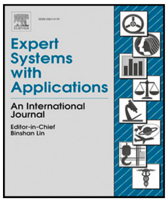
Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Expert Systems with Applications

An International Journal

RESI: A Region-Splitting Imputation method for different types of missing data

Dunlu Peng [*], Mengping Zou, Cong Liu, Jing Lu

Shanghai Key Lab of Modern Optical System, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, 200093, Shanghai, China

ARTICLE INFO

Keywords:
Data mining
Missing data imputation
Region-splitting
k-fold cross validation

ABSTRACT

A certain degree of data loss seriously affects the accuracy and availability of data, especially on the effects of the subsequent in-depth data analysis and mining. It is of great value in practical applications to construct a data imputation model, which is suitable for completing different types of missing data, including numerical only, categorical only and mixed-type data, and has strong capability of generalization. To address this issue, this paper defines a new metric, mean integrity rate, to measure the missing degree of a dataset, and proposes ***RESI***, a novel tuple-based ***RE***gion-***S***plitting ***I***mputation model, to impute different type missing data. We first select features and assign weights to each attribute by using the entropy weight method, and then partition the tuples into a subset of complete tuples and several subsets of incomplete tuples based on their integrity rate, which is formulated with the weights of attributes and the missing degree of tuples. The model performs training iterations on the complete tuple subset. In each iteration, the trained model is used to impute the next missing subset, and the computed subset is merged into the complete subset for training the next model. To improve the imputation accuracy, we leverage *k*-fold cross validation to correct errors. Besides imputing diverse types of missing data, extensive experimental results have shown that our model, RESI, significantly outperforms the state-of-the-art methods in the sensitivity to *missing rate* and accuracy of imputed data.

1. Introduction

Both in the field of research and engineering, missing data is a common and serious problem that cannot be ignored. The missing of data not only seriously affects the accuracy and availability of data, but also affects the subsequent work of in-depth analysis and data mining (Che, Purushotham, Cho, Sontag, & Liu, 2018). If incomplete tuples are deleted directly, treated as special values or not processed, some important information hidden in them will be lost. In comparison, imputing the missing values (Purwar & Singh, 2015) is undoubtedly a better choice. However, with the shift of data collection from manual to automatic, the increasingly rapid and high-dimensional expansion of data makes the transformation of data missing from the single numerical or categorical to the mixed-type of both, which brings severe challenges for researchers.

To date, many methods for missing data imputation have been proposed in the literature (Dohoo, 2015). The simplest way to fill in the data is to use statistical values, such as mean (Chen, Wang, & Chen, 2012), median, etc. or the data with the highest frequency for interpolation. Although this method is simple and easy to use, it is hard to get a satisfying effect. Among the imputing methods based on statistics, regression method (Yang, Li, & Wang, 2006) is more commonly applied, which includes linear regression, logistic regression and multiple regression. For numerical variables, generally, linear regression is often leveraged to fit and estimate their values (Zhao & Tang, 2016). For categorical variables, logistic regression can be exploited to estimate their values (Sentas & Angelis, 2006). In addition, we can make use of support vector regression method to predict the missing numerical variables and support vector machine to predict the missing of categorical variables. Furthermore, Expectation Maximization Imputation (EMI) (Rahman & Islam, 2016) and Bayesian-based imputation approach are also common methods for numerical variables; Schafer et al. propose the saturated multinomial model for categorical variables (Schafer, 2010).

Although the above-mentioned methods have achieved good results under some specific conditions, they are limited to a single type of missing data (Tseng, Wang, & Lee, 2003). Currently, there is little literature on mixed-type data imputation. The consideration of mixed-type of variables first appeared in the Multiple Imputation (Taylor, Rubin, & Rubin, 2012) for missing data proposed by Rubin (1978). Till 1999, Van Buuren and Oudshoorn propose a more elaborate model,

[*] Corresponding author.
E-mail addresses: pengdl@usst.edu.cn (D. Peng), zoump0814@163.com (M. Zou), liucong198408@sina.com (C. Liu), jing.lu@usst.edu.cn (J. Lu).

https://doi.org/10.1016/j.eswa.2020.114425
Received 6 June 2019; Received in revised form 29 November 2020; Accepted 29 November 2020
Available online 5 December 2020

Multivariate Imputation by Chained Equations (MICE) (van Buuren & Oudshoorn, 1999), which utilizes chain equation to combine different regressions for mixed-type data. Constructing a decision tree model can also predict the missing data of mixed types (Rahman & Islam, 2013), where a sample tree model is trained with samples without missing attributes and is used to calculate the missing data. Many ensemble imputation algorithms leverage subspace-based algorithms, which divide the dataset based on different processing models or partitioning the attributes, to improve the imputing performance (Gao, Jian, Peng, & Liu, 2017). Considering that the random forest in ensemble learning process mixed-type data well, Daniel et al. (2012) introduce it into the imputation of missing data and constructs a model named missForest (Stekhoven & Bühlmann, 2012), which can produce good results. These existing filling methods have better effectiveness in datasets with small scale and low missing degree, but with the increase of the missing degree, the assurance of accuracy needs more given information. Therefore, their training of filling model is on the basis of missing data, which is convenient but easy to get uncertain estimation. Uncertain estimation often brings great deviation, especially for large-scale data with high missing degree, which has great negative effects on subsequent prediction and mining. The purpose of imputation is to ensure the accuracy of data statistical indicators, so as to obtain an effective classification and prediction model. So, when the accuracy of input cannot be guaranteed, we should pay more attention to how to use a small amount of given information to train a more powerful model, rather than directly estimate the missing value.

In order to deal with the missing values of different types by a small amount of complete data, we propose a region-splitting (Ohlander, Price, & Reddy, 1978) model of imputation which can reduce the deviation caused by uncertain data and fill the missing value by taking advantages of known information. Considering the uncertainty of the missing value in the real data will bring deviation to the modeling and known information is relatively little, to minimize the deviation caused by missing data, we divide the dataset into different subsets according to the missing degree of each record. These subsets include a subset of complete tuples and several subsets with different degrees of incomplete tuples. The missing degree of a subset can be evaluated according to the attribute weight and the missing situation, and simply it can be divided from low to high. Initially, the imputing model is trained on the complete subset and used for the prediction and imputation of the missing subset with the lowest missing degree. Previous experiments have shown that some existing filling methods have a good effect for data with low *missing rate*, which assures the accuracy of the first imputation of our model. To increase the amount of known information, the subset imputed is incorporated into the complete subset for the training of subsequent models, which effectively enhances the generalization ability of the model to some extent. Similarly, in the process of each iteration, imputed tuples with low missing degree is exploited to process the subset with higher missing degree, till all the incomplete subsets are imputed. To further enhance the imputing accuracy, the model applies *k*-fold validation (Dougherty, Carlson, Blackburn, & Getz, 2017) to alleviate the inaccurate imputations after the iteration.

To investigate the effectiveness of our proposed framework, we conducted extensive experiments on multiple datasets to compare RESI with several state-of-the-art methods for computing the missing data, including Complete Case Analysis (CCA), Deleted Case Analysis (DCA), Analysis with Missing Case (AMC), Replacement of Observed Frequencies (ROF), K-Nearest Neighbor Imputation (KNNI) (Zhang, Cao, Wang, & Li, 2019), Expectation Maximization Imputation (EMI), Multivariate Imputation by Chained Equation (MICE), Decision Tree Imputation (DTI) and Missing Data Imputation with Random Forest(missForest). The detailed description of the methods is presented in the section of experiment. Experimental results have show that our model, RESI, is superior to the comparative methods in terms of generalization, missing-degree sensitivity and imputation efficiency.

We summarize the main contributions of this work as follows:

1. A region-splitting iterative framework, named RESI, is proposed to complete the values for different types of missing data with known information. In order to obtain more accurate imputing results, in the process of iterations, the model employs the same base to predict the missing values for each partition, and then the *k*-fold validation is leveraged to revise the predicated values (Section 4).
2. To measure the missing degree of a certain tuple, we define the *tuple integrity rate* of tuples based on their attribute weights, and in terms of what a partitioning algorithm is constructed to partition all tuples, which realizes different missing cases having different processing and also greatly increases the scale of training model (Section 4.2).
3. Different from *missing rate* and *complete ratio*, a new metric, *mean integrity rate* (Definition 6), is defined to measure the missing degree of a dataset. As is verified by the experiments, the existing imputation methods are more sensitive to this metric compared to other metrics.
4. We conducted extensive comparative experiments on real datasets to testify the effectiveness of our model. The experimental results confirm that our model outperforms the competitive models in the effects of imputation.

The rest of this paper is organized as follows. We give an outline of the current state of missing data imputation study in Section 2. Section 3 describes the preliminaries as well as the problem formulation. In Section 4, we present our model in details. Then, the experiments and their analysis are given in Section 5. Finally, we conclude our work in Section 6.

2. Related work

In recent years, researchers have developed multiple methods of missing data imputation. According to the number of predictive values generated for a missing value, they can be easily divided into the following two categories: (a) *single imputation schemes*, which output one predictive value for each missing value, such as ROF, EMI and KNNI; (b) *multiple imputation schemes*: such as MICE and missForest, they either repeatedly use one algorithm to generate multiple predictive values, or they use multiple imputation techniques to generate multiple predictive values respectively or jointly. Then, some rule, such as the Rubin rule (Rubin, 2009), is used to combine these predicted values to get a final effective prediction.

For further discussion, depending on whether the number of parameters in the model increases with the number of examples in the dataset, the single imputation schemes are divided into *parametric method* (Ma & Chen, 2018) and *non-parametric method* (Derek, Lyndsay, & John, 2019). *Parametric method* means that the parameter is fixed and does not change with data quantity, such as regression-based imputation and EMI. When we have enough prior knowledge of a certain dataset, *parametric method* can achieve a good imputing effect, but if the dataset is not well understood or the improper model is used for processing the data, the imputation results will be greatly deviated, which will seriously affect the quality of subsequent in-depth analysis. Actually, in most practical applications, it is usually difficult to get prior knowledge of the dataset being processed. In these cases, a *non-parametric method*, such as KNNI or classification tree based imputation (Kezban, Irina, Karen, & Ryan, 2018), which does not mean that it has no parameters but the number of which varies with the amount of data, can provide more effective imputing results (Pan, 2011).

Some existing models, such as KNNI, decision tree imputation, MICE, missForest and XGBoost (Zhang, 2018), have been proved having good capability of dealing with mixed type missing data by experiments. KNNI, as the base imputer for the model in this article, will be detailed described in Section 3.1. For the brevity, we only introduce the other four approaches that are competitive to ours.

Table 1
Comparison of state-of-the-art imputation methods.

Methods	Type of variables	Type of method	Init nan?	Superiority	Weakness
EMI	Numerical	Parametric	√	Strong adaptability and suitable for large samples	Slow convergence speed, complex calculation, and only for numerical missing data
DTI	Mixed-type	Semi-parametric	√	Can predict the direction of data very well and the imputation result has high credibility	Overfitting will result in low classification accuracy and long decision time
MICE	Mixed-type	Semi-parametric	√	Having good convergence effect and flexible applicability	easy to ignore the nonlinear term and cause bias
missForest	Mixed-type	Non-parametric	√	High accuracy in imputing different data types	Large operation cost and high efficiency
XGBoost	Mixed-type	Non-parametric	√	High accuracy in imputing different data types and effectively solve the problem of overfitting	Large operation cost and high efficiency
KNNI	Mixed-type	Non-parametric		Having simple working mechanism and no explicit training process	Relatively weak effect on subtype variables

Decision Tree Imputation (DTI) Decision Tree Imputation is to construct a decision tree model with the given data in a dataset. The decision tree model is constructed by top-down recursion (Dong & Liu, 2013). After obtaining the observed samples by processing the original data, readable rules and decision trees are generated from the training set and its related class labels. With the establishment of the decision tree, the dataset is recursively divided into several smaller subsets, which are used to predict and impute the missing values.

Most of the existing DTI technologies are combined with EMI, where the decision tree classifies the original dataset and *EM Imputation* algorithm is carried out on each class (Kezban et al., 2018). Although the combination method is an improvement of *EMI* and effectively improves the imputation accuracy, it is only applicable to the numerical missing data. The missing value imputation technology based on decision tree proposed by Rahman, *DMI* (Rahman & Islam, 2010), combines with *EM Imputation* to process the mixed-type missing value. In the process of dealing with mixed-type missing values, DMI first fills in the numerical missing data with *EMI*, and constructs a decision tree model with the completed dataset to impute categorical data. In view of EMI's slow convergence speed, complexity of calculation and easy to fall into local extremities, *Regression Tree* (Madan & Basav, 2018) is employed to deal with the numerical values, and *Classification Tree* for categorical data. DTI takes all the data information to better predict the trend of the data, and its imputing results have a high credibility; however, as there is a certain gap between training data and real data, the classification model trained by the training data may not applicable to the real data because overfitting may occur that leads to low classification accuracy, long decision time and reducing imputation effect.

MICE Using chain equations for multiple regression, MICE is an attempt to combine the advantages of both regression and multiple imputation. When MICE processes missing data, the missing values are first randomly initialized, then the missing variables are estimated according to the chain equation; in this way, multiple problems are decomposed into a series of univariate problems. The *mice* package in R software can sum up the parameter model of the data generation process, and iteratively draw complete condition specification for the missing values by predicting the posterior distribution, so as to input the missing data for many times. Compared with the general multiple imputation methods, MICE has better convergence effect and flexible applicability. However, in MICE implementation, the default setting is to include numerical variables as linear terms in the model, which may ignore the important nonlinear terms, which may lead to biased results.

missForest Considering that the random forest in ensemble learning can handle mixed data well, Daniel introduces it into the study of missing data imputation and constructs missForest. MissForest uses the random forest as a regression method to estimate missing values. The core idea is to take the known data as the feature, the missing values as the label, the data without missing in the label as the training set, and the missing part as the test set, so as to predict and update the missing data through the random forest model obtained from the training. As a representative of *non-parametric method*, missForest does not need to assume the distribution of data and has a high imputation accuracy for different data types. But practice also shows that compared with other methods, missForest has a big disadvantage in operational efficiency, where the running time will increase with the increase of the number of trees, while the decrease of the number of trees will lead to a higher estimation error.

XGBoost As a high realization of Gradient algorithm in ensemble learning, XGBoost is an optimization of the traditional Gradient Decision Tree. XGBoost handles mixed types of data as well as missForest, and has built-in routines that handle missing values. Unlike missForest, however, XGBoost introduces regularizer, in combination with the column sampling and shrinkage technology, can effectively address the problem of overfitting. Corresponding, it has higher computation cost that cannot ignore.

Table 1 compares the characteristics of each imputation method. It is easy to find that, except KNNI, the other methods all simply estimate the missing values at the beginning of the model, on which the subsequent training of the model is based. This processing is very convenient but easy to get uncertain estimation, which will bring great deviation to the prediction.

3. Preliminaries

In our proposed region-splitting imputation model, imputation and partition are the most critical links. Therefore, the selection of base imputing technology and weight calculation in partitioning algorithm are both crucial. The former needs to consider both the imputing effect and the computational cost brought by the iterative framework, while the latter needs to comprehensively evaluate the impact of each attribute on tuples. In this work, the K-Nearest Neighbor Imputation (KNNI) method and the Entropy Weight Method are selected as the base imputer and the way to calculate weight respectively, which are briefly introduced in the following description.

3.1. KNNI

KNN Imputation (KNNI) is a commonly used non-parametric imputation method, whose idea is as follow: Given a test sample of missing value, select the k nearest samples in the observed set (complete samples) based on some distance measurement, generally Euclidean distance, and then the prediction is made according to the information of the k *neighbors*. Typically, for categorical data, it uses the *voting method* (Sun, Zhang, Zhou, & Liu, 2010) to choose the category tag with the most occurrence in the attribute as the filling result in the k nearest samples. For numerical data, the *average method* can be adopted, that is, the average value or weighted average value of the attribute of the k nearest samples can be taken as the filling result. However, this parameter is not known beforehand. In order to facilitate the subsequent comparison with missForest, the cross-validation KNN imputation algorithm (Stekhoven & Bühlmann, 2012) in missForest is adopted to obtain a suitable k.

As for the region-splitting iterative training model proposed in this paper, if other prediction models are adopted, it will cost a large amount of time to train the model in each iteration. Different from other training methods, KNN, as a famous representative of *lazy learning* (Wettschereck, Aha, & Mohri, 1997), has almost no explicit training process. In the training stage, it only saves the observed samples and processes them after receiving the test samples. The training time cost of this learning technology is almost zero, which effectively solves the computational inefficiency that may occur in the learning method.

3.2. Entropy weight method (EWM)

The Entropy Weight Method (EWM) (Cui, Feng, Jin, & Liu, 2018), which utilizes information entropy (Liang & Shi, 2004) to determine the weight of each attribute, is a basis of multi-attribute comprehensive evaluation. Its main idea is to decide the objective weight according to the magnitude of attribute variability. Generally, low information entropy indicates that the attribute has a large variability and it can provide more information for the following data analysis, which means it has greater impact on the data mining results and its weight naturally increases. On the contrary, greater information entropy implies that the impact of the attribute puts on the following data analysis is less, so the weight given to the attribute is small. The calculation steps of EWM can be simply divided into the following three steps:

Step 1: Normalizing data.

Given s attributes $A_1, A_2, \dots, A_i, \dots, A_s$, where $A_i = \{a_{i1}, a_{i2}, \dots, a_{ij}, \dots, a_{in}\}$, we denote the normalized value of attribute A_i as $Y_i = \{y_{i1}, y_{i2}, \dots, y_{in}\}$ and each $y_{ij} \in Y_i$ is formulized as:

$$y_{ij} = \frac{a_{ij} - min(A_i)}{max(A_i) - min(A_i)} \tag{1}$$

Step 2: Calculating the entropy of each attribute.

Suppose Y_i is the normalized set of the attribute A_i, the possibility of each value $a_{ij} \in A_i$ can be computed as $p_{ij} = \frac{Y_{ij}}{\sum_{i=1}^{n} Y_{ij}}$. Specially, when $p_{ij} = 0$, $\lim_{p_{ij} \to 0} p_{ij} lnp_{ij} = 0$. Thus, the entropy of the attribute A_i can be calculated as:

$$E_i = -ln(n)^{-1} \sum_{i=1}^{n} p_{ij} lnp_{ij} \tag{2}$$

Step 3: Determining the weight of each attribute.

After computing the entropy of each attribute with Formula (2), next is to compute the weight of each attribute with :

$$w_i = \frac{1 - E_i}{k - \sum E_i} (i = 1, 2, \dots, k) \tag{3}$$

3.3. Problem definition

Given a dataset contains lots of tuples, each of which is composed of several attributes. The attribute is in either numerical or categorical type. Suppose there are several various types of values missed in part of tuples, our goal is to establish a model which can effectively complete the tuples with numerical only, categorical only or mixed-type of both. For ease of reading, we list the main notations used in this paper in Table 2. Concisely, we formally define the problem as follows:

Let $\mathcal{D}$ be a dataset (Johannesson, 2002), and $\mathrm{A} = \{A_1, A_2, \dots, A_s\}$ be the set of its attributes and $A_i \in A$ be a single attribute. We denote it as $\mathcal{D} = \{t_1, t_2, \dots, t_n\}$, where $t_i \{a_1, a_2, \dots, a_s\} \in \mathcal{D}$ is a tuple of $\mathcal{D}$ and $a_i \in t_i$ is the value of t on the attribute A_i and is also called an *item*. In particular, if $a_i = null$, we define it as an *missing item* (*mi*) and denote it as a_i^{mi}. Based on the completeness of their attributes, we can classify the tuples $t_1, t_2, \dots, t_n$ into two types : *complete tuples* and *incomplete tuples*.

Definition 1 (*Complete Tuple and Incomplete Tuple*)**.** Given a tuple $t\{a_1, \dots, a_i\} \in \mathcal{D}$, if $\nexists a^{mi} \in \{a_1, a_2, \dots, a_s\}$, we call t as a **complete tuple** (CT), otherwise, as an **incomplete tuple** (ICT).

The definition of CT and ICT implies CTs have no missing items and ICTs have at least one missing item. Next, we employ the metric, *missing rate* and *complete ratio* (Li, Shao, & Li, 2014), to quantitatively measure the incompleteness of the dataset from the perspective of missing items and incomplete tuples.

Definition 2 (*Missing Rate*)**.** Suppose a dataset $\mathcal{D}$ contains s attributes and n tuples, which means there are totally $n * s$ items in the dataset. The **missing rate** of $\mathcal{D}$, denoted as $MRate_{\mathcal{D}}$, represents the percentage of its missing items in its total items and is computed with Eq. (4).

$$MRate_{\mathcal{D}} = \frac{N_{miss}}{n * s} \tag{4}$$

where N_{miss} is the number of missing items in $\mathcal{D}$.

Definition 3 (*Complete Ratio*)**.** Given a dataset $\mathcal{D}$, its **complete ratio**, represented as $CRatio_{\mathcal{D}}$, refers to the proportion of complete tuples in $\mathcal{D}$ and is formulized in Eq. (5):

$$CRatio_{\mathcal{D}} = \frac{CT_{\mathcal{D}}}{CT_{\mathcal{D}} + ICT_{\mathcal{D}}} \tag{5}$$

in which, $CT_{\mathcal{D}}$ and $ICT_{\mathcal{D}}$ are the set of complete and incomplete tuple of $\mathcal{D}$, respectively.

Definition 4 (*Missing Type*)**.** Let $\mathcal{N}$ be the numerical variables and $\mathcal{C}$ be categorical variables. $M_{\mathcal{D}} = \{ a_1^{mi}, a_2^{mi}, \dots, a_s^{mi}\}$ is the set of missing items in $\mathcal{D}$, then $M_{\mathcal{D}}$ can be divided into two subsets, that is, $M_{\mathcal{D}} = M_{num} \bigcup M_{cat}$ where M_{num} and M_{cat} stand for numerical and categorical missing values respectively, and obviously $M_{num} \cap M_{cat} = \phi$. According to whether there exists elements in the set of M_{num} and M_{cat}, missing type of $\mathcal{D}$ can be classified as:

1. **Categorical-missing dataset** means $\mathcal{D}$ with only categorical items missing, that is, $\mathrm{M_{num}} = \phi$, $\mathrm{M_{cat}} \neq \phi \rightleftharpoons \mathrm{M} \subset \mathcal{C}$;
2. **Numerical-missing dataset** implies $\mathcal{D}$ with only numerical items missing, formally, $\mathrm{M_{num}} \neq \phi$, $\mathrm{M_{cat}} = \phi \rightleftharpoons \mathrm{M} \subset \mathcal{N}$;
3. **Mixed-type missing dataset** represents $\mathcal{D}$ with both numerical and categorical item missing, namely, $\mathrm{M_{num}}$, $\mathrm{M_{cate}} \neq \phi \rightleftharpoons \mathrm{M} \subset \mathcal{N} \bigcup \mathcal{C}$.

In reality, it is more common for the above three types of missing values to exist simultaneously. A good imputation method should be able to effectively complete all of them. If an imputing method is effective only for one or two of them, but not all of them, this method is not an ideal method. Therefore, the goal of this article is to design and implement a solution that can effectively impute these three different types of missing values at the same time. It is not hard to see that the

Table 2
Notations and their explanations.

Name	Meaning	Name	Meaning
$\mathcal{D}$	Dataset	$\mathrm{A}\left\{A_1, A_2, \ldots, A_s\right\}$	Attributes
s	The number of attributes	m	The number of incomplete subsets
$\mathcal{T}\left\{T_1, T_2, \ldots, T_m\right\}$	Partition set of incomplete tuples	CT	Complete Tuple
V_{value}	Attributes with values	ICT	Incomplete Tuple
V_{miss}	Attributes missing values	N_{value}	The number of the items having value
$CRatio_D$	*complete ratio*	N_{miss}	The number of the missing items
$MRate_D$	*missing rate*	$\mathcal{N}$	Numerical variables
r^t	*tuple integrity rate*	mir^D	*mean integrity rate*
$f(\cdot)$	function	$\mathcal{C}$	Categorical variables
CCA	Complete Case Analysis	AMC	Analysis with Missing Case
DCA	Deleted Case Analysis	ROF	Replacement with Observed Frequencies
$KNNI$	K-nearest neighbor imputation	EMI	Expectation Maximization Imputation
$MICE$	Multivariate Imputation by Chained Equations	CT_0	Initial complete subset with complete tuples
DTI	Decision Tree Imputation	F_1	F1-scores

mixed-type missing dataset is the most complex of the three. Therefore, our work focuses on developing a data imputation approach that addresses the presence of mixed-type missing values, while ensuring that it performs well even if only categorical or numerical missing values are completed.

4. Our model

4.1. Framework of RESI

Fig. 1 depicts the framework of RESI, which firstly divides the input dataset into two parts: *complete tuples* and *incomplete tuples*. For the set of *complete tuples*, it is used as the training set to train a KNNI (Section 3.1) imputation model, which is employed to predict the missing values of the *incomplete tuples*. For the part of *incomplete tuples*, they are partitioned into several subsets, named *incomplete subsets*, according to their missing degree. Each incomplete subset is respectively taken as a test set, and is filled with the previously trained KNNI model. To gradually extend the training samples, the imputed tuples with similar missing degree are merged into the set of *complete tuples* for the next KNNI training, which improves the generalization ability (Xu & Raginsky, 2017) of the model. As the key of the RESI framework, the expansion of training set is an iterative process with fixed times, which is determined by the number of *incomplete subsets*. Each iteration is self-consistent, which means that the imputing technique performed in each iteration is essentially the same, for example, all the iterations employ KNNI to impute *incomplete tuples*. For clarity, Algorithm 1 presents the main steps of the computation.

RESI takes the dataset with missing items as input and generates one complete dataset after imputing the missing items. Following the division of the original dataset, RESI accomplishes the imputation in three steps: (1) *tuple partition*; (2) *incomplete subsets imputation*; (3) *cross correction*. The final filling results are obtained by averaging the predicted values of (2) and (3). We describe the three steps in detail as follows.

4.2. Tuple partition

There are a lot of incomplete tuples in the real dataset, and typically their missing degree varies greatly. It is not reasonable to deal with all incomplete tuples at the same time during the imputing process. Before processing the tuples with different missing degree, it is necessary to create corresponding criteria for evaluating the missing degree of tuples. Practice shows that different attributes have different effects on subsequent analysis, which means, in a dataset instance, different missing items of tuples will also have different impact on subsequent modeling. We define an indicator called *tuple integrity rate*, which is used to measure the impact of attributes and their missing situation on the tuple. The definition is given as follows:

Algorithm 1: Algorithm of Subset Imputation.

Input: $\mathcal{D}$: a dataset of relation schema having missing value to impute; m: number of incomplete subsets
Output: $\mathcal{D}'$: dataset with filled values

```
Begin:
$CT_0$=ExtractCompleteTuples($\mathcal{D}$)
$ICT$=ExtractIncompleteTuples($\mathcal{D}$)
GenerateTuplePartitions($ICT$, method=EWM, output=$\mathcal{T}\{T_1, T_2, \cdots, T_m\}$)
for i in 1:$m$ do
    $T_i'$=KNNImputation(train_set=$CT_{i-1}$, test_set=$T_i$)
    $CT_i$=merge($CT_{i-1}$, $T_i'$)
Do cross validation:
for i in 1:$m$-1 do
    $T_i''$=KNNImputation(train_set=$CT_m$, test_set=$T_i$)
$T_m''$=KNNImputation(train_set=$CT_0$, test_set=$T_m$)
for i in 1:$m$ do
    $CT_i$=merge($CT_{i-1}$, mean($T_i'$, $T_i''$))
$\mathcal{D}' = CT_m$
return $\mathcal{D}'$
```

Definition 5 (*Tuple Integrity Rate*). Given a tuple t with s attributes $a_1, a_2, \ldots, a_s$, whose weights are $w_1, w_2, \ldots, w_s$, the **tuple integrity rate** of t, denoted as r^t, is calculated as $r^t = \sum_{i=1}^{s} \varepsilon_i w_i$, where if a_i is a missing item, $\varepsilon_i = 0$, otherwise $\varepsilon_i = 1$.

For the computation of $w_1, w_2, \ldots, w_s$, we employ the method of EWM which is described in Section 3.2.

Example of computing tuple integrity rate Suppose we are given the relevant attributes of name, age, place of birth (*birtplace*), place of work (*workplace*), degree, occupation and the length of service, and we want to model wages. Intuitively, each attribute contributes differently to the model. In other words, the imputation accuracy of the missing items on various attributes has different impact on the subsequent data analysis. In current example, the importance of *birthplace* varies from that of *workplace* in the modeling of wages. Assuming that with EWM, we compute the weight of *birthplace* is 0.10 and that of *workplace* is 0.11, and the *birthplace* is a missing item in tuple t_a, while the *workplace* is a missing item in tuple t_b. Both t_a and t_b have no other missing items. Then, the *tuple integrity rate* of t_a and t_b are 0.90 and 0.89, respectively.

According to Definition 5, the *tuple integrity rate* reflects reversely the missing degree of a tuple. For a dataset, its missing degree can be measured with the *mean integrity rate* of all its tuples, which can be formally defined as:

Definition 6 (*Mean Integrity Rate*). Let $\mathcal{D} = \{t_1, t_2, \ldots, t_n\}$ be a dataset. For each tuple $t_i \in \mathcal{D}$ $(0 \le i \le n)$, the *tuple integrity rate* of it is r_i^t.

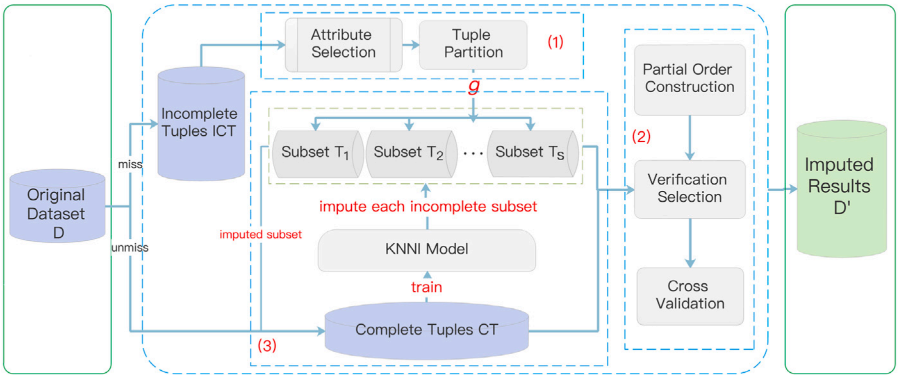


Fig. 1. Overview of RESI framework.

Algorithm 2: Algorithm of Generating Tuple Partitions.

```
Input: $CT$: subset with complete tuples; $ICT$: subset with incomplete tuples; $m$: number of incomplete subsets; $s$: number of attributes
Output: $\mathcal{T}$: a queue of subsets
Begin:
$W$=ComputeAttributeWeights (train_set=$CT$, method=EWM, output=$\{w_1, w_2, \cdots, w_s\}$)
Do Intergrity Computation:
for $t_i$ in $ICT$ do
    $t_i = \{a_1, a_2, \cdots, a_s\}$
    $r_i = 1$
    for j in 1: $s$ do
        if $a_1$=0 then
            $r_i = r_i - w_j$
$\mathcal{P}$=SortIncompleteTuples(object=$ICT$, by=$r$, order=descending)
$\mathcal{T}$=GenerateTuplePartition($\mathcal{P}$,$m$)
return $\mathcal{T}$
```

Then the **mean integrity rate** $mir^{\mathcal{D}}$, for measuring the integrity rate of dataset $\mathcal{D}$, is computed as $mir^{\mathcal{D}} = \frac{\sum_{i=0}^{n} r_i^t}{n}$.

After calculating the *tuple integrity rate* of all tuples in the dataset, the tuples are sorted in descending order of their *tuple integrity rate*. Depending on the sorting result, the tuples are partitioned into several subsets with different degree of integrity. Formally, given a dataset $\mathcal{D}$ whose incomplete tuples form a set $ICTs$, we utilize their *tuple integrity rate* to divide ICTs into m partitions $\{T_1, T_2, \ldots, T_m\}$, these partitions satisfy (i) $T_1 \cup T_2 \ldots \cup T_m = ICTs$; (ii) $T_i \cap T_j = \phi$ $(i \neq j)$; (iii) given a $T_i \in ICTs$, all members contained in it are close to each other at *tuple integrity rate*; (iv) $\forall t_x \in T_i$, $\forall t_y \in T_j$ and $i < j$ satisfying $r_x^t > r_y^t$. When partitioning, the choice of parameter m has a large effect on the performance of imputation. Unfortunately, this parameter cannot be known beforehand, therefore, we usually apply the experimental method to get the optimal value of m. Specifically, we set a series of values for m in the experiment and choose the one corresponding to the best experimental results as the final value of m. It is clear that the optimal m varies from dataset to dataset, so for the sake of conciseness, we will not discuss this further. Algorithm 2 gives the pseudocode to represent the core idea of our partitioning method.

4.3. Incomplete subsets imputation

This section aims to impute the incomplete tuples in all partitions by using the base imputation technique, KNNI. The partitions are generated according to the *tuple integrity rate* of tuples by Algorithm 2. We initially leverage KNNI to train an imputation model by taking the original complete tuples as the training set, and then use the model to predict the missing items partition-by-partition with finite iterations.

In the following iteration, before training, the training set is extended by appending the complete tuples imputed in the previous iteration. Each iteration selects the first partition as the test set from the incomplete partitions and removes the selected partitions from the incomplete partition set. After being imputed by KNNI, the tuples in the test set are merged into the training set for training a new KNNI model for the next iteration.

Commonly, incomplete tuples with high *tuple integrity rate*, i.e. low missing degree, can be well imputed with a small number of complete tuples. In contrast, for incomplete tuples with high missing degree, to ensure the effect of imputation, we need more complete tuples as the training samples. This observation is also reflected in the process of the iteration: the partitions are generated according to the descending order of *tuple integrity rate*, and the test set selected in the iteration is also carried out in the partition order. Each iteration enlarges the training samples by adding the imputed tuples into the training set, which guarantees the incomplete tuples with higher missing degree can be imputed with more samples. As we can see from the experiment results, this approach greatly improves the accuracy and reliability of imputation.

4.4. Cross correction

Although in the iteration, we adopt the strategy of imputing the incomplete tuples with low missing degree first and those with high missing degree later, incomplete tuples with low missing degree may be under-learned due to too few observed samples during the imputing process, and those with a high missing degree may also be over-learned because of over-reliance on the imputed samples. These will contribute to the imputing model returning deviated results, therefore, to improve the effects of imputation, it is necessary to correct the deviation of imputing results in each iteration.

In RESI, the modified k-fold Cross Validation (k-CV) (Grimm, Mazza, & Davoudzadeh, 2018) is utilized to reduce the deviations, where k is the number of incomplete partitions. For the first $k-1$ partitions, we take each incomplete partition $T_i(1 \leqslant i \leqslant (k-1))$ as the validation set respectively, and the union of the remaining $k-1$ partition and the initial complete subset CT_0 as the training set, and then perform the KNNI. The mean value of these predictions and the imputing results generated in Section 4.3 are regarded as the final imputing values from the incomplete tuples contained in the current partition $T_i(1 \leqslant i \leqslant (k-1))$. For the last incomplete partition T_k, the KNNI is performed on the training set of the initial complete subset CT_0 and the validation set of the partition itself. The mean values between these predicted results and the previous predictions generated in Section 4.3 are taken as the final imputing values for the incomplete tuples.

For a dataset, *cross correction* can be applied to each incomplete tuple to achieve higher accuracy. However, when the dataset is large in size or grows with high dimensions, it will cost high if we process the incomplete tuples one-by-one. Therefore, this paper introduces the partial order (Chai, Li, Li, Deng, & Feng, 2018) theory to define a partial order structure on each partition:

Definition 7 (*Partial Order Structure*). Let $T = \{(t_1, \ldots, t_i) \mid t \in ICTs\}$ denote a set of incomplete tuples in a partition, and the tuples in T are sorted in descending order of their *tuple integrity rate*. Given a pair of tuples (t_q, t_{q+1}) (t_q denotes the q-th tuple of T) and their *tuple integrity rate*, r^t_q and r^t_{q+1} (see Definition 5 in Section 4.2), there exists $r^t_q \geqslant r^t_{q+1}$, which is expressed as $t_q \succ t_{q+1}$. This means t_q and t_{q+1} are comparable and t_q is superior to t_{q+1} or t_{q+1} is inferior to t_q. Thus, for all incomplete tuples in T, a **partial order structure** $(T, \succ)$ can be established to represent the relationship between them.

$(T, \succ)$ facilitates the selection of validation sets for cross correction. For example, suppose we are modeling the wages in a dataset, the weights of occupation and age are respectively w_{ocp} and w_{age}, and $w_{ocp} > w_{age}$. Given two tuples t_a and t_b, t_a misses age and t_b misses occupation, their *tuple integrity rate* satisfies $r^t_a > r^t_b$. After being imputed, the tuples are denoted as t'_a and t'_b.

The impact of a missing item on a tuple decreases as its *tuple integrity rate* increases, which means, for the subsequent modeling, the higher the *tuple integrity rate* is, the smaller the impact of missing items will put on the modeling results. When the computational cost is limited, t_a and t_b can be only chosen either of them to conduct the cross correction. Since $t_a \succ t_b$, correspondingly, in the previous step the imputation accuracy of t'_a is slightly higher than that of t'_b, which indicates t_b is more necessary to be revised. Therefore, in order to reduce the cost, when the number of tuples to be verified in each partition is fixed, the selection of tuples with low *tuple integrity rate* for cross validation can play a more positive role in following data modeling.

Whether it is full-element correction for small datasets or partial cross-correction for large datasets, experiments have proved that the cross-correction of deviation can effectively avoid over- and under-learning while minimizing the computing costs, so as to obtain imputing results with smaller deviations.

4.5. Complexity of algorithm

Given an incomplete dataset D, containing s attributes and t tuples. In D, the number of incomplete tuples and missing items is c and n respectively. The time complexity of RESI is specifically analyzed as follows:

1. Weight calculation, during which all data for each attribute is traversed and normalized, so the time complexity is $O(s \times t)$.
2. Tuple Partition. Each tuple in the dataset is traversed to determine if it is an incomplete tuple. The determination of an incomplete tuple requires traversing the data items in the tuple to determine whether the tuple contains missing items, and calculating the integrity rate of each tuple. Therefore, the time complexity is $O(s \times t)$.
3. Missing value imputation. It is necessary to traverse and fill all missing items in turn, and cross-verify the correction strategy for each tuple. Then, the time complexity of a certain missing item predicted and corrected by KNN is approximately $O(t)$, so the time complexity of filling all missing items is $O(n \times t)$.

Hence, the overall time complexity is $T = O(s \times t) + O(s \times t) + O(n \times t)$, and when $n \gg s$, the complexity approximates $O(n \times t)$.

5. Experiments and analysis

5.1. Datasets and experimental settings

In order to fully control the missing items and effectively verify our proposed model, we selected four complete datasets (Abalone, Iris, Lymphography and Parkinsons) without missing items. All datasets are from the University of California Irvine (UCI), and each of which has with different number of attributes and instances that are listed in Table 3. In the experiment, in order to better testify the accuracy of the results, we randomly selected a certain proportion of items from the complete datasets and set them as missing items. The method is described in detail as follows.

Table 3
Datasets used in this thesis.

No	DataSet	Attributes	Instances	Area	Type
1	Abalone	8	4177	Life	Mixed-type
2	Iris	5	150	Biophysical	Mixed-type
3	Lymphography	19	148	Life	Categorical
4	Parkinsons	23	195	Life	Numerical
5	KDD Cup 99	42	500,000	Computer	Mixed-type

Mechanism of generating missing data In order to evaluate the sensitivity of each method to the *missing rate* $MRate_D$ (see Definition 2) and the *complete ratio* $CRatio_D$ (see Definition 3), we use the random generator to remove 5% to 50% items on the *complete ratio* of $\frac{1}{2}$, $\frac{1}{3}$ and $\frac{1}{6}$, respectively. Thus, we create the simulated datasets with different missing degree for the experiments, and represent them in the form of $Dataname\_CRatio_D\_MRate_D$. For example, Iris_$\frac{1}{2}$_5% explains the simulated Iris dataset contains 50% complete tuples and 50% incomplete tuples, and its total missing items takes 5% of the entire Iris data.

5.2. Evaluation metrics

Following the most prevalent evaluation metrics, we assess the performance of imputing the numerical variables with the Root Mean Squared Error (RMSE) and Mean Absolute Percent Error (MAPE) (Kyureghian, Capps, & Nayga, 2011), and the categorical variables with F1 − scores (F_1) (Zhang, Wang, & Zhao, 2015) and the Proportion of Falsely Classified entries (PFC) (Stekhoven & Bühlmann, 2012).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{j=1}^{\mathrm{N}} \left(y_j - \hat{y}_j\right)^2}{\mathrm{N}}} \tag{6}$$

$$\mathrm{MAPE} = \sum_{j=1}^{\mathrm{N}} \left| \frac{y_j - \hat{y}_j}{y_j} \right| * \frac{100}{\mathrm{N}} \tag{7}$$

where y is the real value, $\hat{y}$ is the predicted value, and N is the total number of missing items.

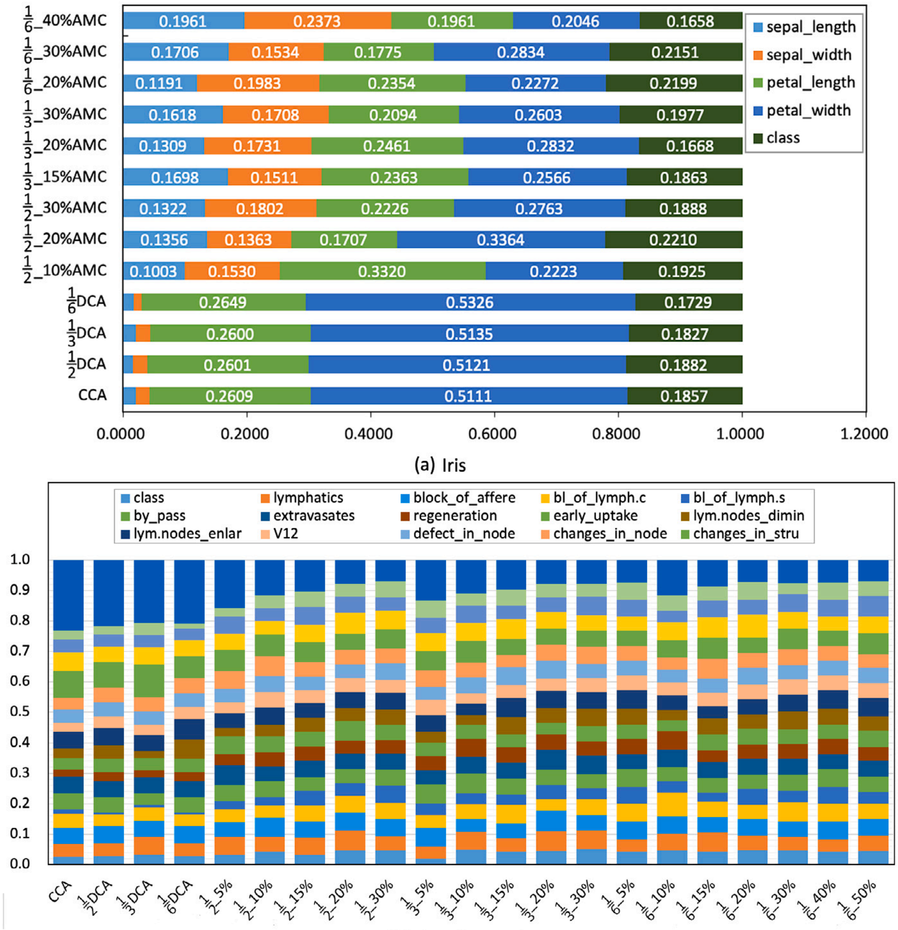
$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{8}$$

In the formula, TP represents the number of *true* items that are predicted to be *true*; FP is the number of *false* items that are predicted to be *true*; FN indicates the number of *true* items that are predicted to be *false*.

$$PFC = \frac{\mathrm{N_{falseC}}}{\mathrm{N_{falseC}} + \mathrm{N_{trueC}}} \tag{9}$$

in which, $\mathrm{N_{falseC}}$ represents the number of falsely predicted items and $\mathrm{N_{trueC}}$ stands for the number of truly predicted items.

Among the above four metrics, RMSE and MAPE measure the imputation deviation generated by a model whose imputation effect generally upgrades as they decrease. They are usually used to compare the performance of different models in the same dataset, especially MAPE. For example, if model A's MAPE is larger than model B's, means B outperforms model A. In other words, just saying MAPE = 20% of a model cannot judge whether the model performs well or badly. For a particular model, the closer its PFC is to 0 or F_1 is to 1, indicates the better performance it has.



(a) Iris

(b) lymphography

Fig. 2. Comparison of Weights in Iris and Lymphography.

5.3. Implementation and results

5.3.1. Weight comparison

In order to further examine the uncertain data containing missing items, which is likely to cause large deviation to the subsequent analysis, we take the EWM to calculate the weights of attributes in the experiment. A variety of methods, including CCA, DCA and AMC in different missing cases, are used to weight the attributes in simulated datasets with multiple missing degrees of Iris and Lymphography. Among the methods, CCA is a comparative experiment on the original complete datasets without missing items. DCA calculates weights after deleting all the tuples containing missing items and is divided into $\frac{1}{2}DCA$, $\frac{1}{3}DCA$ and $\frac{1}{6}DCA$. AMC performs weights evaluation directly on all data containing missing items, which can be recorded as $CRatio_D\_MRate_DAMC$ or $CRatio_D\_MRate_D$, depending on the missing situation. For example, $\frac{1}{2}\_10\%AMC$ or $\frac{1}{2}\_10\%$ means the calculation is conducted directly on the dataset with *complete ratio* of $\frac{1}{2}$ and *missing rate* of 10%.

The experimental results are displayed in Fig. 2, in which Fig. 2(a) represents the attribute proportion pileup histogram of CCA, DCA, AMC in various cases for calculating EWM weight on Iris, and Fig. 2(b) illustrates the attribute proportion pileup histogram of these methods on Lymphography. As can be seen from the figure, the calculation results of three groups of DCA are similar to those of CCA, whereas, the results of AMC are all deviated greatly, and the more serious the missing is, the greater the deviation will be. This proves that the results obtained from the analysis of certain information (complete tuples) are closer to the real results, while the uncertain data with missing items leads to the deviation, and further the deviation goes up as the *missing rate* increases. Therefore, the weight calculation of the model in our experiments is only carried out on the complete tuples of the current simulated dataset.

5.3.2. Imputing numerical-missing data

We first examine the effectiveness of RESI in the imputation of numerical-missing data. It is compared with ROF, EMI and KNNI in Parkinsons datasets with $\frac{1}{2}$ *complete ratio* and several different *missing rate*, and RMSE is utilized as the evaluating indicator, which measures the deviation between the data imputed and the original data.

Fig. 3 presents the results of the experiment. According to Fig. 3(a), we observe that RESI and KNNI are basically in the same changing trend and showing the smallest deviation. However, compared to KNNI, the base imputer, RESI wins the better effects. With the increase of the *missing rate*, the deviation created by EMI increases the fastest, possibly because it relies too much on the correlations among attributes (Dougherty et al., 2017).

In addition, to investigate the effectiveness of error correction process, we compared the cross-validation supported RESI, denoted as RESI, with the non-cross-validation support RESI, denoted as RESI* in Fig. 3(b). The figure suggests that, on the dataset with same *missing rate*, the improved RESI with error correction significantly outperforms RESI*, which does not use. This can be explained as cross-validation technique can effectively alleviate the impact of errors caused by under- and over-learning.

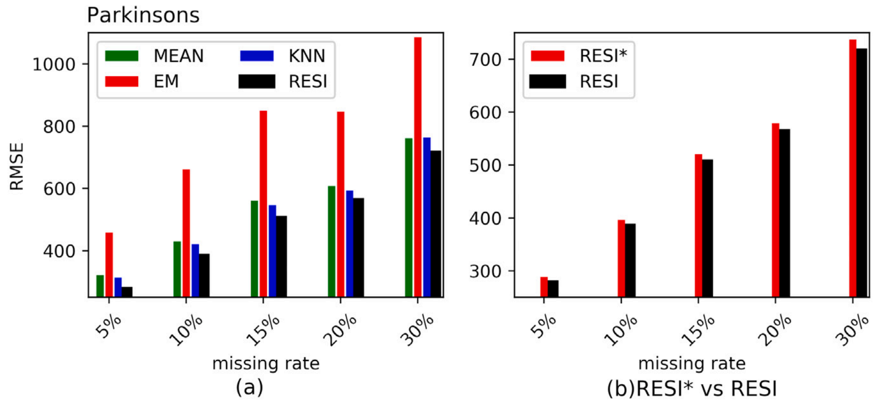


Fig. 3. Comparison of imputation in numerical data.

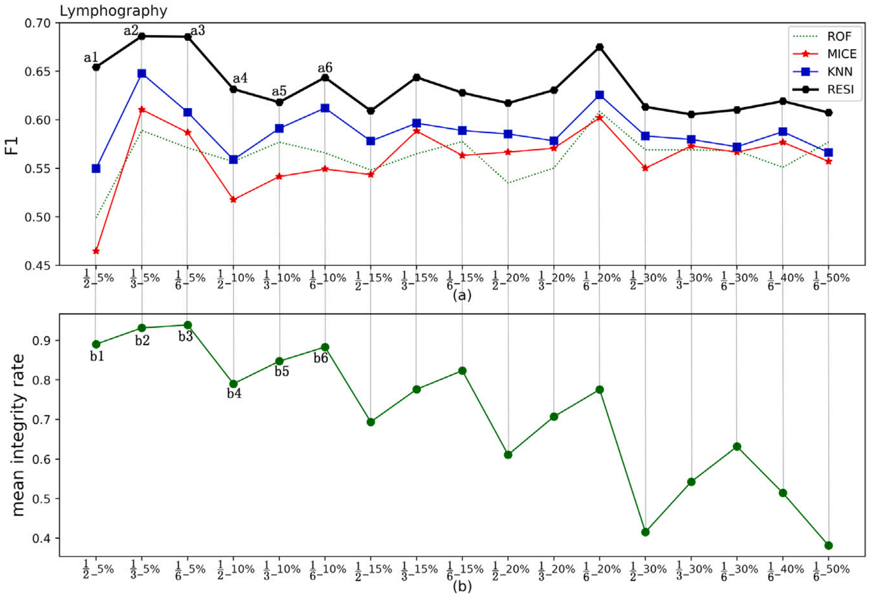


Fig. 4. Comparison of imputation with categorical data.

5.3.3. *Imputing categorical-missing data*

We have testified the effectiveness of imputing categorical-missing data by conducting experiments on the public available dataset, Lymphography, which contains 19 attributes, all of which are categorical data. In the experiment, ROF, MICE and KNNI were compared with our RESI, and F_1 was applied to evaluate the accuracy.

Fig. 4 reveals the experimental results. Fig. 4(a) shows the accuracy of imputation, and to intuitively observe the variation of accuracy with the integrity of dataset, Fig. 4(b) displays the integrity rate of simulated datasets. As can be seen from Fig. 4(a), RESI has all the outstanding advantages in any situation. In comparison with other methods, it guarantees that the accuracy in all simulated datasets is greater than 0.6 and has the minimal sensitivity to the *missing rate*. In other words, as the *missing rate* increases, the declining trend of RESI's accuracy is relatively gentle.

Clearly, with the growth of the *missing rate*, accuracy does not decline or increase linearly. It is because the imputation accuracy is somewhat affected by the average missing degree of each tuple in the dataset. Combining Fig. 4(a) and (b), it is not difficult to find that the trend of accuracy is similar to the *mean integrity rate* of datasets. Higher *mean integrity rate*, i.e., lower average missing degree, is associated with better imputation effect.

Take the three points a_1, a_2 and a_3 in Fig. 4(a) as the examples, whose *missing rate* is the same and complete tuples in the datasets are $\frac{1}{2}$, $\frac{1}{3}$ and $\frac{1}{6}$. Intuitively, the reduction of complete tuples will make the imputing accuracy decline, that is, $F_1^{a_1} > F_1^{a_2} > F_1^{a_3}$. Nevertheless, the actual is $F_1^{a_1} < F_1^{a_2} < F_1^{a_3}$. The reason is that the integrity rate of the three corresponding simulated datasets gradually gets greater, which is uncovered by Fig. 4(b) at points b_1, b_2 and b_3.

Additionally, the accuracy is also affected by the number of complete tuples in the dataset. For example, in Fig. 4(a), at the point of a_4, a_5 and a_6, the *missing rate* is the same, according to $b_4 < b_5 < b_6$, $F_1^{a_4} < F_1^{a_5} < F_1^{a_6}$ should exist, however, the number of complete tuples in the dataset corresponding to point a_4 is greater than that corresponding to point a_5. In the combination of influence of the number of complete tuples and average missing degree, the actual accuracy satisfies $F_1^{a_5} < F_1^{a_4} < F_1^{a_6}$.

5.3.4. *Imputing mixed-type missing data*

To verify the effect of our proposed framework on imputation of mixed-type missing dataset, we compare MICE, DTI, missForest, KNNI and RESI in this group of experiments, which were implemented on the datasets of Iris and Abalone. Both datasets have only one categorical attribute and more numerical attributes. RMSE and MAPE are utilized to assess the effectiveness of the imputed numerical data, whereas F_1 and PFC are employed to evaluate the imputed categorical data. Here, *predictive mean matching* (pmm) and *Random Forest* are used as the base filling methods in MICE respectively, which are denoted as MICE

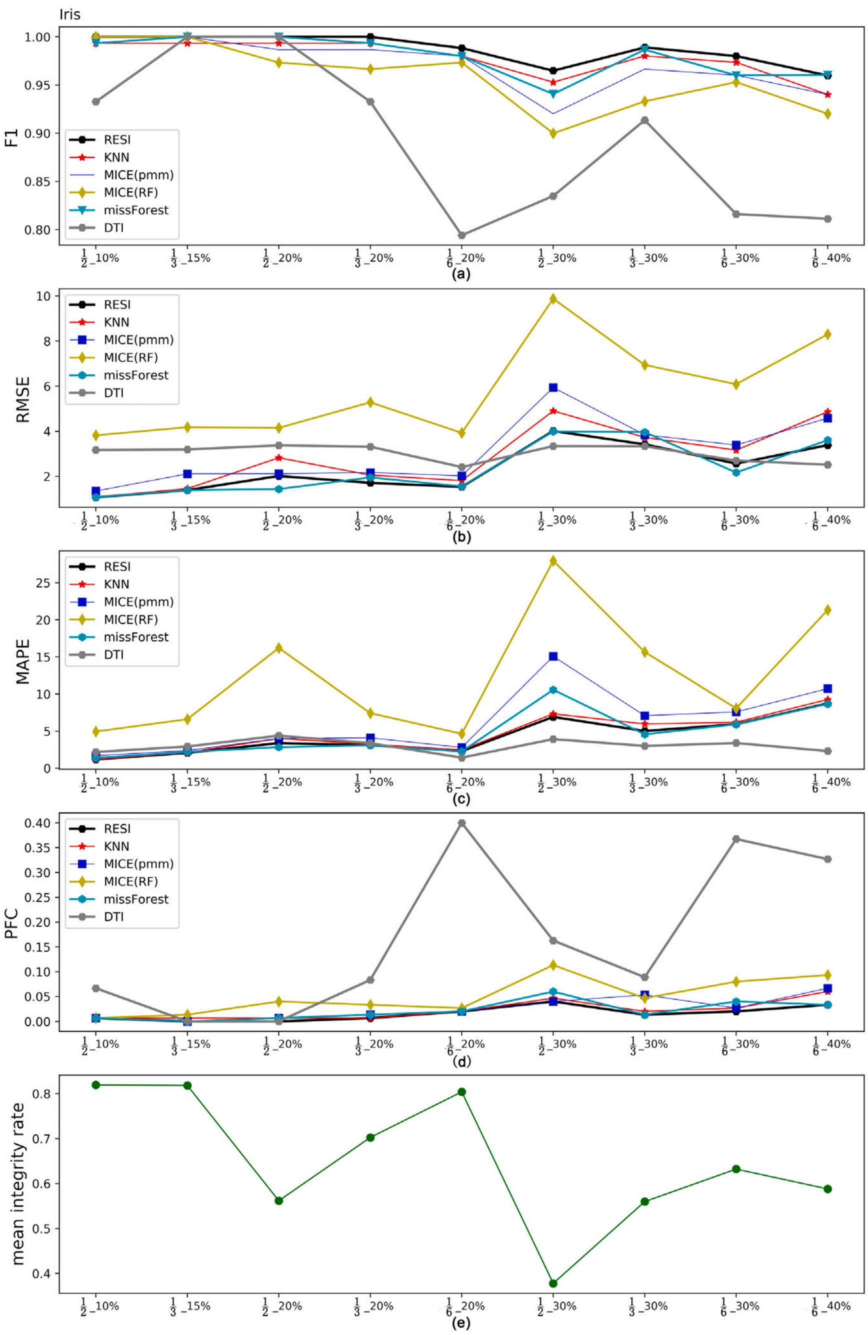


Fig. 5. Results of imputation in Iris.

(pmm) and MICE(RF). The results of this comparison are given in Fig. 5 (Iris) and Fig. 6 (Abalone), which indicate that RESI is superior to the five competitive methods in imputing mixed-type missing dataset with both numerical and categorical data.

By comparing Fig. 5(a)~(d), we find that the comprehensive effect of KNNI in all cases is significantly better than that of the other three methods except RESI and missForest. This means that it is appropriate to select KNNI as the basic imputer of RESI. Because on the basis of KNNI, RESI not only retains the original precision of the imputer, but also effectively reduces the deviation of imputation. With the increase of the missing degree, RMSE and MAPE of the DTI imputation are the most stable. To be specific, they almost change around 3.5. However, the imputing effect of DTI on categorical data is obviously poor, more seriously when the dataset with a small number of complete tuples and a high *missing rate*. Meanwhile, we also find that the missForest has a relatively good completing effect in the processing of both categorical and numerical missing data, but it could not be able to widely used due to its longer running time.

In contrast to missForest, our RESI have very similar performance of imputation, but it is more tactful in filling categorical-typed missing data. Combining with Fig. 5(e), we can further investigate the impact of *mean integrity rate* of the dataset mentioned in Section 5.3.3 on the imputing effect. With the decrease of the *mean integrity rate*, i.e., the upgrade of the average missing degree of the dataset, RMSE and MAPE gradually increase while F_1 and PFC continuously decrease. To some extent, the impact of the *mean integrity rate* on the imputing effect is higher than that of the *missing rate* and *complete ratio* of a dateset

The experimental results on the Abalone dataset shows the similar pattern. As we can observe from Fig. 6(a)~(c), in either case the RESI, with relatively smaller RMSE, minimum PFC and highest F_1, has

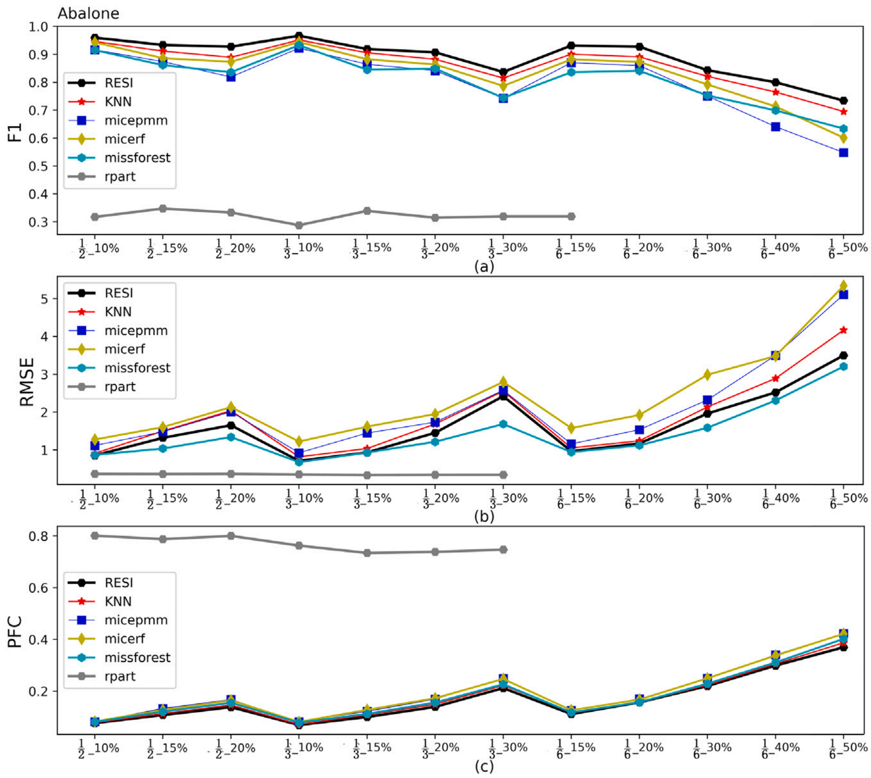


Fig. 6. Results of imputation in Abalone.

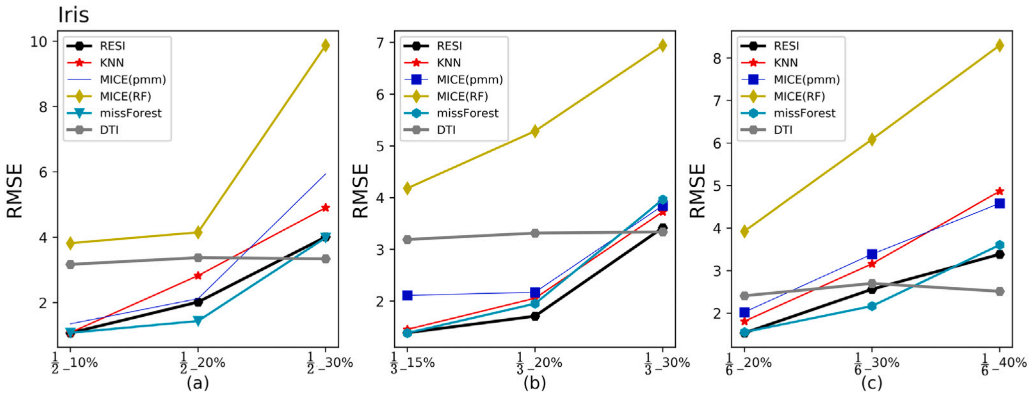


Fig. 7. Sensitivity with missing rate in Iris.

significantly better imputation effectiveness than the other methods. Even the *missing rate* is as high as 30%, RESI performs in F_1 and PFC almost as well as MICE and KNNI do at 10% *missing rate*.

5.3.5. *Sensitivity analysis*

It can be seen from the above experimental results that: (1) Given the *complete ratio* of the dataset, the increase of the *missing rate* will greatly affect the imputing effect. As shown in Fig. 7(a)~(c), RMSE increases with the increase of the *missing rate*, and all the comparative methods are highly sensitive to the *missing rate*. However, the effect of RESI on the dataset with higher *missing rate* can still be as good as that of the competitive methods on the dataset with lower *missing rate*. (2) When the *missing rate* of the dataset is constant, the imputing accuracy is not proportional to the *complete ratio*. Because the *missing rate* is the same, the missing degree for incomplete tuples will be more serious for the simulated dataset with high *complete ratio*. To measure the missing degree of a dataset as a whole, this paper proposes a new evaluation metric of missing degree of a dataset, namely, the *mean integrity rate*.

According to the experimental results, we find that the trend of imputing effect is almost proportional to the *mean integrity rate*, that is, as the *mean integrity rate* grows, the deviation of numerical imputation gets low and the accuracy of imputing categorical data turns high. Compared with the *complete ratio* as well as the *missing rate*, the *mean integrity rate* is a better quantitative representation of the missing degree of a dataset. The *mean integrity rate* is calculated according to the *tuple integrity rate* of each tuple, which also confirms the partial order theory mentioned above to some extent, i.e., tuples with higher *tuple integrity rate* have better imputing effect. RESI is only slightly better in terms of sensitivity to *mean integrity rate*. As with other methods, it cannot effectively solve the problem that the effect of imputation degrades with the decline of the *mean integrity rate*.

5.3.6. *Classification effect after imputation*

In order to further verify the improvement of the missing value imputation on data quality, 500,000 complete training data from KDD Cup 99 is selected as the original dataset for the experiment. We

Table 4
Comparison of classification accuracy.

No	Training data	Method	Accuracy
1	KNNI_K99Train	KNN	75.31%
3	MICE_K99Train	MICE	74.82%
4	missForest_K99Train	missForest	80.14%
2	XGBoost_K99Train	XGBoost	83.07%
5	RESI_K99Train	RESI	83.59%
6	Missing data	Untreated	67.35%

randomly select 10% of the data as the original test set (K99Test), and the remaining 90% as the original training set (K99Train). On K99Train, we delete 15% of the records by random generator and gain a simulated set K99Train′ to be imputed. The experiment contains four steps:

Step 1: Impute the missing data of K99Train′ with KNNI, MICE, missForest, XGBoost and RESI respectively, and obtain five imputed training sets after imputation, which are KNNI_K99Train, MICE_K99Train, missForest_K99Train, XGBoost_K99Train and RESI_K99Train.

Step 2: Train classification models on the five imputed datasets and K99Train′ with decision tree.

Step 3: after deleting the class identification, which is used to indicate that the connection record is normal or a specific type of attack, in K99Test, classify the test set by the six classification models respectively, and generate six result sets.

Step 4: Compare the six result sets with the original test set K99Test, we get the classification accuracy of the six models.

Table 4 presents the six training sets, the comparative methods and their corresponding accuracy of classification. It is clear from the table that RESI has the highest accuracy followed by XGBoost, and all methods are superior to missing data that is not processed. Although RESI and XGBoost have similar results, XGBoost is more expensive and takes longer to compute. Consequently, after imputing with RESI, data quality is significantly improved, and compared with other models, RESI has better ability to handle data quality and perform well in classification tasks.

6. Conclusion

In this work, we propose a novel region-splitting framework, RESI, for missing value imputation on both numerical and categorical data. We devise a EWM-based tuple partition method to divide the tuples into different partitions so that we can fill the missing values by partitioning. To improve the imputation accuracy, we utilize the cross-validation to modify the predicted value and prove that cross-validation can correct errors effectively. We define the *mean integrity rate* which indicates the missing degree of a dataset and reflects more comprehensively the missing status of the dataset. The experiments, carried on different real datasets from different areas, show that our framework achieves good effects and outperforms competitive methods. In our future work, we plan to further optimize the proposed method to improve the imputation accuracy and the computational performance, and also we will conduct more extensive experiments on real datasets in order to find better imputation solution.

CRediT authorship contribution statement

Dunlu Peng: Conceptualization, Formal analysis, Funding acquisition. **Mengping Zou:** Writing - original draft, Writing - editing. **Cong Liu:** Writing - review. **Jing Lu:** Data curation, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grant No. 61772342 and No. 61703278. We would like to express our special thanks to the members in our lab for their valuable discussion on this work.

References

Chai, C., Li, G., Li, J., Deng, D., & Feng, J. (2018). A partial-order-based framework for cost-effective crowdsourced entity resolution. *VLDB Journal*, *27*(6), 745–770. http://dx.doi.org/10.1007/s00778-018-0509-6.

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, *8*(1), 6085. http://dx.doi.org/10.1038/s41598-018-24271-9.

Chen, S. F., Wang, S., & Chen, C. Y. (2012). A simulation study using EFA and CFA programs based the impact of missing data on test dimensionality. *Expert Systems with Applications*, *39*(4), 4026–4031. http://dx.doi.org/10.1016/j.eswa.2011.09.085.

Cui, Y., Feng, P., Jin, J., & Liu, L. (2018). Water resources carrying capacity evaluation and diagnosis based on set pair analysis and improved the entropy weight method. *Entropy*, *20*(5), http://dx.doi.org/10.3390/e20050359.

Derek, T. J., Lyndsay, S., & John, R. L. (2019). Handling missing data in self-exciting point process models. *Spatial Statistics*, *29*, 160–176. http://dx.doi.org/10.1016/j.spasta.2018.12.004.

Dohoo, I. R. (2015). Dealing with deficient and missing data. *Preventive Veterinary Medicine*, *122*(1–2), 221–228. http://dx.doi.org/10.1016/j.prevetmed.2015.04.006.

Dong, Y. J., & Liu, T. Z. (2013). Parameter optimization based on genetic algorithm in the research of equivalent pruning effect on fuzzy decision tree. *Advanced Materials Research*, *756–759*, 3809–3813. http://dx.doi.org/10.4028/www.scientific.net/amr.756-759.3809.

Dougherty, E. R., Carlson, C. J., Blackburn, J. K., & Getz, W. M. (2017). Correction: A cross-validation-based approach for delimiting reliable home range estimates [Movement Ecology, 5, (2017) (19)] DOI: 10.1186/s40462-017-0110-4. *Movement Ecology*, *5*, 26. http://dx.doi.org/10.1186/s40462-017-0116-y.

Gao, H., Jian, S., Peng, Y., & Liu, X. (2017). A subspace ensemble framework for classification with high dimensional missing data. *Multidimensional Systems and Signal Processing*, *28*(4), 1309–1324. http://dx.doi.org/10.1007/s11045-016-0393-4.

Grimm, K. J., Mazza, G. L., & Davoudzadeh, P. (2018). Model selection in finite mixture models: A *k*-fold cross-validation approach. *Structural Equation Modeling A Multidisciplinary Journal*, *24*, 1–11, Retrieved from http://www.tandfonline.com/doi/abs/10.1080/10705511.2016.1250638?journalCode=hsem20.

Johannesson, P. (2002). A method for transforming relational schemas into conceptual schemas. *Data Engineering, International Conference*, *19*, 0–201. http://dx.doi.org/10.1109/icde.1994.283030.

Kezban, Y. S., Irina, S. D., Karen, S., & Ryan, B. (2018). Incomplete information imputation in limited data environments with application to disaster response. *European Journal of Operational Research*, *269*(2), 466–485. http://dx.doi.org/10.1016/j.ejor.2018.02.016.

Kyureghian, G., Capps, O., & Nayga, R. M. (2011). A missing variable imputation methodology with an empirical application. *Advances in Econometrics*, *27A*, 313–337. http://dx.doi.org/10.1108/S0731-9053(2011)000027A015.

Li, H. H., Shao, F. F., & Li, G. Z. (2014). Semi-supervised imputation for microarray missing value estimation. In *Proceedings - 2014 IEEE international conference on bioinformatics and biomedicine* (pp. 297–300). http://dx.doi.org/10.1109/BIBM.2014.6999172.

Liang, J., & Shi, Z. (2004). The information entropy, rough entropy and knowledge granulation in rough set theory. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *12*(01), 37–46. http://dx.doi.org/10.1142/s0218488504002631.

Ma, Z. H., & Chen, G. H. (2018). Bayesian methods for dealing with missing data problems. *Journal of the Korean Statistical Society*, *47*(3), 297–313. http://dx.doi.org/10.1016/j.jkss.2018.03.002.

Madan, L. Y., & Basav, R. (2018). Handling missing values: A study of popular imputation packages in R. *Knowledge-Based Systems*, *160*, 104–118. http://dx.doi.org/10.1016/j.knosys.2018.06.012.

Ohlander, R., Price, K., & Reddy, D. R. (1978). Picture segmentation using a recursive region splitting method *. *Computer Graphics & Image Processing*, *8*(3), 313–333.

Pan, M. (2011). Based on kernel function and non-parametric multiple imputation algorithm to solve the problem of missing data. In *2011 International conference on management science and industrial engineering* (pp. 905–909). http://dx.doi.org/10.1109/MSIE.2011.5707554.

Purwar, A., & Singh, S. K. (2015). Expert systems with applications hybrid prediction model with missing value imputation for medical data. *Expert Systems with Applications*, *42*(13), 5621–5631. http://dx.doi.org/10.1016/j.eswa.2015.02.050.

Rahman, M. G., & Islam, M. Z. (2010). A decision tree-based missing value imputation technique for data pre-processing. In *Conferences in research and practice in information technology series, vol. 121* (pp. 41–50).

Rahman, M. G., & Islam, M. Z. (2013). Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques. *Knowledge-Based Systems*, *53*(9), 51–65. http://dx.doi.org/10.1016/j.knosys.2013.08.023.

Rahman, M. G., & Islam, M. Z. (2016). Missing value imputation using a fuzzy clustering-based EM approach. *Knowledge and Information Systems*, *46*(2), 389–422. http://dx.doi.org/10.1007/s10115-015-0822-y.

Rubin, D. B. (2009). Multiple imputation for nonresponse in surveys. *Journal of Marketing Research*, *137*(1), 180.

Schafer, J. L. (2010). Analysis of incomplete multivariate data. Retrieved from http://books.google.com/books?hl=en&lr=&id=3TFWRjn1f-oC&pgis=1.

Sentas, P., & Angelis, L. (2006). Categorical missing data imputation for software cost estimation by multinomial logistic regression. *Journal of Systems and Software*, *79*(3), 404–414. http://dx.doi.org/10.1016/j.jss.2005.02.026.

Stekhoven, D. J., & Bühlmann, P. (2012). Missforest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118. http://dx.doi.org/10.1093/bioinformatics/btr597.

Sun, C. T., Zhang, L. B., Zhou, C. G., & Liu, X. H. (2010). An iris recognition algorithm based on weighted KNN and weighted majority voting. *Journal of Chinese Computer Systems*, *31*(9), 1846–1849.

Taylor, P., Rubin, D. B., & Rubin, D. B. (2012). Multiple imputation after 18 + years. *Publications of the American Statistical Association*, *91*(2013), 37–41.

Tseng, S. M., Wang, K. H., & Lee, C. I. (2003). A pre-processing method to deal with missing values by integrating clustering and regression techniques. *Applied Artificial Intelligence*, *17*(5–6), 535–544. http://dx.doi.org/10.1080/713827170.

van Buuren, S., & Oudshoorn, K. (1999). *Flexible multivariate imputation by MICE* (pp. 1–20).

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In *Lazy learning, Vol. 11* (pp. 273–314). http://dx.doi.org/10.1007/978-94-017-2053-3_11.

Xu, A., & Raginsky, M. (2017). Information-theoretic analysis of generalization capability of learning algorithms. Retrieved from http://arxiv.org/abs/1705.07809.

Yang, K., Li, J., & Wang, C. (2006). Missing values estimation in microarray data with partial least squares regression. In *Lecture notes in computer science, Vol. 3992* (pp. 662–669). http://dx.doi.org/10.1007/11758525_90.

Zhang, X. (2018). Predicting short-term electricity demand by combining the advantages of ARMA and XGBoost in fog computing environment. *Wireless Communications and Mobile Computing*, *2018*, Article 5018053. http://dx.doi.org/10.1155/2018/5018053.

Zhang, Y., Cao, G., Wang, B., & Li, S. (2019). Nearest neighbor selection for iteratively kNN imputation. *Pattern Recognition*, *85*, 13–25. http://dx.doi.org/10.1016/j.patcog.2018.08.003.

Zhang, D., Wang, J., & Zhao, X. (2015). Estimating the uncertainty of average F1 scores. In *International conference on the theory of information retrieval* (pp. 317–320). http://dx.doi.org/10.1145/2808194.2809488.

Zhao, P., & Tang, X. (2016). Imputation based statistical inference for partially linear quantile regression models with missing responses. *Metrika*, *79*(8), 991–1009. http://dx.doi.org/10.1007/s00184-016-0586-8.