Applied Intelligence (2020) 50:860–877
https://doi.org/10.1007/s10489-019-01560-y

Data-driven missing data imputation in cluster monitoring system based on deep neural network

Jie Lin[1] · NianHua Li[1] · Md Ashraful Alam[1] · Yuqing Ma[1]

Published online: 19 October 2019

Abstract

Due to cluster instability, not in the cluster monitoring system. This paper focuses on the missing data imputation processing for the cluster monitoring application and proposes a new hybrid multiple imputation framework. This new imputation approach is different from the conventional multiple imputation technologies in the fact that it attempts to impute the missing data for an arbitrary missing pattern with a model-based and data-driven combination architecture. Essentially, the deep neural network, as the data model, extracts deep features from the data and deep features are further calculated then by a regression or data-driven strategies and used to create the estimation of missing data with the arbitrary missing pattern. This paper gives evidence that if we can train a deep neural network to construct the deep features of the data, imputation based on deep features is better than that directly on the original data. In the experiments, we compare the proposed method with other conventional multiple imputation approaches for varying missing data patterns, missing ratios, and different datasets including real cluster data. The result illustrates that when data encounters larger missing ratio and various missing patterns, the proposed algorithm has the ability to achieve more accurate and stable imputation performance.

Keywords Missing data · Cluster monitoring · Deep belief networks · Multiple imputation

1 Introduction

Clusters not only making an accurate decision but also monitoring with partial missing cluster data is the practical problem. Systems are becoming increasingly complex due to the large number of services and resources. The cluster monitoring system is what we use in a systematic way to process or analyze cluster data at a remote location under normal circumstances. It is quite necessary for the cluster system to address many threats that may arise in systems by providing a statistic overall summary, especially a detailed view of computing resources.

Collecting, analyzing and drawing inference from cluster data are three primary procedures in a cluster monitoring system [1]. This cluster can be a sensor cluster or computer cluster. Unfortunately, for any number of reasons such as single point of failure or network unreliable, it is rarely possible to reliably collect the intended data for all nodes in a cluster environment. It means that data values intended by monitoring design to be observed are in fact missing. The ubiquity missing data not only means low performance for monitoring decision but also many traditional data analysis applications that depend on good access to accurate data cannot be immediately used in the system [2, 3]. The ability to manipulate missing data has become a fundamental requirement for classification, regression, and time series prediction problems [4]. Therefore, processing missing data among the original data in order to get an unbiased analysis result becomes a primary problem in the cluster monitoring research area.

A simple approach to deal with missing data is to delete them, and it is called list-wise deletion method [5]. The disadvantage of the method is that it may result in a significant loss of statistical information and precision

✉ Jie Lin
linjie@uestc.edu.cn

NianHua Li
linianhua@uestc.edu.cn

Md Ashraful Alam
Alamma@uestc.edu.cn

Yuqing Ma
Mayuqing@uestc.edu.cn

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

under a complex multivariate analysis [2]. On the other hand, to relieve the impact of missing data is to use missing data imputation technique. The main idea of those methods for dealing with missing data is using an approach to fill them in and maintain the original distribution of the data as approximate as possible, so that standard methods that have been developed to analyze complete data sets can be applied [6]. Many imputation methods have been proposed in statistics, mathematics and other various disciplines. In general, they can be classified into two classes:

1) Single Imputation (SI): Single imputation approaches, such as mean/mode substitution, dummy variable method, and single regression, aim to fill a single value for missing observation. One of its intrinsic disadvantages is that it reduces variability, resulting in biased estimates or the uncertainty associated with the model used for imputation.
2) Multiple Imputation (MI): Instead of imputing a single value for each missing data, multiple imputation creates many completed candidate datasets about the missing data case, and then combines these candidate datasets into one estimate for the missing data [7]. Multiple imputation does not attempt to give an accurate estimate for the missing data, but rather to represent a random sample of the missing data which constructs the valid statistical inferences that properly reflect the uncertainty due to missing data [3]. Hence, it retains the advantages of single imputation while allowing the data analysts to obtain valid assessments of uncertainty.

In the last decades, a large number of multiple imputation methods have been proposed and some of them will be discussed in the Section 2. In general, in a data imputation procedure, some experience and knowledge about the missing pattern of the original data is required so that we can choose an available imputation method according to the type of the missing data pattern. However, in real-world data analysis applications that face massive volume of data, at the same time many missing data patterns may exist. In addition, as the volume of data rapidly growing, the effects of these traditional methods reduce.

In this paper, we focus on resolving the partial data missing problem in data preprocessing part of a cluster monitoring system, with arbitrary missing data patterns. The deep neural network shows the capability for modelling complex structures and dependencies in the data. Imputation of the missing data on features extracted from data by deep neural network may be better than the traditional methods which directly analyze on original data. This adantage motivates us to combine the deep neural networks into multiple imputation framework. Firstly, we investigate a model-based multiple imputation algorithm for monotone missing data pattern by using deep neural networks to generate multiple estimations of the missing data. We show that the deep neural network has the ability to accurately model missing data. In addition, we extend the ability of this method to deal with the various missing data pattern by constructing a new data-driven imputation model to build filling candidates that will be fused with a top k nearest neighbors' weighted matrix and output the final fill values of that missing data. Finally, we construct a hybrid MI system (HMI) with the proposed two methods for overcoming missing data imputation problem with huge data volume, large missing ratio and arbitrary missing data pattern. Our experimental results prove that if we can train a deep neural network to construct the deep features of data, imputation based on deep features is better than that directly on original data. We construct a new Hadoop cluster monitoring system by applying HMI to recover the missing node data before they are input into the traditional decision module. This new system has shown the ability to handle partial data missing problems and restore the node data.

The rest of this paper is organized as follows. Section 2 presents the related work on missing data imputation. Section 3 illustrates the background of multiple imputation and deep learning, followed by the proposed method in Section 4. Section 5 provides the experiments and discussions, and finally, Section 6 concludes the paper and introduces future work.

2 Related work

One of the classic missing data processing methods in the monitoring system is the missing data imputation. As an example, Zhang and Liu [8] applied least squares support vector machines (LS-SVMs) to predict missing traffic flow in an intelligent transportation monitoring systems. Suh et al. [9] proposed to use imputation technology into remote congestive heart failure monitoring system for predicting the missing sensor data.

The missing data imputation technology recovers incomplete data by generating an estimation of missing data to create a "completed" data set, which will be further fed into the following learning and analysis applications. In last few decades, a number of methods were developed for the imputation of missing data. Some of them were reviewed by Allison and Paul [10].

According to the number of imputed values, imputation methods can be divided into two categories: single imputation and multiple imputation, as described above. Also, considering the model constructing approaches used for imputing data, these technologies mostly can be classified into statistical-based and model-based.

Hot deck is a widely used statistical-based method. Srebotnjak et al. [11] demonstrated that the hot-deck imputation method can get better inform decision makers on the types and extents of water quality problems in the context of limited globally comparable water quality monitoring data. Turrado C.C et al. presented the missing data imputation method based on multivariate adaptive regression splines for handling missing data in electrical data loggers and showed that the proposed method outperformed the multivariate imputation by chained equations [12].

Imputation methods based on model strategy learn the predictive models from the available information in the data sets, and these models are then used to estimate absent values. Approaches such as multi-layer perception (MLP), k-nearest neighbors (KNN), self-organi-zing maps (SOM) and decision tree (DT) construction algorithms were commonly used for learning models [5, 13, 14]. Recently, the deep neural networks were also applied for modeling missing data. Duan et al. [15] proposed an approach based on deep learning networks to impute the missing traffic data. The deep learning network approach discovers the correlations contained in the data structure by a layer-wise pre-training and improves the imputation accuracy by conducting a fine-tuning afterwards. Che et al. [16] proposed to capture the long-term temporal dependencies of time series observations as well as utilizes the missing patterns for improving the prediction results by incorporating masking and time interval into a deep model architecture. The difference with our model is that above methods predict the missing data directly by a deep neural network.

Recently, many researchers have done a lot of studies in missing data imputation. Thirukumaran and Sumathi [17] utilized well-known classifiers, such as LSVM, RIPPER, C4.5, SVMR, SVMP and KNN to improve the imputation accuracy, and finally found that the mean method by step digression imputation method is better among all other methods. A fuzzy-neighborhood density-based clustering technique was used in literature [18] to group the similar patterns and find the best donors for each incomplete target pattern in the imputation system. Azim S and Aggarwal [19] implemented a 2-stage hybrid model for filling in the missing values, which used fuzzy c means and multi layer perceptron.

In 2017, based on the random forest algorithm(RF), Tang and Ishwaran [7] revealed RF imputation to be generally more robust performance with increasing correlation. Nikfalazar et al. [2] and Soni and Sharma [20] both made some improvement on the imputation data with fuzzy clustering. In addition, Myneni et al. [21] presented a framework for correlated cluster-based imputation to improve the quality of data for data mining applications. Another work in literature [22] handled the missing data by using dynamic bayesian network and support vector regression algorithm is used for predicting the filling values.

In 2018, Chen [23] proposed a missing values imputation method by combining sample self-representation strategy and underlying local structure of data in a uniformed framework. The evidence chain approach [24] was also applied to mine all relevant evidence of missing values and build the further estimation of missing values. Moreover, Zhao et al. [25] developed a novel local similarity imputation method that estimates missing data based on fast clustering and top k-nearest neighbors, and in order to improve the imputation accuracy, a two-layer stacked autoencoder combined with distinctive imputation is applied to locate the principal features of a dataset for clustering. Tsai et al. [26] introduced a class center based missing value imputation approach to produce effective imputation results more efficiently based on measuring the class center of each class and then the distances between it and the other observed data are used to define a threshold for the later imputation. In addition, literature [4] presented a new approach based on an extension of the incremental neuro-fuzzy gaussian mixture network, using an approximated incremental version of the expectation maximization algorithm, to carry out the imputation process of the missing data during the execution of the recalling operation in the network layer.

The difference with our model is that above methods predict the missing data directly and individually by a neural network, cluster or regression model. In our task, the deep neural networks are used to extract the deep features and imputation data estimation is implemented on the deep features by both regression and data-driven strategies. Hence, the performance of these methods should be lower than our hybrid method, described in Section 4.

3 Preliminaries

3.1 Multiple imputation

The Multiple Imputation framework consists of a two-step processes, as described in literature [27]:

- Firstly estimating M "complete" data sets candidates.
- The M data candidates are pooled into one estimation for the missing data.

Figure 1 depicts the common Multiple Imputation architecture. This architecture is also used in our proposed methods.

Many types of missing data patterns and analysis models can be handled within this framework, making it presently, the best option for dealing with most missing data problems.

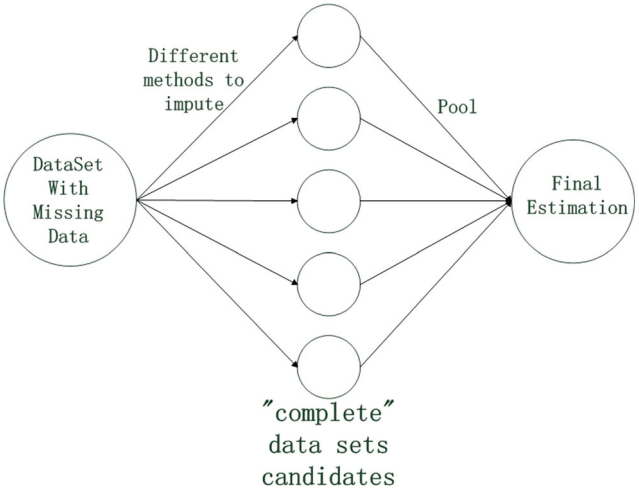


Fig. 1 The brief architecture and mechanism of the Multiple Imputation(MI)

Facing different types of missing data patterns, we should choose various analysis models within the first step of MI.

For the monotone missing data pattern, the missing variables can be ordered in such a way that once a case has a missing data on one observation it is then subsequently missing on everything else [6], such as in Fig. 2a.

Imputation tasks are relatively easier if the missing pattern is monotone either a parametric regression method that assumes multivariate normality or a nonparametric method that uses propensity scores is appropriate to apply in monotone missing condition [5].

However, the missing data do not satisfy the requirement of monotone missing pattern in most real cases, and usually appear a kind of arbitrary missing data pattern. An arbitrary missing data pattern is that the missing data of variables is a random distribution in one observation [6], shown in Fig. 2b.

Analyzing this data with arbitrary missing pattern is more difficult, ultimately special procedures are required. A Markov Chain Monte Carlo [28] method, which creates multiple imputations by using simulations from a Bayesian prediction distribution for normal data. It is often used to solve arbitrary missing data pattern.

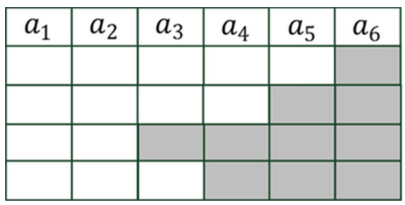
3.2 Deep belief networks

Deep learning, mentioned in [29] neatly, is a part of machine learning which has been used in various aspects.

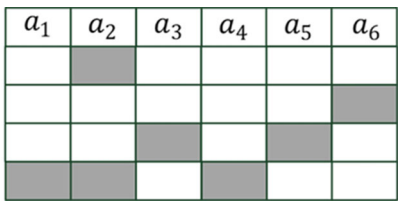
Bengio, Courville and Vincent [30] reviewed the works in the area of unsupervised feature learning and deep learning, covering advances in probabilistic models, auto-encoders, manifold learning, and deep networks.

The Deep Belief Networks(DBNs) was mentioned first by Geoffrey Hinton [32] in 2006. It with many hidden layers is appeared capable of modelling complex structures and dependencies in the data. Hence the DBNs has been widely applied in the feature extracting stage recently. By training the weights between neurons, the whole neural network generates training data with maximum probability. In 2011, Hinton and Alex Krizhevsky [33] used DBNs to obtain semantic codes which mean a high-level expression of image features for image retrieval. And in 2015, Mehdi Hajinoroozi et al. [34] explored the cognitive states prediction based on DBNs for effective features extraction. Besides, Zhikui Chen et al. [35] proposed a data imputation method which makes DBNs remove the noise brought by incomplete data and extract high quality features. In our imputation methods, we implement the missing data imputation with model-based technology. We will apply Deep Belief Networks within the first step of MI for extracting the representative features and then creating M "complete" data sets. In this section, we first give a brief theoretical background of Deep Belief Networks.

The DBNs was built by using multi Restricted Boltzmann Machines (RBMs). A typical net structure of DBNs is shown in Fig. 3, A DBNs consists of multiple layers of neurons, divided into dominant neurons and recessive neurons. Dominant neurons are used to receive input, and recessive neurons are used to extract features, so recessive neurons are also called feature detectors. The connection



(a) monotone missing data pattern



(b) arbitrary missing data pattern

Fig. 2 The monotone missing data pattern (a) and arbitrary missing data pattern (b). The shaded cells are missing data

between the top two layers is undirected and constitutes an associative memory. Only the upper and lower directional connections are connected between the lower layers. The bottom layer (i.e. visiable layer) represents the data vector, and each neuron represents one dimension of the data vector. As mentioned earlier, the composition of the DBNs is a layered structure, and the process of training the DBNs is carried out layer by layer. In each layer, the data vector is used to infer the hidden layer, and this hidden layer is used as the data vector of the next layer.

First, regardless of the two layers that constitute the associative memory at the top, a DBNs connection is guided by the top-down generated weights. An RBM is just like a building block, and it's easy to connect weights for learning. In the beginning, RBM pre-trains the weights of the generated model through an unsupervised layer-by-layer greedy method. This method is called a contrastive divergence in Geoffrey Hinton's paper, and the author proved its validity. During the training phase, a vector is generated at the visible layer, passing the value to the hidden layer. In turn, the input to the visible layer is randomly selected to attempt to reconstruct the original input signal. Finally, these new dominant neurons will forward the reconstructive recessive neurons to obtain a hidden layer. In the training process, the vector value of the visible layer is firstly mapped to the hidden layer, then the visible layer is reconstructed by the hidden layer, and the new visible layer is mapped to the hidden layer again, which gives a new visible layer. This repeated step is the Gibbs sampling. The correlation difference between the input vector of the visible layer and the hidden layer is the main basis for the weight update.

In DBNs, it uses RBMs as the base where the standard type of RBM has binary-valued (Boolean/Bernoulli) hidden and visible units such as Fig. 4, and consists of a matrix of weights $W = (w_{ij})$ associated with the connection between hidden unit h_j and visible unit v_i, as well as bias weights a_i for visible units and b_j for hidden units. For example, for

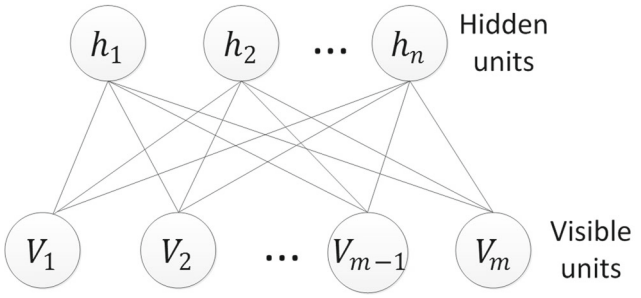


Fig. 4 The structure of RBM

an already trained RBM, the weight between each recessive neuron and dominant neuron is represented by the matrix W.

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N,1} & w_{N,2} & \cdots & w_{n,m} \end{bmatrix} \tag{1}$$

Where $w_{i,j}$ represents the weight from the $i-th$ dominant neuron to the $j-th$ recessive neuron, m represents the number of dominant neurons, and n represents the number of recessive neurons. When we assign a new data $x = \{x_1, x_2, \cdots, x_m\}$ to the visible layer, RBM will decide to turn on or off recessive neurons according to weight W.

The specific operation is to firstly calculate the excitation value of each recessive neuron:

$$h_j = W_j.x + b_j \tag{2}$$

Here we need to use the conditional independence between the neurons mentioned earlier. Then, the excitation values of each recessive neuron are normalized by the sigma function to become the probability value of their open state:

$$P(h_j = 1) = \sigma(h_j) = \frac{1}{1 + e^{-h_j}} \tag{3}$$

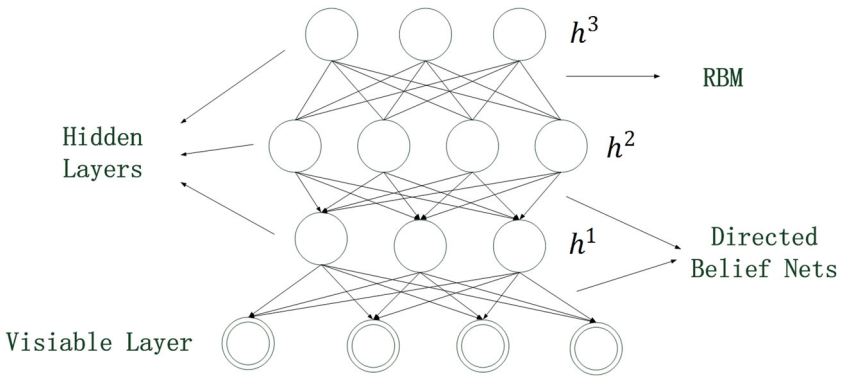


Fig. 3 The structure of a 4-layer DBNs. It is composed of a visible layer and three hidden layers

So we calculate the probability that each recessive neuron h_j is turned on, and the probability of being in the off state is complement:

$$P(h_j = 0) = 1 - (h_j = 1) \tag{4}$$

So in the end whether the neuron is turned on or off, we need to compare the probability of opening with a random variable $u \sim U(0, 1)$ extracted from the uniform distribution of 0-1.

$$h_j = \begin{cases} 1 & P(h_j = 1) \geq u \\ 0 & P(h_j = 1) < u \end{cases} \tag{5}$$

let (5) decide whether to turn on or off the corresponding recessive neurons like that. The calculation in the hidden layer is similarly as that in visible layer.

The model combines energy function and canonical distribution to give individual activation probabilities as

$$P(h_j|v, w) = f(\sum_{i=1}^{I} w_{ij} v_i + b_j) \tag{6}$$

$$P(v_i|h, w) = f(\sum_{j=1}^{J} w_{ij} h_j + a_i) \tag{7}$$

where the function $f(\cdot)$ is the activation function. It can be a Sigmoid or Relu function. It is worth noting that the neurons inside the visible layer and the hidden layer are not interconnected, only the neurons between the layers have symmetric connections. This approach has the advantage that, given the values of all dominant neurons, the value of each recessive neuron is irrelevant. Similarly, when a hidden layer is given, the values of all the visible layers are not related to each other. In the acutal training, the corresponding parameter updating process can be shown as

$$\begin{aligned} \triangle w_{ij} &= \triangle w_{ij} + [p(h_j = 1|v^{(0)})v_i^{(0)} - p(h_j = 1|v^{(k)})v_j^{(k)}] \\ \triangle a_i &= \triangle a_i + [v_i^{(0)} - v_i^{(k)}] \\ \triangle b_j &= \triangle b_j + [p(h_j = 1|v^{(0)}) - p(h_j = 1|v^{(k)})] \end{aligned} \tag{8}$$

To train an RBM, it is actually to find a probability distribution, so that the probability of generating training samples is the largest, which is determined by the weight, and the goal is to find the best weight, which can be derived by using the maximum log likelihood function. Training will use the famous Contrastive Divergence algorithm to learn [31], and we will describe the process of weight convergence to the optimal through the $i - th$ epoch of the process of weight update in algorithm 1.

Algorithm 1 One epoch of RBM training process.

For the input v_i of visible layer:
It firstly calculate the probability that it will cause the hidden layer neurons to be turned on with (6).
Then, from the calculated probability distribution above it can extract a sample h_j ,which is:

$$h_j \sim P(h_j|v, w) \tag{9}$$

Afterwards, it can reconstruct visible layer with (7).
Similarly, extract a sample v_j of the visible layer, which is:

$$v_j \sim P(v_j|h, w) \tag{10}$$

Recalculating the probability of the hidden layer neurons being turned on again using the reconstructed visible layer neurons based on (6).
Update the weight according to the (8).
Then, it successfully updates the weight within this epoch. When the $\triangle w_{ij}$ is below a tiny value in many epoches, it can approximate that the weight has converged and the training process is over.

Such trained RBM can accurately extract the features of the visible layer, and can also restore the visible layer according to the features represented by the hidden layer.

4 Method

Based on the architecture of Multiple Imputation(MI), we attempt to propose a hybrid MI system, as a complement of model-based MI technologies with partial incomplete data. This hybrid MI framework consists of two MI methods, identically each method uses deep neural network within the imputation procedure, however, the designed function of the network models is not similar. These two methods cover the monotone and arbitrary missing pattern respectively.

Assume that the missing data is not independent from some other available variables (i.e., MAR condition) and the first method, called "Features Regression", pre-trains M deep neural networks and applies the output of these deep neural networks to learn M regression machines by using M complete data sample sets within the overall input data. In the first step in MI, these M regression machines estimate M candidates of the missing data and these candidates can be further evaluated and pooled into an unbiased estimation for that missing data. The same as traditional regression MI methods, this method also can be effectively applied to resolve monotone missing pattern, but neither can suit to arbitrary missing pattern.

Further, we propose an imputation method to handle missing data involved in the arbitrary missing pattern. "Data Driven Imputation", a new MI architecture, is formed by combining the model-based and data-driven strategies.

This method selects M reference sample sets, each sample set includes N complete samples randomly sampled from the input analysis data set. In each reference sample set, these N reference samples are classified into C number

of clusters. Thus the number of samples in each cluster is equal to N/C. Before the imputation process, M deep networks are pre-trained with M reference sample sets and $M \times N$ features are calculated by these M deep networks. In imputation task, we extract M deep features for the input data with partial missing values also by using these pre-trained M deep networks.

The $j-$th, $\{j = 1, 2, ..., M\}$ deep feature for the input data is sorted to one of C clusters belong to $j-$th reference sample set. Then k nearest data candidates from this cluster are chosen by measuring the distance between the feature and associative features of the reference samples. These k nearest candidate estimations are further fused into one estimation for $j-$th deep feature with a weight matrix. Overall M estimations from M deep features consist the M fill candidates for the missing data. Because these imputation estimations are generated directly from k nearest reference sample data, it means the imputation result values are driven by the reference sample data. Therefore, we call this is a data-driven strategy.

Choosing the samples based on the deep features level makes this method more robust and accurate compared with traditional data-driven methods that directly get the evaluations on the original data or statistics information of the original data. Any missing data conditions can be imputed by this data-driven method and thereby it can be used to deal with imputation tasks with arbitrary missing pattern.

We combine the first and second methods to construct the new hybrid MI system for providing higher accurate and efficient missing data imputation with different missing patterns. The detail of these methods will be described in the following subsections.

4.1 Data preparing procedure

Before describing these methods, we should first give the detail of the data used in our systems and the experiments that follow. Assume K-sample analysis data sets $X = \{x^1, x^2, x^3, \cdots, x^K\}$, and each sample x^i has m attributes $(a_1^i, a_2^i, \cdots, a_m^i)$.

All the data set is separated into two subsets, Co and Im, Co notes the complete dataset, where all attributes can be observed in the set (no missing data), and Im is the incomplete dataset, where partial missing data can be found). The intersection of Co and Im is null.

The Co subset is used in three parts: pre-training the DBNs, training regression machines and as the reference dataset for the data-driven. During the experiments, the DBNs is used as the feature extracting tool and pre-trained before imputation process. In each training the dropout strategy is applied, so some samples in Co subset are partial replaced by zero value to let the DBNs can finish fault-tolerant feature extraction when the input data suffers from partial missing. The Im is used for all imputation performance testing. When the pre-trained DBNs is used to extract features of samples in Im, that missing value parts will be replace by zero.

Both missing data patterns, monotone and arbitrary missing data pattern, exist in the subset Im. As Fig. 2 shows, it is said to have a monotone missing pattern when an attribute a_j^i is missing for the individual $i-$th sample that all subsequent attributes $a_{j'}^i$, $j' > j$, are all missing for the sample. Arbitrary missing data pattern is where attributes missing for the individual $i-$th sample is random. We further artificially segment the incomplete data set Im into two subsets, according to the prior knowledge about the dataset. Subset $Mo \subset Im$ consists of missing data with monotone missing pattern, and subset $Ar \subset I$ represents arbitrary missing data pattern.

4.2 The first MI method - features regression

Assume each missing attribute of a sample is not independent of other available attributes (i.e., MAR), a regression model handles a monotone missing data pattern and makes an estimation for each missing attribute, using the non-missing attributes as covariance. Traditional regression models use original data in the calculation, even though, regression with original data is more sensitive to the increase of missing data ratios than using the features of original data, due to obtaining a high-level abstract from the original data and therefore is more robust to partial missing. For this reason, the first method extracts the features of original data as the input of regression method instead of direct regression on the original data.

This feature extracting implementation by the DBNs is described in the Section 3.2 and the DBNs extracts deep features of the original data, consequently these deep features are fed to the regression method to get the estimations of missing data. In our following experiments these features have showed more robustness to the increase in missing data volume. Moreover, these features can represent original data very well and the dimension of features can be smaller than original attributes, thus helping decrease the complexity of regression calculation.

Assume one of the samples in data set Mo miss two attributes, noted a_{m-1}, a_m, and other $m-2$ attributes can be observed. For detail, we further define this sample as $x_{missing} = (a_1, a_2, \cdots, a_{m-2})$. The imputation method can fill one missing attribute each time. For multiple-missing attributes, overall missing attributes can be imputed one by one. For the monotone pattern, we should first fill the attribute a_{m-1}.

Assume d samples selected randomly from set Co generate the instance set $E = \{x^i | i = 1, 2, \cdots, d\}$, $x^i =$

$(a_1^i, a_2^i, \cdots, a_m^i)$. The training set $T = \{\tilde{x}^i | i = 1, 2, \cdots, d\}$ is constructed from the instance set E by deleting the attributes a_{m-1}^i, a_m^i for each sample in the set E. All deleted values a_{m-1}^i compose the training goal sets $G_{training} = \{g^1, g^2, \cdots, g^d\}, g^i = (a_{m-1}^i)$.

To ensure that subset Co and E have no significant difference in statistical distribution, a T-test with Statistical Product and Service Solutions (by SPSS) between the data sets Co and E [36] are performed. If the result of T-test passes a predefined significant factor, the above procedure will repeat until the result is lower than the predefined significant factor.

The DBNs just need to be trained with the unsupervised learning procedure organized by one input layer and $L_n (\geq 1)$ hidden layers. The dropout strategy is adopted in training procedure. Using algorithm 1 described in Section 3.2 with $\tilde{x}^i$ as the input, the last hidden layer output of DBNs can be defined as (9), where $W = (w^{(1)}, w^{(2)}, \cdots, w^{(L_n)})$ is weight and $b = (b^{(1)}, b^{(2)}, \cdots, b^{(L_n)})$ is bias.

$$v^{(L_n)} = f(W\tilde{x}^i + b) \tag{9}$$

These trained matrices W, b are applied to extract feature set $T_{feature}$ of training set T with (9). The $T_{feature}$ and training goal sets $G_{training}$ are together used to learn a regression model formulated in (10).

$$G_{training} = \widetilde{W} \times T_{feature} + \widetilde{b} \tag{10}$$

The $\widetilde{W}$ and $\widetilde{b}$ in Formula (10) can be trained by a regression method.

When the incomplete record $x_{missing}$ needs to be actually filled, with the learned W, b, $\widetilde{W}$ and $\widetilde{b}$ through the (11) we can predict one of the candidates for one missing variable at each time.

$$O = \widetilde{W} f(W x_{missing} + b) + \widetilde{b} \tag{11}$$

In our multiple imputation framework, we learn M DBNs and M regression models on M training data sets which is a random sample of total Co. All M regression models generate M filling candidates $(O_1, O_2, \cdots, O_M)$, and finally these candidates are fused into an estimation $\bar{O}$ for the missing data a_{m-1}.

The following Algorithm 2 illustrates the whole process of the proposed method for calculating one missing variable.

Algorithm 2 Features regression.

Require:

$T_j, G_{j,training}$: a training data set and the training goal set, randomly sampled on Co $x_{missing}$: an input incomplete data

Ensure:

$\bar{O}$: imputation value

1: **for** $j = 1$ to M **do**
2: Using T_j to train a DBNs and get weight W_j and bias b_j.
3: Calculating features set $T_{j,feature}$ for training set T_j with (11).
4: Learning regression model to obtain a $\widetilde{W_j}$ and $\widetilde{b_j}$.
5: **end for**
6: **for** $j = 1$ to M **do**
7: Obtaining $j-th$ predictions of missing data $x_{missing}$ by (13) with $j-th$ DBNs and regression models.
8: **end for**
9: Pooling M estimations into one final fill value $\bar{O}$ using method proposed in [27].

Note that for multiple-missing variables condition in the monotone missing pattern, the framework above creates imputation values for the missing data consecutively. As a two-missing variables example described in Section 4.2, the missing variable a_{m-1} is firstly estimated and then a new incomplete record $(a_1, a_2, \cdots, \tilde{a}_{m-1})$ joined the estimation of variable a_{m-1} is further used to fill the a_m. The same as filling a_{m-1}, a new training set $\acute{T}$ is constructed on the instance set E by deleting the attributes a_m^i for each record in set E. All deleted values a_m^i compose the new training goal set $\acute{G}_{training}$. The Algorithm 2 is repeatedly used to calculate the estimation of a_m.

The reason why the method is called Feature Regression is that it uses the features from DBNs to construct a regression model and it can solve data missing problem in MAR mechanism with monotone missing pattern. But it has drawback mentioned above that it may not be effective for arbitrary missing data pattern. In the next section, we discuss a further improvement of this method aiming to offer an imputation to arbitrary missing data pattern.

4.3 The second MI method - data driven imputation

We furthermore extend the capability of the first method for resolving arbitrary missing data pattern. The proposed new model, called Data Driven Imputation (DDI), combines the model-based and data-driven strategies.

In this model, DBNs are also used to extract high-level features, different from the Feature Regression model, these extracted features are sent into a proposed data-driven

system for getting the candidates of missing data. The DDI framework for generating imputation candidates for a missing data is illustrated in Fig. 5.

We randomly sample instance set E_j $\{j = 1, ..., M\}$ from All Co. Assume we need to handle a sample with two missing attributes a_3, a_5, which shows arbitrary missing data pattern, such as in Fig. 2. A training set $T_j = \{\tilde{x}^i | i = 1, 2, \cdots, d\}$ is constructed on an instance set E_j by deleting the attributes a_3^i, a_5^i for each record in a set E_j. Using M training data sets in this framework, we pre-generate M DBNs models by algorithm 1 as same as Features Regression method described above. Each training data set is used to train a DBNs. The $j-$th $(j = \{1, 2, \cdots, M\})$ DBNs model is defined as $v_j^{(L_n)}$.

All M DBNs models are pre-learned and then used to extract features of the training set before imputation process, generating M feature matrices. Each matrix F_j, $j = \{1, 2, \cdots, M\}$ can be defined as follows:

$$F_j = \begin{pmatrix} feature_{j,1} \\ \vdots \\ feature_{j,d} \end{pmatrix} \tag{12}$$

$i-$th row in F_j is a feature vector of sample i in training data set T_j, obtained by $j-$th DBNs model.

In (12), the F_j acts as a feature dictionary of training samples. We conduct a K-Means [37] clustering on this feature dictionary, and this phase divides these features within F_j into C reference clusters and records each cluster center of these reference clusters in a vector $Center_j = (center_{j,1}, \ldots, center_{j,c}, \ldots, center_{j,C})$.

In imputation task, the system extracts the feature $fmissing^{(j)}$, $j = \{1, 2, \cdots, M\}$ for the incomplete data $x_{missing}$ with $j-$th, $j = \{1, 2, \cdots, M\}$ DBNs model.

Then the distance $dis(center_{j,c}, fmissing^{(j)})$ between $fmissing^{(j)}$ and every center of $center_{j,c}$ is calculated by a distance metric. Various types of distance metrics can be applied to our system. In this paper, we choose the Euclidean Distance to calculate the distance values.

The feature $fmissing^{(j)}$ is classified to the $c-$th cluster if the distance $dis(center_{j,c}, fmissing^{(j)})$ is minimum.

Assume the $c-$th cluster in F_j is further defined as

$$F_{j,c} = \begin{pmatrix} feature_{j,c,1} \\ \vdots \\ feature_{j,c,s} \end{pmatrix} \tag{13}$$

where s is the size of samples in $c-$th cluster in the feature dictionary F_j. For the $fmissing^{(j)}$ we further apply the Euclidean Distance to measure the distance between $fmissing^{(j)}$ and every feature in $F_{j,c}$, getting a distance vector $D = (dis_{j,c,1}, dis_{j,c,2}, \cdots, dis_{j,c,s})$.

In the general data-driven method, the nearest neighbor in the equation of (13), feature dictionary $F_{j,c}$ is usually chosen as the complete substitution for incomplete $x_{missing}$. However, it might be a disadvantage of the nearest neighbor that does not cover uncertainty of missing variable, hence it is not a good unbiased estimation for $x_{missing}$. To overcome uncertainty of missing variable, we select the smallest k $(k \leq s)$ distances $(dis_1, dis_2, \cdots, dis_k)$ from distance vector D and find the nearest k observed attribute samples $(\rho_1, \rho_2, \cdots, \rho_k)$ in training data set T_j, which

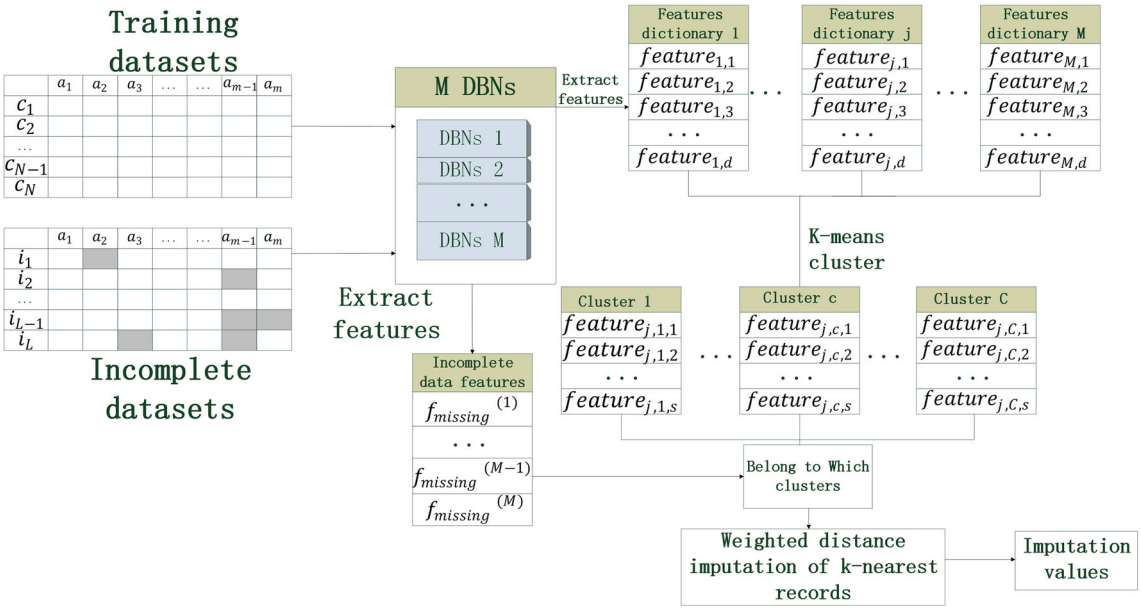


Fig. 5 The architecture of the Data Driven Imputation (DDI) for creating M candidates of the missing data

correspond to the nearest k neighbors in feature dictionary $F_{j,c}$ respectively.

Equation (14) estimates the final imputation candidate O_j for the input incomplete data:

$$O_j = \sum_{i=1}^{k} \psi_i \rho_i \tag{14}$$

where ρ_i is corresponding values of attributes in complete data samples, and in our assumption $\rho_i = (a_3^i, a_5^i)$. The weight ψ_i is calculated by (15).

$$\psi_i = \frac{1}{dis_i} / \sum_{i=1}^{k} \frac{1}{dis_i} \tag{15}$$

The imputation process for creating one imputation candidate value O_j for the input incomplete data is described in Algorithm 3 and illustrated in Fig. 5:

Algorithm 3 Data driven imputation.

Require:

T_j: a training data set

$x_{missing}$: an input incomplete data

Ensure:

O_j: imputation value

1: **for** $j = 1$ to M **do**

2: Using T_j to train the $j-$th DBNs.

3: Calculating feature dictionary F_j of training set T_j with (11).

4: Dividing this feature dictionary into C reference clusters by K-Means clustering, getting cluster center list $Center_j = (center_{j,1}, center_{j,2}, \ldots, center_{j,C})$.

5: **end for**

6: **for** $j = 1$ to M **do**

7: Obtaining the feature $fmissing^{(j)}$ for the input incomplete data $x_{missing}$ by $j-$th DBNs and measuring the distance $dis(center_{j,c}, fmissing^{(j)})$ between this feature and every cluster center.

8: Choosing the cluster c with the nearest $dis(center_{j,c}, fmissing^{(j)})$ as the data-driven set $F_{j,c}$.

9: Measuring distance between $fmissing^{(j)}$ and every feature in $F_{j,c}$, getting a distance vector $D = (dis_{j,c,1}, dis_{j,c,2}, \cdots, dis_{j,c,s})$.

10: Selecting the nearest k samples from instance set E according to distance D and creating the imputation candidate O_j by (16) and (17).

11: **end for**

Repeating Step7 to Step10, we can generate M imputation candidates $O_1, O_2, \cdots, O_M$ for an incomplete data $x_{missing}$. These M estimations can be pooled into one final value by method proposed in [27]. In this paper, we use the method of averaging.

4.4 Hybrid MI system

The first method illustrated above can effectively handle monotone missing data pattern with low computation complexity. In contrast, the second method has the ability to impute arbitrary missing data pattern, but brings high computation complexity, additionally, it is more robust than the first method to the increase of missing ratio based on the results of experiments. Considering respective advantages of these two methods, we construct a hybrid MI (HMI) system by integrating these two methods for resolving different missing patterns with good performance. Figure 6 shows the framework of hybrid MI system. The Feature Regression and DDI share M DBNs models. Before imputation procedure, all DBNs and regression models have been pre-trained with complete datasets, and all feature dictionaries have been obtained.

As illustrated in Fig. 6, (1) the system firstly divides the input incomplete data into different missing patterns, and analyzes the missing ratio in the monotone missing pattern. (2) For the data involved monotone missing pattern, if the missing ratio is lower than a threshold Th, the system will choose the low computation complexity method (i.e. the first method, Feature Regression) to implement the multiple imputation procedure, otherwise it applies the DDI to fill the missing data. The arbitrary missing pattern data set can only be processed by the DDI method.

The threshold Th is a hyperparameter for the system based on our experience, it should be set a value lower than 9%.

As shown in Fig. 6, (3) the MI framework can generate M imputation candidates for an incomplete data. Overall M candidates then will be used to calculate the final value.

4.5 Complexity Analysis

Before imputation calculation, all the DBNs models have been pre-trained; all feature dictionaries have been prepared and the center list of these reference clusters has been calculated by K-Means method. The main computation cost of HMI is in regression and data-driven processing, therefore, the computation cost of HMI is similar to the traditional regression and clustering methods.

In imputation step, only the time consumption of feature extraction for the input data is considered. The time consumption is related to the matrix dimension and the network depth, which can be roughly expressed as $O(\alpha^2 * \beta^2 * C_{in} * C_{out})$. The α represents the side length of each feature map that convolution kernel outputs, which is related

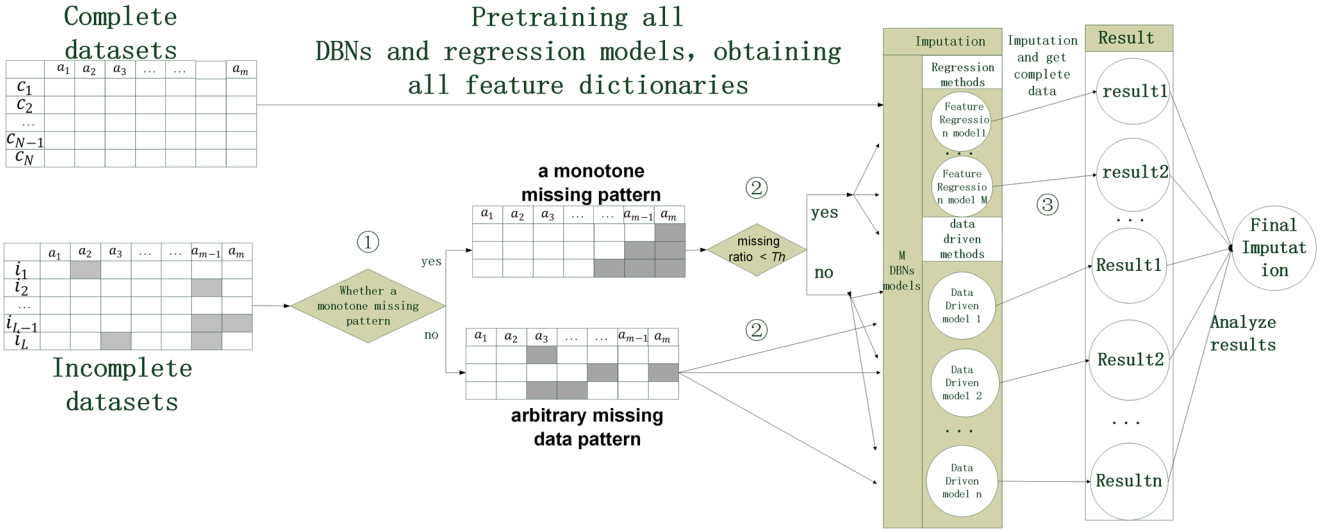


Fig. 6 Hybrid Multiple Imputation (HMI) architecture for dealing with arbitrary missing data pattern by an analysis of missing ratio and missing pattern in input data. All DBNs models shared by two methods are pre-trained with complete datasets before imputation procedure. Imputation procedure consists of 3 steps

to the matrix size Γ ,the convolution kernel size β, padding p, and stride γ, expressed as follows:

$$\alpha = (\Gamma - \beta + 2 * p)/\gamma + 1 \tag{16}$$

β is the side length of each convolution kernel, and C_{in} is the number of channels per convolution kernel, that is, the number of input channels, is also the number of output channels of the previous layer, and C_{out} is the number of convolution kernels that the convolution layer has, that is, the number of output channels. In actual application, the type of data missing will be judged first through a simple conditional judgment, so the time complexity can be approximated as $O(1)$. If it is judged to use the Feature Regression method, the complexity only need to add a logistic regression time cost $O(M \times O(C_{regression}))$, where M is the number of regression machines and $O(C_{regression})$ is the complexity for the regression method. If it is judged to use Data Driven Imputation method, in addition to the time consumption of the feature extraction by M DBNs, the data-driven time complexity can be approximated as $O(M \times (O(C_{locating}) + O(C_{choosing})))$,where $O(C_{locating})$ means the time cost for locating the cluster, M means the number of DBNs and $O(C_{choosing})$ means the computational burden for choosing k nearest reference samples in the $c-th$ cluster. Therefore, the most important time consumption in HMI method is the time cost on the feature extraction of DBNs and candidate samples selected, which is feasible in parallel implementation. The HMI method is able to process efficiently the imputation task within a big incomplete data set.

The spatial complexity of the model is mainly composed of the space occupied by the DBNs model and the space occupied by the reference data samples for calculating imputation values. The spatial complexity of DBNs mainly includes the total parameter quantity and the output feature map of each layer. The total parameter quantity refers to the total weight parameters of all layers, and the space complexity can be approximated as $O(\sum_{l=1}^{D} \beta_l^2 \cdot c_{l-1} \cdot c_l)$, which is only related to the size of the convolution kernel β , the number of channels c , and the number of layers D , regardless of the size of input data.The spatial complexity of the output feature map is relatively simple, that is, the product sum of the space size α^2 and the number of channels c , which is approximately $O(\sum_{l=1}^{D} \alpha^2 \cdot c_l)$. Therefore, the spatial complexity of DBNs can be approximated as $O(\sum_{l=1}^{D} \beta_l^2 \cdot c_{l-1} \cdot c_l + \sum_{l=1}^{D} \alpha^2 \cdot c_l)$. In addition, the space complexity of the HMI method is related to the number of clusters C and the number of reference samples per cluster s , which can be approximated as $O(M \times C \times s \times d)$,where d means the dimension of feature. For a cluster system with thousands of GB memory resource that space complexity can be ignorable.

4.6 Application in a Hadoop cluster monitoring system

As we all know, Hadoop is a fantastic distributed platform, which provides so powerful parallel and distributed computation. It's obvious from above that HMI can be joined into the preprocessing module of any monitoring systems to resolve the partial data missing problem, therefore, for our purpose, we apply the HMI method to construct a new Hadoop cluster monitoring system based on the Ganalia framework [38].

The proposed monitoring system gets statistical data about the cluster resource information for further supporting better job schedule planning and job behavior prediction. As described in Fig. 7, the system architecture has been designed as three levels: 1) cluster information collection level, 2) data preprocessing level and 3) data analysis level. The Ganglia collects data from each host. Different from other Hadoop cluster monitoring system also constructed with Ganglia, this new system adds the HMI method into the preprocessing module before the data integrating processing. The HMI extends the capability of the new system to deal with the missing data. Finally, data analysis module takes the integration data and some statistical analysis methods are applied to get the decision.

5 Performance evaluation

In this section, we describe the experiments conducted to evaluate the effectiveness and efficiency of the new Data Driven Imputation(DDI) and hybrid MI(HMI) system for dealing with partially missing data and large data volume. In addition to our new methods, we also implement the following eight existing techniques for comparison:

1. Regression method: the traditional regression model described in [27].
2. MCMC method: the traditional MCMC method proposed by Schafer [28].
3. Expectation Maximization Imputation (EMI): Expectation Maximization Imputation method is a popular tool for statistical missing data imputation in various fields, the algorithm has reasonable accuracy in missing data imputation [39].
4. KNN method: KNN is the most common method because it shows a stable performance regardless of the size of the missing data with easy implementation. Therefore, many researchers use this method as a benchmark to compare imputation performance [17].
5. Random Forest regression method(RF): The random forest algorithm(RF) used in [7] as a regression model to estimate the missing data.
6. Iterative Fuzzy Clustering(IFC): Presented in [2], the iterative fuzzy clustering approach is applied to obtain the clusters, then the non-missing data in the cluster with their membership degree and the centroids provide information for estimating the missing values.
7. Sample Self-representation Strategy(SSR):This method combines sample self-representation strategy and underlying local structure of data in a uniformed framework for estimating missing data [23].
8. Class Center based Missing Value Imputation(CCMVI): Proposed in [26], it produces imputation results based on the distance between the class center of each class and a threshold calculated by the other observed data.

Besides observing various missing data patterns, missing ratios, different databases in performance comparison between these imputation methods by Mean Absolute Percentage Error(MAPE), we also compare the effectiveness and efficiency of our two MI methods (Feature Regression and DDI)in the following experiments.

5.1 Database

All the experiments are conducted on three data sets: A real Hadoop Cluster Monitoring Dataset, Cover Type Dataset and TV News Channel Commercial Detection Dataset. The Hadoop Cluster Monitoring data are collected by a Ganglia system from a real Hadoop cluster. The cluster has 30 nodes and each node is a X86 server with dual 3.7Ghz i3-6100 processors, 32GBRAM, 500GB disk and Gigabit Ethernet networking. The whole monitoring data includes 103,418

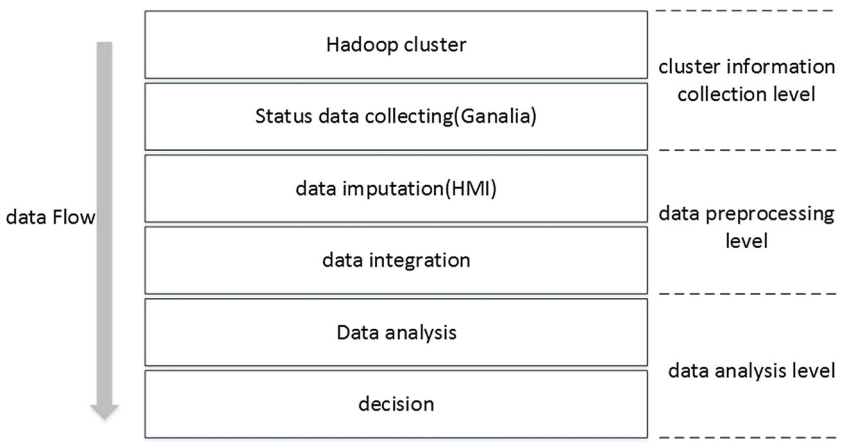
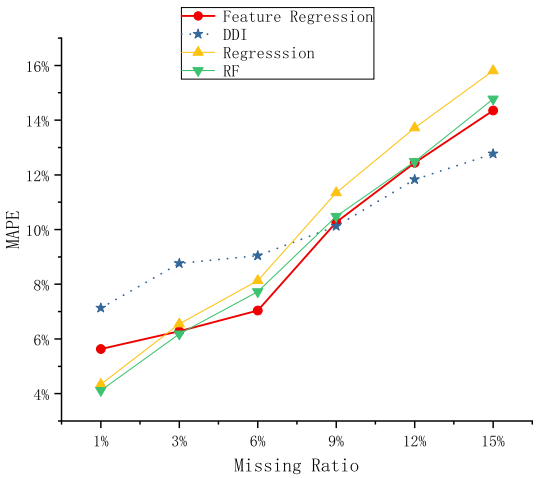
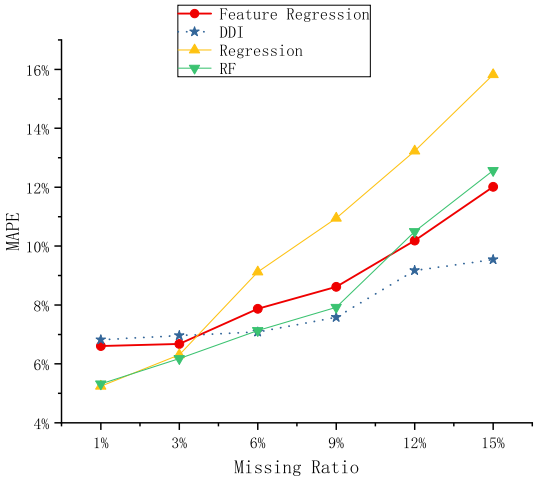


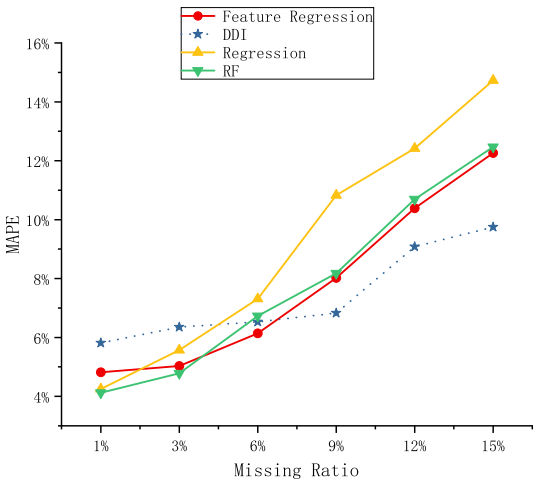
Fig. 7 The structure of the proposed Hadoop monitoring system based on Ganalia and using HMI to impute missing data in preprocessing module



(a) Cover type Dataset

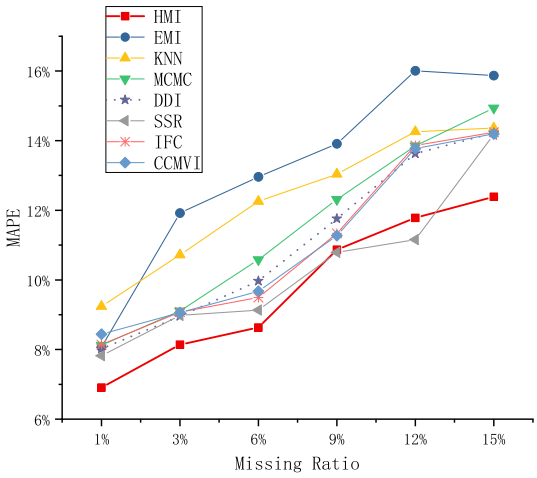


(b) Hadoop Cluster Monitoring Dataset

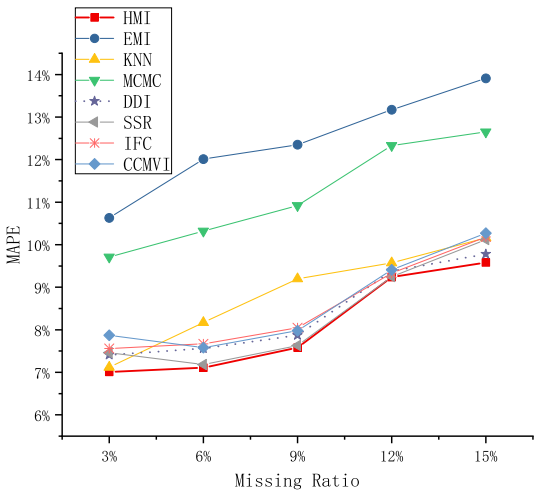


(c) TV News Channel Commercial Detection Dataset

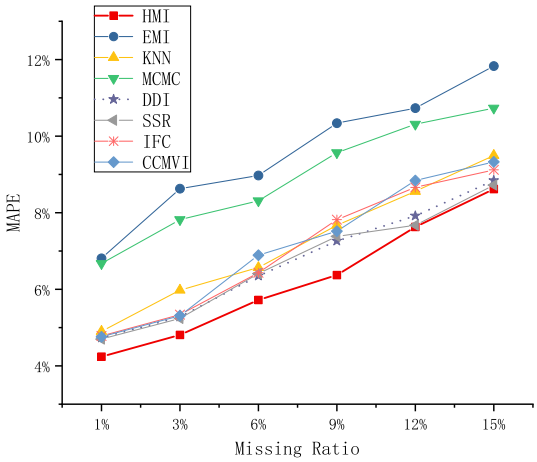
Fig. 8 Performance of MAPE, as a function of missing ratios, obtained by our two methods (Feature Regression, DDI), the Regression method and RF method on three datasets, respectively



(a) Cover type Dataset



(b) Hadoop Cluster Monitoring Dataset



(c) TV News Channel Commercial Detection Dataset

Fig. 9 Performance of MAPE for different missing ratios and datasets, obtained by HMI, EMI, KNN, MCMC, DDI, IFC, SSR, CCMVI methods respectively

records, 116 attributes which include data(percentage CPU, memory used, I/O...) for each node in a Hadoop system. The last two datasets called Cover Type Dataset were the simulated data of the sensor clusters from the UCI Machine Learning Repository, which records the cover type data from four wilderness areas located in the Roosevelt National Forest of northern Colorado. It includes 581,012 records, 54 attributes and 7 classes.There are 129,685 records in the TV News Channel Commercial Detection Dataset, and each record has 98 attributes and 2 classes (Commercials/ Non-Commercials).

We randomly select 600,000 records from Hadoop Cluster Monitoring Dataset, 300,000 records from Cover Type Database and 80,000 records from TV News Channel Commercial Detection Database to construct three analysis data sets.

Although each data attribute in these analysis datasets has a different domain, we preprocess all the values in these three datasets so that make them normalized to [0, 1] for better learning computing. In real imputation task, the system output can be conducted an anti-normalization processing to recover the original domain. In our experiments, we test these algorithms under the same conditions and hence do not execute anti-normalization.

Cover Type Dataset and TV News Channel Commercial Detection Dataset are artificially regenerated to six data subsets such that have 1%, 3%, 6%, 9%, 12%, 15% missing data ratios respectively. The missing mechanism is MAR, missing patterns include monotone missing pattern and arbitrary missing data pattern. For the Hadoop Cluster Monitoring Dataset we artificially regenerate five arbitrary missing test data subsets for missing data ratios=3%, 6%, 9%, 12%, 15%. Meanwhile, we artificially construct six monotone missing testing data subsets with missing data ratios=1%, 3%, 6%, 9%, 12%, 15% on this dataset.

5.2 Quantitative measures for evaluation

To evaluate the performance of the imputation algorithms, well-known evaluation MAPE is used, formulated as follows:

$$MAPE = \frac{\sum |\frac{e_i}{O^i}|}{N} \times 100\% \tag{17}$$

where $e_i = \bar{O}^i - O^i$ is the error for estimate value, $\bar{O}^i$ is the imputed value of the $i-$th missing data, O^i is the actual value of the $i-$th artificially created missing data, and N is the number of artificially created missing data. It's important to note that we use the actual average of $i-$th value to replace the zero of O^i. The values of MAPE can range from 0 to ∞, and a lower value indicates better accuracy.

5.3 Results and discussions

The DBNs, in these experiments, have three hidden layers, and the nodes for each layer are 40, 28, 14 respectively. All the layers are randomly initialized before training. We chose $Sigmoid$ as the activation function. The learning rate is set to 0.01, alpha is set to 1, and hyper parameters are set to 0.

In the following experiments the multiple imputation $M = 15$ for all methods means randomly sample 15 training data sets under total training data. Depending on the classes of each dataset, we select the number of cluster $C = 7$, 10 and 2 for our systems under three different datasets respectively. $k = \lfloor 0.01 \times s \rfloor$ is used for the DDI method, where the s is the size of samples in the chosen cluster.

5.3.1 Experiment on monotone missing pattern data

The first experiment investigates the performance of the two proposed methods Feature Regression and DDI in dealing with missing data imputation in monotone missing pattern. We choose the traditional regression method and RF method as the control group, based on the imputation ability of the monotone missing pattern.

Figure 8 shows the MAPE values, as a function of missing ratios, obtained by our two methods, the Regression method and RF method on the three datasets, respectively.

Figure 8 indicates that all the systems can achieve good performance in case of low missing ratio $<$ 6%. After averaging all three datasets, the accuracy of regression and RF methods show a result which is slightly 1.0 $\sim$ 1.2 higher than our methods, while missing ratio $<$ 2% and RF method gives better performance than all other methods within missing ratio $<$ 3%. The regression strategy is contained in our Feature Regression method for estimating the imputation candidates similar as the traditional regression method. These better MAPE results show that regression method has better prediction ability than data-driven approach when it is applied to get the estimation candidates for the missing data on a low missing ratio condition. As observed in Fig. 8, with the increase of missing ratio, a rise of the MAPE values are found for all methods, but comparing with the traditional regression and RF approach, our methods especially the DDI method show more robustness. Our experiments show the same results as literature [7] and prove that the RF regression method can perform better than traditional regression model.

Our Feature Regression method shows lower MAPE values than the traditional regression system and RF for all test databases, when the missing ratio is greater than 9%, except 0.59% drop compared with RF method for missing ratio at 9% in the Hadoop Cluster Monitoring Dataset. Especially when missing ratio $>$ 12%, our two methods both outperform the traditional regression and

Table 1 MAPE values obtained by the MCMC, EMI, KNN,IFC, SSR, CCMVI, DDI and HMI respectively, averaging overall results on arbitrary missing pattern and three datasets

MisRatio	DDI	MCMC	EMI	KNN	IFC	SSR	CCMVI	HMI
3%	7.33%	8.95%	10.39%	8.03%	7.33%	7.23%	7.41%	6.65%
6%	7.95%	9.72%	11.33%	9.08%	7.86%	7.57%	8.05%	7.16%
9%	9.02%	11.02%	12.20%	9.99%	9.07%	8.58%	8.92%	8.27%
12%	10.31%	12.16%	13.26%	10.88%	10.61%	9.37%	10.67%	9.55%
15%	10.97%	12.89%	13.84%	11.45%	11.18%	11.04%	11.26%	10.21%

RF regression method on all datasets. The deep features can give a better abstract for the data and improve the regression effectiveness when the missing ratio rises. This explains why our two models show the possibility to outperform the traditional regression methods in these experiments as the missing ratio increases. The larger the attribute volume, the more effective using DBNs features in imputation.Comparing the attribute number between each database (Cover type=54, TV News Channel Commercial Detection=98, Hadoop index = 116), the higher MAPE improvement can be produced by our methods in the larger attribute volume database, as showed in Fig. 8b and c.

From Fig. 8, we observe that the DDI method outperforms the Feature Regression method when the missing rate is greater than 9% because of poor ability of the regression algorithm with the loss of reliable information to build the prediction model while getting an accurate statistical estimation. In contrast, the DDI method estimates the missing variable by data-driven method, which reduces the dependence of the system on missing data.

The results give us an experience Th value for our hybrid MI system. The $Th = 9\%$ is set in our hybrid MI system in the second experiment.

5.3.2 Experiment on arbitrary missing pattern data

Furthermore, we validate the performance of our system (Data Driven Imputation (DDI) and hybrid MI(HMI)) in these datasets involved arbitrary missing pattern and different missing ratios. We have compared the MAPE values of our methods, MCMC, KNN, IFC, SSR and CCMVI methods in these three analysis datasets, all with arbitrary missing pattern. We do not use the RF method in this experiments, because it is a kind of regression model which can not work well in the arbitrary missing pattern.

Figure 9 shows the MAPE values, as a function of missing ratios, obtained by DDI, HMI, EMI, KNN, MCMC, IFC, SSR and CCMVI methods on three datasets, respectively. Although, the MAPE rises for the all methods with the increase of the missing ratio, our DDI method shows more robustness than MCMC, KNN and EMI. Broad range missing attributes may reduce the effectiveness of statistical distribution estimation, and this causes the poor performance of MCMC. Unlike the MCMC method, DDI method creates the imputation candidates in a data-driven way on some data dictionaries, thus avoids the offer of the statistical error caused by missing variables in imputation.

Our hybrid MI(HMI) system is a combination of Feature Regression and DDI methods as described in Section 4.4. The results of this experiment show that our new HMI performs evidently better than the EMI, MCMC and KNN methods and slightly better than IFC, SSR and CCMVI methods, but SSR for the missing ratio 12% in Cover Type Dataset. In some cases the SSR method generates similar results as HMI, such as for missing ratio $6\% - 12\%$ in Hadoop Cluster Monitoring Dataset. The HMI, IFC and CCMVI methods construct the fill values for missing data all by similar cluster-based and sample-based strategies. However, different from the IFC and CCMVI methods both calculated on the original data, the HMI method constructs the cluster method above the deep features of data.

As mentioned above, the deep features also contribute to the improved performance for the HMI method. The better performance brought by the feature of data also can be seen in SSR method. From Fig. 9, we can see that SSR method is slight better than other methods in many cases in these experiments. Self-representation framework of data explains that the SSR method obtains more effective estimation for the missing data. In contrast to the SSR model which uses the graph regularized local self-representation framework to represent structure features of data, our method utilizes DBNs to extract the deep features of data. The better results obtained by HMI show that the DBNs has better ability for modeling complex structures and dependencies in the data.

Table 1 shows each MAPE value obtained by the MCMC, EMI, KNN, IFC, SSR, CCMVI, DDI and HMI respectively, averaging all results on all missing patterns in three datasets. As indicated in Table 1, our new method HMI improve the adaptability and robustness for missing data imputation while dealing with missing data which involves large missing ratios and arbitrary missing pattern. The performance of HMI is better than that of the DDI

because HMI chooses the optimal system between Feature Regression and DDI to impute missing data for different missing patterns and missing ratios.

6 Conclusion and future work

Within the multiple imputation frameworks, we investigate a Hadoop cluster monitoring system which is robust to partial missing data, apart from this, two novel missing data imputation methods: Feature Regression and Data Driven Imputation have been presented in this paper firstly.

These methods are different from the conventional algorithms in following two aspects:

1. Missing-data estimation procedures of two methods are both dependent on the deep features rather than original data. The benefit of working on deep features is that these methods enhance the ability to represent the high-level structure and dependence of the original data, so that this way improves the robustness to a larger data missing ratio.
2. The combination of the model-based and data-driven imputation strategies reduces the dependence on the accuracy in statistical estimation. As a result, it consequently improves the performance to a larger missing ratio.

Furthermore, we construct a hybrid MI system (HMI) with the proposed methods which inherits the advantages of both methods regardless of low or high missing ratios, regardless of monotone missing patterns or arbitrary missing patterns.

Experimental results show that proposed methods outperform the other methods with the most testing datasets and missing ratios except the 1% drop compared with the regression method in the monotone missing pattern and missing ratio lower than 2%. By taking consideration of the above point, compared with traditional multiple imputation - Regression and MCMC, our methods improve the robustness for larger missing ratios. Meanwhile, comparing the HMI with other two commonly used imputation methods - Expection Maximization Imputation and KNN, the proposed HMI method outperforms other two systems in all testing datasets and missing ratios. It gives the proof that the HMI is a selectable technology to deal with partial missing data within a cluster monitoring application,at the same time, briefly speaking, since DBNs is offline training, the running time of HMI in application mainly includes DBNs feature extraction time and linear regression or classification selecting candidate time. The judgment time about the missing rate and the missing type is almost negligible, so it is more feasible in practical applications.

To improve this algorithm even further, it will be beneficial to study the performance of imputation methods with respect to difference missing mechanisms, i.e. Missing Completely at Random. Moreover, our future work should study a strategy to determine the number of cluster C and the threshold Th, and compare various deep neural network technologies to determine which is the best deep neural network for application in cluster monitoring system.In addition, the DBNs training process can be accelerated with larger data sets, larger memory, more advanced GPU devices and CUDA programming, and higher-frequency CPU devices. In the future practical application, PCA dimensionality reduction or Laplacian feature mapping can be used to reduce the complexity of regression calculation after feature extraction, so as to achieve better real-time effect.

Acknowledgements This work is supported by Sichuan Science and Technology Support Program, No.: 2018GZ0103.

References

1. Massie ML, Chun BN, Culler DE (2004) The ganglia distributed monitoring system: design, implementation, and experience. Parallel Comput 30(7):817–840
2. Nikfalazar S, Yeh C, Bedingfield S, Khorshidi HA (2017) A new iterative fuzzy clustering algorithm for multiple imputation of missing data, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Naples, pp 1–6
3. Jea K, Hsu C, Tang L (2018) A missing data imputation method with distance function. International Conference on Machine Learning and Cybernetics (ICMLC), pp 450–455
4. Mazzutti T, Roisenberg M, de Freitas Filho PJ (2018) Adaptive missing data imputation with incremental Neuro-Fuzzy gaussian mixture network (INFGMN), International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, pp 1–8
5. Berglund P, Heeringa S (2014) Multiple imputation of missing data using SAS. Int Stat Rev 83(2):326–327
6. Little RJA, Donald BR (2002) Statistical Analysis with Missing Data, 2edn
7. Tang F, Ishwaran H (2017) Random forest missing data algorithms. Stat Anal Data Min: The ASA Data Sci Journal, pp 363–377
8. Zhang Y, Liu Y (2009) Missing traffic flow data prediction using least squares support vector machines in urban arterial streets. In: IEEE Symposium on Computational Intelligence and Data Mining, pp 76–83
9. Suh MK, Woodbridge J, Lan M, Bui A, Evangelista LS, Sarrafzadeh M (2011) Missing data imputation for remote CHF patient monitoring systems. International Conference of the IEEE Engineering in Medicine Biology Society, pp 3184–3187

10. Allison PD (2001) Missing data, vol 136. Sage Publications, Newbury Park
11. Srebotnjak T, Carr G, Sherbinin AD, Rickwood C (2012) A global Water Quality Index and hot-deck imputation of missing data. Ecol Indic 17:108–119
12. Turrado CC, Lasheras FS, Calvo-Rolle JL, Piñon-Pazos A. J., Juez FJC (2015) A new missing data imputation algorithm applied to electrical data loggers. Sensors 15(12):31069–31082
13. Sessa J, Syed D (2016) Techniques to deal with missing data, 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, pp 1–4
14. Krause RW, Huisman M, Steglich C, Sniiders TA (2018) Missing network data a comparison of different imputation methods, IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), Barcelona, pp 159–163
15. Duan Y, Lv Y, Kang W, Zhao Y (2014) A deep learning based approach for traffic data imputation. In: 17th International IEEE Conference on Intelligent Transportation Systems, pp 912–917
16. Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2016) Recurrent Neural Networks for Multivariate Time Series with Missing Values. arXiv:1606.01865
17. Thirukumaran S, Sumathi A (2016) Improving accuracy rate of imputation of missing data using classifier methods, 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, pp 1–7
18. Razavi-Far R, Saif M (2016) Imputation of missing data using fuzzy neighborhood density-based clustering. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, pp 1834–1841
19. Azim S, Aggarwal S (2016) Using fuzzy c means and multi layer perceptron for data imputation: Simple v/s complex dataset. 3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, pp 197–202
20. Soni S, Sharma I (2017) An imputation-based method for fuzzy clustering of incomplete data," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, pp 616–621
21. Myneni MB, Srividya Y, Dandamudi A (2017) Correlated Cluster-Based imputation for treatment of missing values, inproceedings of the first international conference on computational intelligence and informatics. Adv Intell Syst Comput 507:171–178
22. Susanti SP, Azizah FN (2017) Imputation of missing value using dynamic Bayesian network for multivariate time series data,International Conference on Data and Software Engineering (ICoDSE), Palembang, pp 1–5
23. Chen X (2018) An Improved Self-Representation Approach for Missing Value Imputation. 24th International Conference on Pattern Recognition (ICPR), Beijing, pp 1450–1455
24. Xu X, Chong W, Li S, Arabo A, Xiao J (2018) MIAEC: Missing Data Imputation Based on the Evidence Chain. IEEE Access 6:12983–12992
25. Zhao L, Chen Z, Yang Z, Hu Y, Obaidat MS (2018) Local Similarity Imputation Based on Fast Clustering for Incomplete Data in Cyber-Physical Systems. IEEE Syst J 12(2):1610–1620
26. Tsai CF, Li ML, Lin WC (2018) A class center based approach for missing value imputation. Knowl-Based Syst 151:124–135
27. Yuan YC (2010) Multiple imputation for missing data: Concepts and new development (Version 9.0), SAS Institute Inc, Rockville, MD, pp 49
28. Schafer JL (1997) Analysis of incomplete multivariate data. CRC Press, Boca Raton
29. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
30. Yoshua B, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
31. Hinton GE (2009) Deep belief networks. Scholarpedia 4(6):5947
32. Hinton GE, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
33. Krizhevsky A, Hinton GE (2011) Using very deep autoencoders for content-based image retrieval. ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges (Belgium), pp 27–29
34. Hajinoroozi M, Jung T, Lin C, Huang Y (2015) Feature extraction with deep belief networks for driver's cognitive states prediction from EEG data. IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp 812–815
35. Chen Z, Liu S, Jiang K, Xu H, Cheng X (2015) A data imputation method based on deep belief network. IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing, Liverpool, pp 1238–1243
36. Green SB, Salkind NJ, Akey TM, Hall P (2012) Using SPSS for Windows: analyzing and understanding data. Am Stat 59(1):113–113
37. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recogn Lett 31(8):651–666
38. Sacerdoti FD, Katz MJ, Massie ML (2003) Wide area cluster monitoring with Ganglia. In: IEEE International Conference on CLUSTER Computin, pp 289–298
39. Gold MS, Bentler PM (2000) Treatments of missing data: A Monte Carlo comparison of RBHDI, iterative stochastic regression imputation, and expectation-maximization. Struct Equ Model 7(3):319–355

Jie Lin He received the B.S. and M.S. degrees in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China, in 2006, and the Ph.D. degree in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China, in 2009. From 2010 to 2012, he was a lecturer with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. Since 2013, he has been an associate professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include image processing, artificial intelligence, big data analysis and data mining.

NianHua Li He received the B.S. degrees in computer science and technology from Chongqing University of Posts and Telecommunications, ChongQing, China, in 2015, and the M.S. degree in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China, in 2018. His research interests include image processing, big data analysis, deep learning and data mining.

Md Ashraful Alam He received the B.Sc. in electrical and electronic engineering from International Islamic University Chittagong (IIUC), Chittagong, Bangladesh, in 2014, and the M.E. degree in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China, in 2018. His research interests include deep learning and data mining.

Yuqing Ma She received the B.S. degrees in electronic information engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2019. Her research interests include data mining, information systems and data science.