in each layer, the size of hidden layer and the sparsity constraints. According to [21], using the same number of hidden units in all the hidden layers generally achieves good results. The number of hidden layers in this work is set as three and the number of hidden units in each hidden layer is set equal to the size of input layer. For all individual DAEs in all SDAE models, the sparsity level ρ and the weight of the sparsity penalty β are empirically identified to be 0.05 and 1, respectively. The architecture details of the SDAE models for each cluster are shown in Table I.

TABLE I. ARCHITECTURE DETAILS OF THE SDAE MODELS

Cluster	Input and Output Size	Number of Hidden Layers	Number of Neurons in Each Hidden Layer
1	17 x 288	3	4896
2	9 x 288	3	2592
3	8 x 288	3	2304
4	7 x 288	3	2016

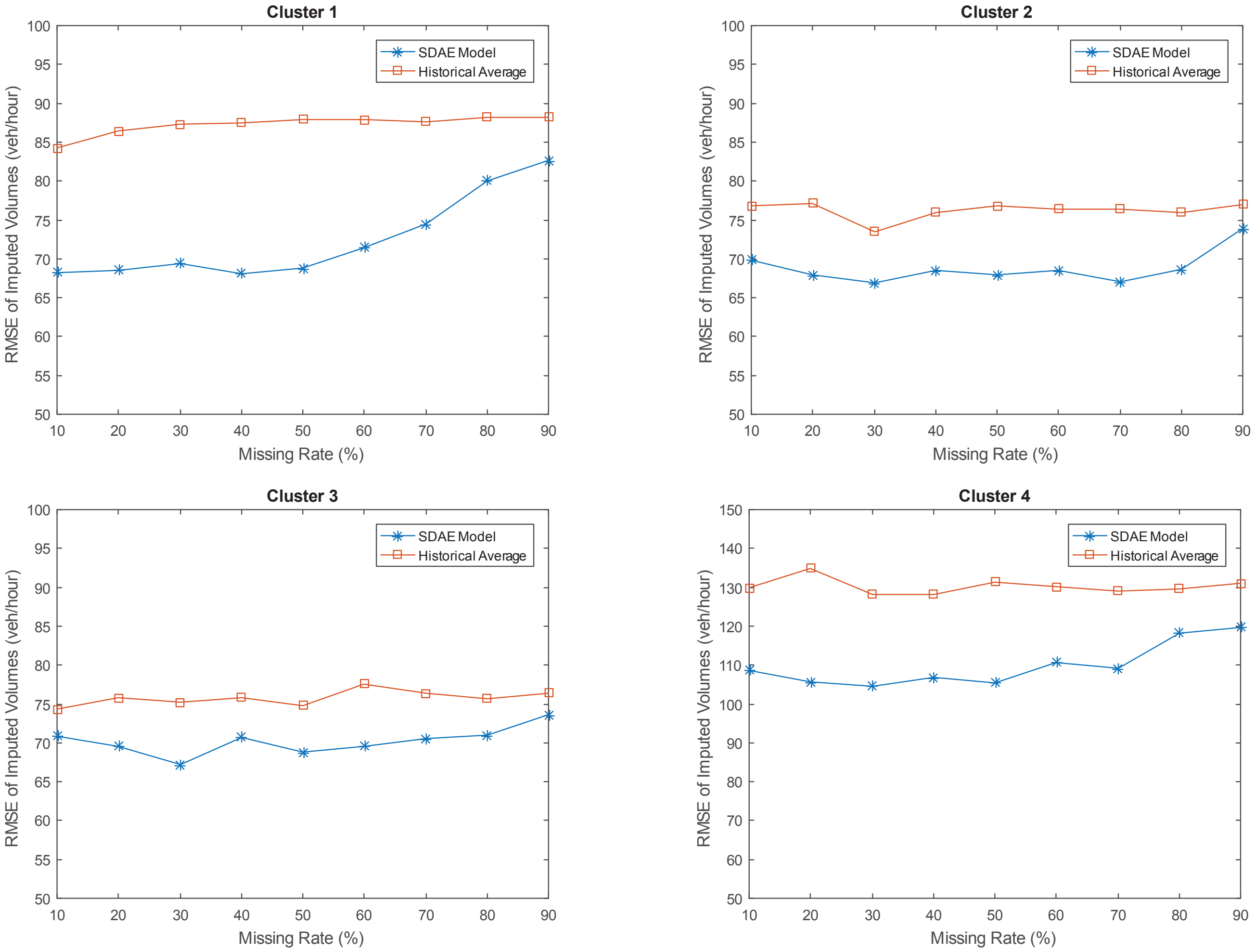

Fig. 5. Evaluation of imputation accuracy

D. *SDAE Model-based Missing Data Imputation Results*

The proposed models are tested on ‘missing completely at random’ (MCAR) pattern, in which the missing data points occur at random and are independent of each other [2]. In order to emulate such a missing pattern, some data in test set are randomly masked to zero according to different missing rates, where the missing rate is defined as the number of the missing points per total number of data points. Experiments are conducted with different missing rates, ranging from 10% to 90%, in 10% increments.

The proposed model is evaluated against the conventional historical average imputation method, in which the missing data for a particular time interval is replaced by the mean of the traffic data belonging to the same road segment at the same time interval throughout a historical dataset. For the experiments in this paper, the training dataset is used as the historical dataset. Root mean squared error (RMSE) is used as the evaluation metric to assess the accuracy of imputation,

$$RMSE = \left[\frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2\right]^{\frac{1}{2}} \tag{9}$$

where n is the total number of the missing points, $\hat{z}_i$ is the i^{th} estimated point and z_i is the corresponding observed value.

The RMSE of the imputed data, for both the historical average imputation method as well as the proposed technique, under different missing rates are shown in Fig. 5. The RMSE of the proposed SDAE-based technique degrades gradually and hence confirms that its accuracy does not reduce significantly with increasing missing rates. On the other hand, the RMSE of the historical average imputation method shows stable behavior since the missing data are filled based on historical data points independent of the current missing rate. The SDAE model outperforms the historical average method in all the four clusters for all the missing rates considered. The RMSE of the historical average imputation method is around 15%, 7%, 6% and 20% higher than SDAE model in cluster 1 to 4, respectively.

As a case study, Fig. 6 depicts the imputation results of one of the road segments in cluster 3 based on the SDAE approach with missing rates of 20% and 80%. The figure shows that the proposed SDAE-based technique is capable of estimating the missing traffic data with reasonable accuracy, despite the stochastic variations of the traffic flow.

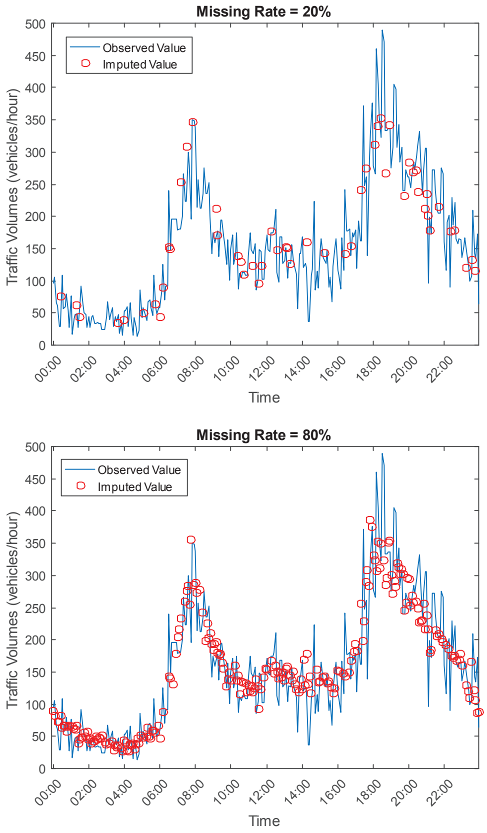

Fig. 6. Imputed values for a road segment under different missing rates

V. CONCLUSIONS AND FUTURE WORK

This paper proposes a data-driven approach based on SDAE models for traffic data imputation for a group of road segments. K-means clustering is used to group road segments that share similar traffic patterns by using a distance measure based on the Pearson correlation coefficient. For each set of road segments, an imputation model is constructed using a deep structural SDAE. Each layer of SDAE is pre-trained using a greedy layer-wise approach and the whole network is then fine-tuned to improve the imputation performance. The SDAE model is able to fill in data gaps at any road segment by using the traffic flow relationships with other road segments in the same cluster. Experimental results based on real traffic volume data show that the imputation performance of the proposed method is robust under high missing rates.

As part of future work, the effectiveness of the proposed method on road networks that are larger and complex than the one used in the current work will be evaluated. The method's suitability for imputing other traffic parameters such as speed will also be investigated. Its performance will also be benchmarked against other state-of-the-art imputation techniques that utilize both spatial and temporal correlations of traffic flows. Future research efforts will also be directed towards determining the optimal parameters of the SDAE architecture, such as the number of hidden layers and the number of neurons per layer, in order to maximize the imputation accuracy.

REFERENCES

[1] M. Zhong, P. Lingras, and S. Sharma, "Estimation of missing traffic counts using factor, genetic, neural, and regression techniques," *Transportation Research Part C: Emerging Technologies,* vol. 12, pp. 139-166, 2004.

[2] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: a systematical approach," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 10, pp. 512-522, 2009.

[3] B. L. Smith, W. T. Scherer, and J. H. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1836, pp. 132-142, 2003.

[4] S. Sharma, P. Lingras, and M. Zhong, "Effect of missing value imputations on traffic parameters estimations from permanent traffic counts," *Transportation Research Board,* vol. 1836, pp. 132-142, 2003.

[5] L. N. Nguyen and W. T. Scherer, "Imputation techniques to account for missing data in support of intelligent transportation systems applications," Center for Transportation Studies, University of Virginia Charlottesville, VA, 2003.

[6] M. Zhong, S. Sharma, and P. Lingras, "Matching patterns for updating missing values of traffic counts," *Transportation Planning and Technology,* vol. 29, pp. 141-156, 2006.

[7] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal of transportation engineering,* vol. 129, pp. 664-672, 2003.

[8] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*: John Wiley & Sons, 2015.

[9] C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya, "Detecting errors and imputing missing data for single-loop surveillance systems," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1855, pp. 160-167, 2003.

[10] H. Al-Deek, C. Venkata, and S. Ravi Chandra, "New algorithms for filtering and imputation of real-time and archived dual-loop detector data in I-4 data warehouse," *Transportation Research Record: Journal of the Transportation Research Board,* pp. 116-126, 2004.

[11] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transportation research part C: emerging technologies,* vol. 34, pp. 108-120, 2013.

[12] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial–temporal correlation," *Physica A: Statistical Mechanics and its Applications,* vol. 446, pp. 54-63, 2016.

[13] T. Warren Liao, "Clustering of time series data-a survey," *Pattern Recognition,* vol. 38, pp. 1857-1874, 2005.

[14] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis* vol. 344: John Wiley & Sons, 2009.

[15] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 15, pp. 2191-2201, 2014.

[16] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 16, pp. 865-873, 2015.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research,* vol. 11, pp. 3371-3408, 2010.

[18] A. Ng, "Sparse autoencoder," *CS294A Lecture notes,* vol. 72, pp. 1-19, 2011.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science,* vol. 313, pp. 504-507, 2006.

[20] *Quantum Inventions* Available: http://cms.quantuminventions.com/

[21] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, ed: Springer, 2012, pp. 437-478.