FUSAIN: Combining Functional Dependencies and Clustering for Missing Values Imputation

Huaiguang Wu, Shuaichao Li, Wenjun Shi, Shaoqing Du

Abstract—**Missing data is a common problem faced with real-world datasets. A large number of missing data will greatly affect the quality of the data and cause deviations in the results of data analysis. Therefore, missing values imputation (MVI) is a critical data processing process. Most imputation methods model the distribution of observed data to approximate the missing values. Such an approach usually models a single distribution for the entire dataset, which ignores the dependencies between data. In this paper, we propose a novel hybrid imputation algorithm, called combining Functional dependencies and clUstering for miSsing vAlues ImputatioN (FUSAIN), which combines Functional Dependencies (FDs), K Nearest Neighbor (KNN), and Affinity Propagation (AP) clustering. This proposed algorithm not only considers the distribution of data but also uses the data dependency relationship represented by FDs to impute missing values. From the experimental results, the imputation performance of the proposed algorithm achieves superior performance compared to common and popular imputation algorithms.**

Index Terms—**Missing value imputation, Affinity propagation clustering, Functional dependencies, K nearest neighbor.**

I. Introduction

THE growing use of machine learning (ML) and deep learning (DL) techniques demand more and more data. However, Missing values are common in real-world datasets, such as medical and financial records, and can cause bias and degrade the quality of supervised learning and classification systems [1], [2]. Statistics and machine learning algorithms typically require complete datasets to accomplish classification or prediction tasks [3], [4], [5]. It underlines the importance of managing missing data correctly. Equipment failure, human mistakes, data corruption, and other factors can all lead to missing values. The three forms of missing data problems are determined by the relationship between the missing and observed values: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [6], [7]. MCAR occurs when the missingness is completely independent of all other variables in the data [8]. Missingness in MAR is only relevant for observable variables. MNAR exists when missingness is determined by both the observed and missing variables [9]. MAR is a more common type of deletion than MCAR and MNAR.

Missing data is generally dealt with by: deletion and imputation. [10]. The first method of processing is used when only a small amount of data is missing. When there are a large number of missing values in the data, deleting them completely would result in a large loss of information, so MVI may be a better option [7]. MVI is a technique for estimating missing values from observable data. To replace missing values for a variable, traditional imputation methods typically use statistical estimates, such as mean [7], [11], [12] and linear regression (LR) [13], [14], [15] imputation. These methods simply infer the missing data from the distribution of data for a single variable, resulting in an underestimation of the variance of the predicted values and poor performance [9]. Advanced methods, such as expectation maximisation (EM), assume a multivariate normal distribution and estimate missing values based on the overall distribution of the data set [16]. Imputation methods based on statistics are simple to construct and interpolate well for data sets with certain distributions, but they do not capture deep correlation information between variables.

A number of ML-based imputation techniques have been proposed since the beginning of machine learning. In 2003, Thompson proposed a method for imputing missing data using a neural network algorithm [17]. In 2004, Jonsson and Wohlin proposed a KNN-based technique for MVI, also known as K-nearest neighbors imputation (KNNI) [18]. In 2005, Hai Hong et al. proposed an imputation algorithm for missing values based on support vector machine (SVM) regression [19]. In 2009, Ling Wang et al. improved the KNNI algorithm and proposed a weighted KNNI, also referred to as weighted K Nearest Neighbor imputation (WKNNI) [20]. In 2014, Burgette and Reiter proposed a non-parametric approach to multiple imputations through chained equations by using a serial regression tree as a conditional model [21]. Yun He and Dechang Pi proposed the RKNN induction algorithm, an improved KNN method for iterative estimation of microarray deletion values, in 2015 [22]. The RKNN induction algorithm iterates over the input deletion data using reduced association as a similarity metric and extends the set of nearest neighbor candidate genes using the input genes. In 2016, Razavi-Far et al. proposed a fuzzy neighborhood density-based clustering technique for missing value attribution [23]. In 2017, Soni and Sharma jointly proposed a fuzzy clustering method based on statistical information particles and applied this method to MVI [24]. In 2020, Raja and Sasirekha proposed a new method for MVI based on fuzzy C-Means rough parameters, using a mixture of fuzzy and rough sets to deal with missing values [25]. In 2021, Saqib Ejaz Awan et al. proposed a new approach to estimating

Manuscript received September 28, 2021; revised March 17, 2022. This work is supported by National Natural Science Foundation of China(61672470) and Major Public Welfare Projects in Henan Province, China(201300210200).

Huaiguang Wu is a professor of the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan Province, 450066, China (e-mail: hgawu@126.com).

Shuaichao Li is a postgraduate student of Zhengzhou University of Light Industry, Zhengzhou, Henan Province, 450066, China (e-mail: shuaichao_li@163.com).

Wenjun Shi is a lecturer of the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan Province, 450066, China (e-mail: swjij@sina.com).

Shaoqing Du is a postgraduate student of Zhengzhou University of Light Industry, Zhengzhou, Henan Province, 450066, China (e-mail: shaoqing_duu@163.com).

missing data by adapting the popular conditional generative adversarial network based on its specific class features [26].

Real-world data contains a wide variety of data distributions, and a single imputation method only performs well on data that satisfy certain specific distributions. Based on this shortcoming, researchers have developed an imputation method based on multiple distributions. The multiple imputation algorithm continues to increase in efficacy by progressively reducing bias and reducing the requirement for prior knowledge of the distribution. In 2004, Dan Li et al. proposed a more sophisticated method for missing value imputation, which combines fuzzy acuity and KNNI [27]. In 2005, Wei Qiao et al. implemented a missing data estimator that uses a combination of particle swarm optimization and neural networks to predict missing values [28]. In 2006, Abdella et al. proposed a combination of neural networks and genetic algorithms to estimate missing data in databases [29]. In 2012, Gajawada and Toshniwal combined clustering theory with KNN and proposed a new missing value imputation algorithm [30]. In 2013, Aydilek and Arslan et al. proposed a fuzzy c-means clustering hybrid imputation method that combines support vector regression and genetic algorithms [31]. In 2014, Jing Tian et al. proposed a hybrid imputation method named Multiple Imputation using Gray-system-theory and Entropy based on Clustering (MIGEC)[32]. In 2015, induced by the thought of collaborative training, Huihui et al. proposed a novel hybrid imputation method, called recursive mutual imputation (RMI)[33]. In 2016, Geaur Rahman et al. proposed a novel technique called a fuzzy expectation-maximization and fuzzy clustering-based missing value imputation framework for data pre-processing (FEMI) [34]. In 2018, Lin Qiao and Ran Ran et al. proposed an effective imputation method based on iterative KNN and extreme gradient boosting (XGBoost) method. The method first determines the priority of attributes, and then iteratively interpolates missing values [35]. In 2019, Aikaterini Karanikola et al. proposed a novel MVI algorithm based on a widely used imputation method and decision tree theory [36]. In 2020, Nikfalazar et al. proposed a new imputation method called DIFC by integrating the merits of decision trees and fuzzy clustering into an iterative learning approach [37]. In 2020, Raja et al. proposed a Novel Fuzzy C-Means Rough Parameter-based missing value imputation method that uses the hybridization of the fuzzy and rough set to deal with missing values [38].

Although the hybrid multiple imputation techniques achieve good interpolation results, there is still room for improvement. In 2003, Dardzinska et al. used relaxed FDs (rules extracted from the dataset) and thresholds for discovered values of attributes to impute the final dataset called [39]. This imputation algorithm takes advantage of attribute dependencies to a great extent but ignores the data's overall distribution information. Furthermore, this approach is limited to discrete data. To make full use of both data distribution information and attribute-related information, we present a hybrid imputation technique that combines ML and FDs. The hybrid algorithm, as opposed to the single imputation approach, takes greater use of the link between the variables in the data and may be applied to a wider range of datasets with different data distributions. Below is a list of some of our most significant contributions:

1) The AP clustering algorithm is used for MVI. The number of clusters does not need to be defined ahead of time with this approach, and the clustering results are more stable than with other clustering algorithms.
2) The similarity measurement method of discrete and mixed data is added to the traditional AP clustering algorithm. The improved AP clustering algorithm can be applied to mixed types of data.
3) The FDs-based imputation algorithm is proposed. The relationships between attributes are used in this technique to identify an imputation value that is near to or even equal to the missing value, which increases the accuracy of missing value imputation even further.
4) The proposed algorithm is compared to two commonly used imputation algorithms, and a large number of tests are conducted to demonstrate that the presented algorithm performs well in MVI.

The rest of the paper is organized as follows. In Section II, we overview the FUSAIN framework and then introduce the concrete implementation of the algorithms in detail. We report the experimental results in Section III, and finally, conclude the paper in Section IV.

II. Proposed method

In this work, we aim to apply both data distribution and attribute dependencies to MVI. To achieve this goal, we propose a novel hybrid imputation algorithm, namely FUSAIN. The FUSAIN algorithm combines FDs [40], improved AP clustering, and the KNN algorithm. The FUSAIN algorithm's general architecture is shown in Fig 1. The structure of this algorithm is briefly discussed below.

Firstly, the dataset is divided into complete datasets and incomplete datasets with missing values based on whether the tuple has missing values or not. An FD discovery algorithm is used to discover the FDs between attributes in the complete dataset. The obtained FDs are then used to identify the complete tuples from the complete dataset that match the currently missing tuples. If no tuples are matched, the complete dataset is clustered using the improved AP clustering algorithm, which produces many clusters. Using a mixed similarity measure, the clusters that are closest to the missing tuples are identified and the nearest-neighbor complete tuples are determined from them. Finally, the complete tuples are used to impute the missing values. The implementation of the hybrid algorithm is detailed below.

A. *Improved Affinity Propagation Clustering Algorithm*

The AP clustering algorithm was proposed by Frey and Dueck in 2007 [41]. It is particularly suitable for fast clustering of high-dimensional, mixed data, and offers substantial improvements in terms of clustering performance and efficiency compared to traditional clustering algorithms. It is currently used mainly in the field of image processing, with some early applications in semi-supervised clustering. The AP clustering algorithm does not require the number of clusters to be specified in advance, and the clustering results are more stable than other clustering algorithms [42]. Based on these advantages, we apply the AP clustering algorithm to MVI. The core idea of the algorithm is to use all data points as potential clustering centers. During the iterative

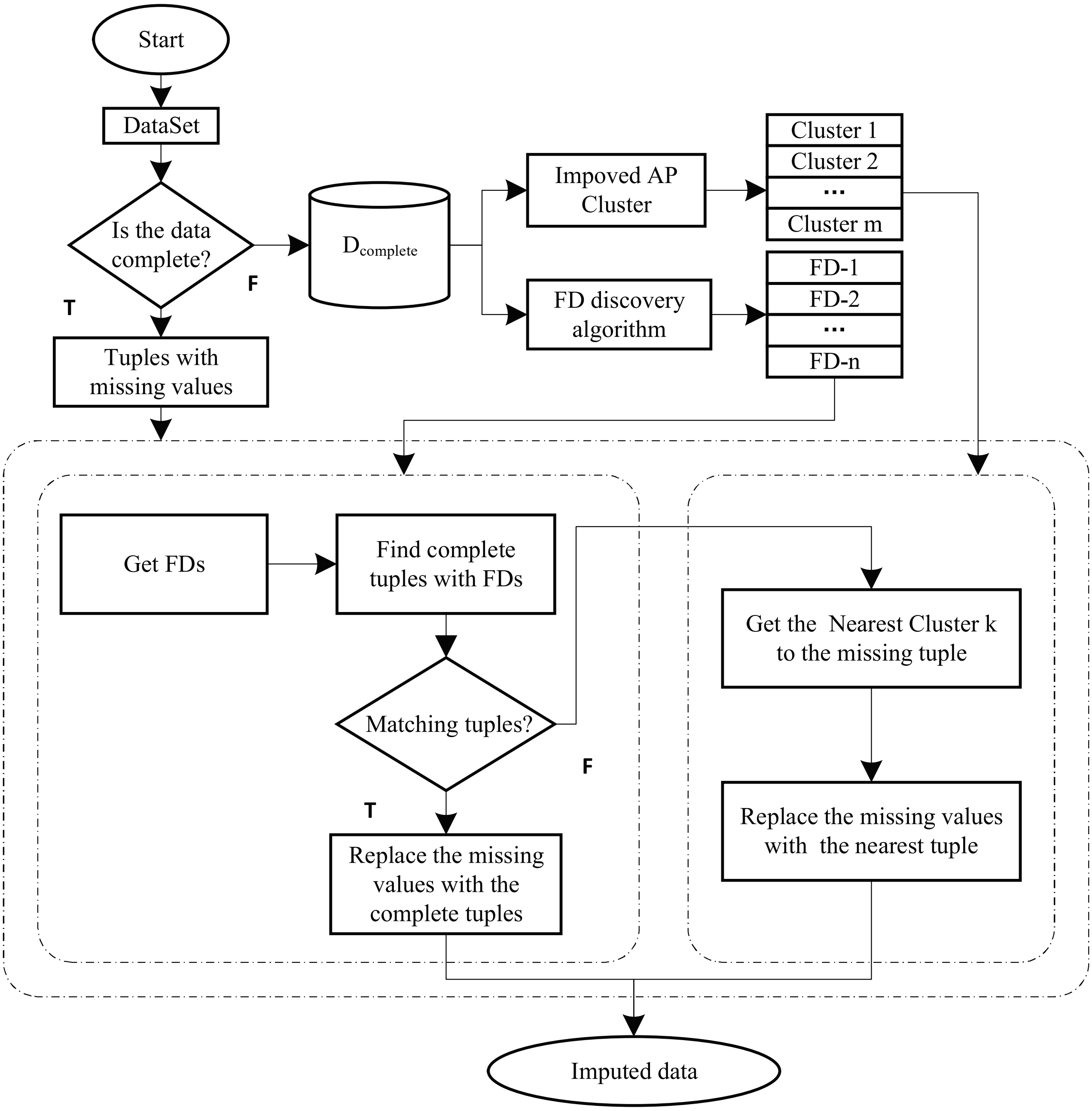

Start
DataSet
Is the data complete?
T
F
Dcomplete
Impoved AP Cluster
Cluster 1
Cluster 2
...
Cluster m
FD discovery algorithm
FD-1
FD-2
...
FD-n
Tuples with missing values
Get FDs
Find complete tuples with FDs
Matching tuples?
F
T
Replace the missing values with the complete tuples
Get the Nearest Cluster k to the missing tuple
Replace the missing values with the nearest tuple
Imputed data

Fig. 1. The overall architecture of the FUSAIN algorithm

process, representative and appropriate information between data points is constantly updated to find the final cluster centers, as well as the location and number of cluster centers. Finally, the sum of the similarity of all data points to the nearest cluster center is maximized [41].

As the traditional AP clustering algorithm uses Euclidean distance to calculate similarity, it does not apply to mixed data. Similarity measures for discrete and mixed data have been added to the traditional AP clustering algorithm so that the improved AP clustering algorithm can be applied to mixed types of data. The improved AP clustering algorithm is described in Algorithm 1.

Mixed data includes continuous variables and discrete variables. The number of continuous variables and the number of discrete variables are represented C$_n$ and D$_n$ respectively. The similarity between continuous variables is given by Equation 1:

$$S_C(R_i, R_j) = \sqrt{\sum_{m=1}^{C_n} (R_i^m - R_j^m)^2} \tag{1}$$

Where R$_i^m$ represents the attribute value of the record R$_i$ on the m-th continuous attribute, and R$_j^m$ represents the attribute value of the record R$_j$ on the m-th continuous attribute. It can be seen that the similarity between continuous variables is calculated using the Euclidean distance.

The similarity measure between discrete variables is given by Equation 2:

Algorithm 1 Improved AP Clustering Algorithm

```
Require: complete data set $D_c$; number of continuous attributes $conNum$; number of discrete attributes $disNum$; maximum iteration number $maxIterNum$; damping coefficient $\lambda$;
Ensure: The number of clusters after clustering and the data contained in each cluster;
$iterNum = 0$;
$r(i,j) = 0$;
$a(i,j) = 0$;
$r(i,j)' = 0$;
$a(i,j)' = 0$;
$computeHybridFieldsDistance()$;
while $iterNum \leq maxIterNum$ do
    $r(i,j) = computeResponsibility()$;
    $a(i,j) = computeAvailability()$;
    $r(r,j) + a(i,j) = computeRASum()$;
    if $r(i,j) + a(i,j) == r(i,j)' + a(i,j)'$ then $break$
    end if
    $r(i,j) = updateResponsibility()$;
    $a(i,j) = updateAvailability()$;
end while
```

$$S_D(R_i, R_j) = \sum_{m=1}^{D_n} s(R_i^m, R_j^m) \tag{2}$$

In Equation 2, s(R$_i^m$, R$_j^m$) represents the similarity of the m-th discrete attribute between R$_i^m$ and R$_j^m$. The s(R$_i^m$, R$_j^m$) is given by Equation 3:

$$s(R_i^m, R_j^m) = \begin{cases} 0 & R_i^m = R_j^m \\ 1 & R_i^m \neq R_j^m \end{cases} \tag{3}$$

The meaning expressed by Equation 3 is that when records R$_i$ and R$_j$ have the same value on the corresponding m-th discrete attribute, the two records are considered to be similar in the dimension of the attribute. Otherwise, record the similarity value of R$_i$ and R$_j$ in the attribute dimension to 1.

The total similarity of the two records of R$_i$ and R$_j$ is given by Equation 4:

$$s(R_i^m, R_j^m) = \alpha \times S_C(R_i, R_j) + \beta S_D(R_i, R_j) \tag{4}$$

$$\alpha = \frac{C_n}{C_n + D_n} \tag{5}$$

$$\beta = \frac{D_n}{C_n + D_n} \tag{6}$$

Where α represents the ratio of the number of continuous attributes to the number of all attributes, and β represents the ratio of the number of discrete attributes to the number of all attributes.

Clustering operations are performed on the complete data set D_C. In the cluster initialization phase, the complete data set, the maximum number of iterations $maxIterNum$, and the damping factor λ, the number of continuous and discrete attributes will be entered into the algorithm as parameters. In the initial phase of the algorithm, the similarity matrix S is calculated using the method of computing similarity. After that, the elements $s(i,i)$ on the diagonal of the similarity matrix S will be formed into a new matrix P. The element $P(i)$ in the matrix P represents the reference degree of the AP clustering algorithm, that is, the reference degree of each data point itself as a cluster centre. The larger the element $P(i)$ value, the more likely the data point i is to become the cluster center.

The alternating process of the Responsibility matrix R and the Availability matrix A is the core of the algorithm. The elements in the matrix R are denoted as $r(i,k)$, which represents the degree to which the data point k is suitable as the cluster center of the data point i. The elements in matrix A are represented as $a(i,k)$, which represents the suitability of data point i to select data point k as its cluster center.

When the number of iterations is less than $maxIterNum$, representative information $r(i,j)$ and the suitable information $a(i,j)$ between data points are given by Equation 7 and 8.

$$r(i,k) = s(i,k) - max[a(i,\dot{k}) + s(i,k')], k' \neq k \tag{7}$$

$$a(i,k) = \begin{cases} min(0, r(k,k) + \sum_{i' \neq i} max(0, r(i',k))) & i \neq k \\ \sum_{i' \neq i} max(0, r(i',k)) & i = k \end{cases} \tag{8}$$

Then the sum of $r(i,j)$ and $a(i,j)$ is given by Equation 9. Finally, the matrix R and A are updated alternately according to Equation 10, 11 and damping coefficient λ.

$$\begin{aligned} r(i,k) + a(i,k) = {} & s(i,k) + a(i,k) \\ & - max_{k' \neq k, k' \neq i}[a(i,k') + s(i,k')] \end{aligned} \tag{9}$$

$$r_{t+1}(i,k) = \lambda \times r_t(i,k) + (1-\lambda) \times r_{t+1}(i,k) \tag{10}$$

$$a_{t+1}(i,k) = \lambda \times a_t(i,k) + (1-\lambda) \times a_{t+1}(i,k) \tag{11}$$

The final cluster center of each cluster is achieved when the sum of the two forms of information between the data points is maximized and the iteration phase of the algorithm is complete. In addition, the algorithm will stop iterating if the number of iterations exceeds the $maxIterNum$ value provided.

B. Missing value imputation algorithm based on FD

FD represents the attribute association relationship in a given relationship R [39]. An FD X→Y, over relation R, where X, Y ⊂ R, states that if any two tuples in an instance of R have equal X-values, then their Y-values should also be identical. Such attribute dependencies can impute the missing values by matching a complete tuple to the uncomplete tuple. The FD-Based Missing Values Imputation Algorithm is described in Algorithm 2.

The algorithm takes three parameters: the complete dataset $D_{complete}$, the missing tuples $T_{missing}$, and the missing attribute $Attr$. First, the current complete dataset is processed using the FDs discovery algorithm, and then all the obtained

Algorithm 2 FD-Based Missing Values Imputation Algorithm

```
Require: Complete data set $D_{complete}$; a tuple $T_{missing}$ with missing values; a missing attribute $Attr$;
Ensure: $BoolValue$ indicating whether missing value imputation is completed.;
$FDSet = HYFD(D_{complete})$;
$BoolValue = False$;
$FDList = newList$;
for $fd \in FDSet$ do
    if $fd$ satisfies $Attr \in RHSoffd$ then
        $FDList.append(fd)$;
    end if
end for
Sort the $FD$ in $FDList$ according to the size of $LHS$ of the $FD$ in ascending order;
for $fd \in FDSet$ do
    if $fd$ can match to the complete tuple corresponding to $T_missing$ then $flag = MissingValueImputationByFD(fd, D_complete)$;
        if $flag == True$ then
            $boolValue = True$;
            $break$;
        end if
        $break$;
    end if
end for
```

FDs are stored in the set $FDSet$. Here, the HYFD algorithm is used to find FDs. HYFD is a hybrid FD discovery algorithm that is faster than state-of-the-art algorithms and can handle larger datasets [43], [44].

Then, iterate through all the FDs in the set $FDSet$ and store all the FDs that satisfy the $Attr \in EHS$ condition in the list $FDList$. The RHS represents the set of attributes to the right of the FDs. The LHS represents the set of attributes to the left of the FDs. Of the FDs contained in the list $FDList$, the fewer the number of attributes contained in LHS, the stronger the correlation between the left and right attribute sets of the FD. Therefore, to improve the accuracy of missing value imputation, the FD with a small number of attributes in LHS is preferred to impute missing values. Sort the FDs of $FDList$ in ascending order according to the number of attributes in LHS. Then, take the first FD in $FDList$ and determine if that FD can find a complete tuple $T_{complete}$ that matches the missing tuple $T_{missing}$ in the complete dataset $D_{complete}$. If it can be found, fill the missing tuple $T_{missing}$ with the value of the $Attr$ attribute in the complete tuple $T_{complete}$. The algorithm then returns True, indicating that the imputation of the value of the $Attr$ attribute is complete. If none are found, then subsequent FDs in the $FDList$ are traversed in turn. if all FDs in the $FDList$ cannot match the complete tuple $T_{complete}$, the algorithm returns False, indicating that the attribution of the $Attr$ attribute value is not complete.

C. FUSAIN Algorithm

The FUSAIN algorithm is divided into two parts. In the first part, the whole dataset is clustered and the clustering index is calculated. In the second part, the process of the imputation of missing values is completed using FDs and KNN algorithms. The FUSAIN Algorithm is described in Algorithm 3. The flow of the FUSAIN algorithm is described in detail in the following sections.

Algorithm 3 FUSAIN Algorithm

```
Require: The dataset $D_{input}$ for MVI;
Ensure: Completed dataset $D_{output}$;
$D_{input} = ReadFile(FileName)$;
while $GetDataNum(D_{missing}) > 0$ do
    $D_{complete} = GetComPleteData(D_{input})$;
    $D_{missing} = GetMissingData(D_{input})$;
    $FDSet = HYFD(D_{complete})$;
    $T_{missing} = GetFirstTuple(D_{missing})$;
    $MissingAttrList = GetMissingAttr(T_{missing})$;
    for $attr \in MissingAttrList$ do
        $Flag = False$;
        if there is an $FD$ that satisfies $attr \in RHS$ of FD then
            $PartFDs = GetSatisfiedFD(FDSet)$;
            $SortFD(PartFDs)$;
            for $fd \in PartFDs$ do
                if $fd$ can match to the complete tuple corresponding to $T_{missing}$ then
                    $Flag = True$;
                    $MVIByFD(fd, attr)$;
                    break;
                end if
            end for
        end if
        if $Flag == False$ then
            $ClusterCenters = ImpovedAPCluster(D_{complete})$;
            $Cluster = GetNearestNeighborCluster(T_{missing})$;
            $KNNImputation(T_{missing}, attr)$;
        end if
    end for
end while
```

The algorithm first requires an incomplete dataset D_{input} from the file. The dataset D_{input} is divided into a complete data subset $D_{complete}$ and an incomplete data subset $D_{missing}$, depending on whether the tuple contains missing values. Then, the complete subset of data $D_{complete}$ is processed using the HYFD algorithm to obtain a set $FDSet$ containing all FDs. A tuple $T_{missing}$ containing missing values is extracted from the incomplete dataset $D_{missing}$. Then, the tuple $T_{missing}$ is processed to obtain all missing attributes and stored in the $MissingAttrList$. The missing attributes in $MissingAttrList$ will be estimated according to the following procedure.

First, select one of the missing attributes $Attr$ from the $MissingAttrList$ as the attribute that currently needs to be imputed. Iterate through all the FDs in the set FDSet and store all the FDs that satisfy the condition $Attr \in RHS$ of the FD in the list $PartFDs$. Sort the FDs in the $PartFDs$ in ascending order according to the number of attributes in the LHS. Then iterate through the FDs in $PartFDs$ in turn. Find the complete tuple matching the missing tuple $T_{missing}$ from the complete dataset D according to the FDs.

If the tuple $T_{complete}$ can be found, the value corresponding to the missing attribute $Attr$ in $T_{complete}$ is directly used to impute $T_{missing}$, and then continue to impute the next missing attribute.

If the tuple $T_{complete}$ does not exist, the KNN algorithm is used to complete the imputation of missing values. First, the complete array $D_{complete}$ is clustered using the improved AP clustering algorithm and the index of the cluster centers is obtained. According to Equation 4, the similarity of the tuple $T_{missing}$ to all clusters is calculated, and then the Nearest Neighbor Cluster(NNC) of the cluster center that is most similar to the tuple $T_{missing}$ is obtained. The continuous attribute value corresponding to the whole data subset in the NNC cluster is used to complete the missing value imputation operation for continuous attribute missing values. This is done by calculating the similarity of the tuple $T_{missing}$ to each complete tuple in the NNC, denoted as $s(inc, c_p)$, and then calculating the corresponding missing values according to Equation 5.

$$x = \frac{\sum_{p=1}^{q} \frac{1}{s(inc,c_p)^2} \times V_{cp\_corr}}{\sum_{p=1}^{q} \frac{1}{s(inc,c_p)^2}} \tag{12}$$

In Equation 12, the value of the continuous attribute corresponding to the complete tuple in the NNC is represented by V_{cp_corr}. The number of complete tuples in NNC is denoted by q. It avoids artificially setting the value of K and also reduces the effect of less similar complete tuples on imputation results by using $frac[1][s(inc, c_p) * 2]$.

For discrete attributes, the algorithm uses the complete tuple in the NNC for statistical analysis, and then uses the statistics to estimate the missing values.

III. Experimental results

In order to illustrate the performance of the FUSAIN algorithm proposed, a series of experiments will be carried out in this section and the experimental results will be analyzed. The main contents of the experiment are described below.

A. Datasets

This experiment was conducted on five datasets from University of California (UCI) Repository of Machine Learning Databases. The five datasets are Energy Efficiency datasets, Yeast datasets and Banknote authentication datasets. The information of the datasets are described in Table I.

Since the initial datasets do not contain missing values, to conduct experiments, the MCAR method is used to deal with the initial datasets and got missing datasets with missing rates of 10%, 15%, 20%, 25%, and 30% respectively.

TABLE I
THE INFORMATION OF THE DATASETS FOR EXPERIMENTATION

Dataset Name	No.of records	No.of attributes
Energy Efficiency	768	10
Yeast	1484	9
Banknote authentication	1372	5

B. Evaluation measures

In order to evaluate the missing value imputation performance of the algorithm, it is necessary to evaluate the experimental results using the appropriate evaluation metrics. In this paper, the Root Mean Square Error (RMSE) and Mis-Classification Rate (MCR) are used to measure errors for continuous and discrete attributes respectively. For continuous attributes, the RMSE is relativised in this paper to eliminate the effect of different attribute dimensions. The RMSE is given by Equation 13:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{C} (\frac{x_{filled} - x_{origin}}{x_{origin}})^2}{C}} \tag{13}$$

where x_{filled} is the estimated value corresponding to the missing attribute in the incomplete tuple, x_{filled} is the original value of the missing attribute in the incomplete tuple, and C is the number of consecutive attributes missing in the incomplete tuple.

For discrete variables, the MCR is given by Equation 14:

$$MCR = \frac{\sum_{j=1}^{D} \delta(x_{filled} - x_{origin})}{D} \tag{14}$$

$$\delta(x_{filled} - x_{origin}) = \begin{cases} 1 & x_{filled} \neq x_{origin} \\ 0 & x_{filled} = x_{origin} \end{cases} \tag{15}$$

Where x_{filled} is the estimated value corresponding to the discrete attribute in the incomplete tuple, x_{origin} is the original value of the corresponding discrete attribute, and D is the number of missing discrete attribute values in the incomplete tuple.

C. Experimental results

To prove the performance of the FUSAIN algorithm proposed in this paper, two comparison imputation algorithms are added in the experiment. The following are the two contrast imputation algorithms:

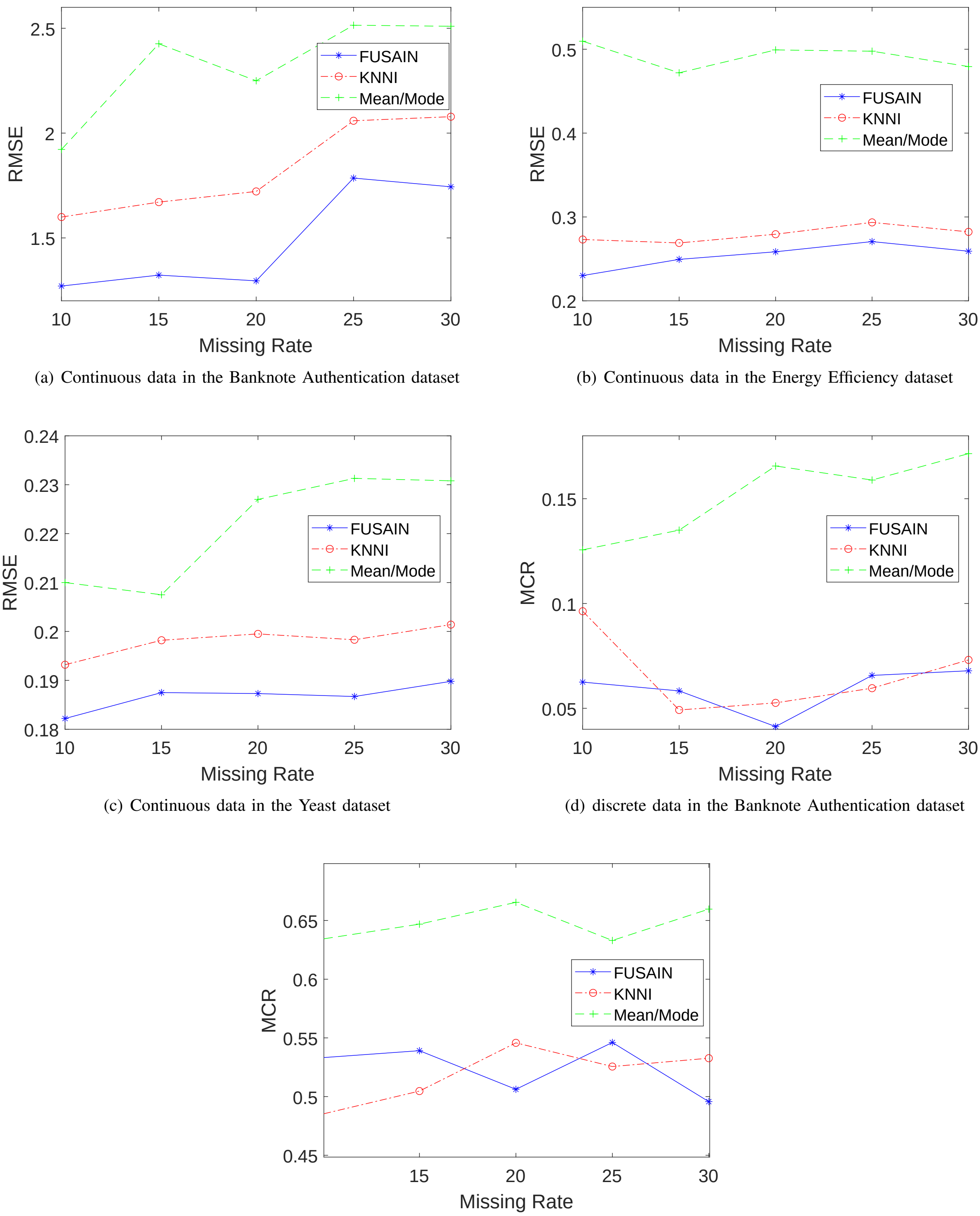
1) Mean/Mode

Simple, easy-to-understand, and statistically-based imputation algorithm. Its operation is straightforward: for continuous missing values, the average of the missing attributes is used to replace the missing values; for discrete missing values, the most frequent value is used to replace the missing values.

2) KNN Imputation (k = 3)

The KNNI method is a machine learning-based imputation algorithm that is widely used and effective. To impute missing data, use the KNN method with k equal to 3. The attribute value from the three complete tuples closest to the incomplete tuple is imputed for each missing value.

At the beginning of the experiment, the random missing algorithm was applied to original complete datasets, and incomplete datasets with missing rates of 10%, 15%, 20%, 25%, and 30% were obtained. The missing value imputation algorithm is then applied to datasets with different missing rates, and the performance of each approach is assessed using the evaluation metrics. As the missing treatments are randomized, the 20 experimental groups are performed for



(a) Continuous data in the Banknote Authentication dataset

(b) Continuous data in the Energy Efficiency dataset

(c) Continuous data in the Yeast dataset

(d) discrete data in the Banknote Authentication dataset

(e) discrete data in the Yeast dataset

Fig. 2. Performance comparison of the imputation algorithm on the continuous and discrete attribute

each missing rate in the dataset and the final assessment is the average of the results of the 20 experimental groups.

The imputation results of all approaches for continuous data are displayed in Fig 2(a) through Fig 2(c). When comparing the FUSAIN method to the KNNI and Mean/Mode algorithms, it can be seen that the FUSAIN algorithm has the best imputation performance for continuous attributes. As the missing rate rises, the performance of all techniques worsens, especially Mean/Mode. The imputation performance of Mean/Model in Fig 2(b) and Fig 2(c) is much inferior to other approaches due to the substantial variation of the Energy Efficiency and Yeast datasets.

Figures 2(d) and 2(e) illustrate the imputation results for all methods for discrete data. The MCR values for the FUSAIN algorithm are lower overall than the other two methods, but it is not significantly better than the KNNI algorithm. Compared to the KNNI method, the FUSAIN algorithm performs better when the missing rate is between 20% and 30%. As the missing rate rises, the imputation performance of all methods fluctuates up and down due to the uneven distribution of discrete data.

Table II and Table III respectively indicate the average performance improvement of the FUSAIN algorithm for discrete and continuous data. On real datasets with different missing rates, the FUSAIN algorithm showed good imputation performance, with an average imputation performance improvement of 11.39% and 33.37% compared to the KNNI and Mean/Mode algorithms, respectively. For the imputation of continuous data, the imputation performance fluctuates up and down, which may be caused by the uneven distribution of discrete data. There is a good improvement in the imputation performance of the FUSAIN algorithm when the missing rate is 10%, 20%, and 30%. The attribution performance of the FUSAIN algorithm decreases when the missing rate is 15% and 25%. Overall, the FUSAIN algorithm improved the imputation performance by 2.86% and 39.63% compared to the KNNI and Mean/Mode methods, respectively. From the experimental results, it can be seen that the FUSAIN algorithm outperforms the KNNI and Mean/Mode algorithms in terms of overall subsumption performance.

TABLE II
THE AVERAGE PERFORMANCE IMPROVEMENT OF THE FUSAIN ALGORITHM FOR CONTINUOUS ATTRIBUTES

Missing Rate	KNNI	Mean/Mode
10%	14.01%	34.00%
15%	11.18%	34.09%
20%	12.79%	36.05%
25%	8.98%	31.30%
30%	10.01%	31.41%
avg	11.39%	33.37%

IV. CONCLUSION

In the paper, a novel missing values imputation algorithm is proposed, which combines FDs, improved AP clustering, and KNN algorithm, namely FUSAIN. We compare the FUSAIN algorithm with two other efficient existing algorithms, KNNI and Mean/Mode. The experiment is conducted on three real datasets from the UCI. Using the evaluation criteria of RMSE and MCR, the effectiveness of the proposed algorithm is judged in terms of both continuous and discrete attributes imputation. From the experimental results, the proposed algorithm achieves good imputation performance.

TABLE III
THE AVERAGE PERFORMANCE IMPROVEMENT OF THE FUSAIN ALGORITHM FOR DISCRETE ATTRIBUTES

Missing Rate	KNNI	Mean/Mode
10%	12.60%	33.08%
15%	-12.65%	36.74%
20%	14.36%	49.49%
25%	-7.06%	36.19%
30%	7.03%	42.64%
avg	2.86%	39.63%

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the University of California (UCI) Repository of Machine Learning Databases at http://archive.ics.uci.edu/ml/index.php, reference number three.

REFERENCES

[1] Yongsong Qin, Shichao Zhang, Xiaofeng Zhu, Jilian Zhang, Chengqi Zhang, "Semi-parametric Optimization for Missing Data Imputation," *Applied Intelligence*, vol. 27, no. 1, pp. 79-88, 2007.

[2] Chengqi Zhang, Xiaofeng Zhu, Jilian Zhang, Yongsong Qin, Shichao Zhang, "GBKII: An Imputation Method for Missing Values," *PAKDD 2007: Advances in Knowledge Discovery and Data Mining*, vol. 4426, pp. 1080-1087, 2007.

[3] Zoila Ruiz-Chavez, Jaime Salvador-Meneses, Jose Garcia-Rodriguez, "Machine learning methods based preprocessing to improve categorical data classification," *International Conference on Intelligent Data Engineering and Automated Learning*, vol. 11314, pp. 297-304, 2018.

[4] Catia M. Salgado, Carlos Azevedo, Hugo Proenca, Susana M.Vieira, "Missing data," *Secondary Analysis of Electronic Health Records*, Springer, Cham, pp. 143-162, 2016.

[5] Marek Smieja, Lukasz Struski, Jacek Tabor, Bartosz Zielin ski, Przemyslaw Spurek, "Processing of missing data by neural networks," *Advances in Neural Information Processing Systems*, pp. 2719-2729, 2018.

[6] Shichao Zhang, Z. Qin, C. X. Ling, S. Sheng, "Missing is useful: missing values in cost-sensitive decision trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1689-1693, 2005.

[7] R. J. A. Little, D. B. Rubin, "Statistical Analysis with Missing Data," *Wiley*, New York, 1986.

[8] F. V. Nelwamondo, S. Mohamed, T. Marwala, "Missing data: A comparison of neural network and expectation maximization techniques," *Current Science Association*, vol. 93, no. 11, pp. 1514-1521, 2007.

[9] Saqib Ejaz Awan, Mohammed Bennamoun, Ferdous Sohel, Frank Sanfilippo, Girish Dwivedi, "Imputation of Missing Data with Class Imbalance using Conditional Generative Adversarial Networks," *Neurocomputing*, vol. 453, pp. 164-171, 2020.

[10] Alireza Farhangfar; Lukasz A. Kurgan, Witold Pedrycz, "A Novel Framework for Imputation of Missing Values in Databases," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 5, pp. 692-709, 2007.

[11] R. J. A. Little, D. B. Rubin, "The Analysisy of Social Science Data with Missing Values," *Sociological Methods and Research*, vol. 18, pp. 292-326, 1990.

[12] YJ Jin, "Imputation adjustment method for missing data," *Application of Statistics and Management*, vol. 20, no. 5, pp. 47-53, 2001.

[13] Li Xuying, "Imputation method for regional missing data using spatial auto-regression model," *Application of Statistics and Management*, vol. 24, no. 5, pp. 45-50, 2005.

[14] C. Abhishek, T. Cai, "Efficient and Adaptive Linear Regression in Semi-Supervised Settings," *Annals of Statistics*, vol. 46, no. 4, pp. 1541-1572, 2018.

[15] N. Karmitsa, S. Taheri, A. Bagirov, P. Makinen, "Missing Value Imputation via Clusterwise Linear Regression," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-1, 2020.

[16] Lin T. , Lee J. C. , Ho H. J. , "On fast supervised learning for normal mixture models with missing information," *Pattern Recognition*, vol. 39, no. 6, pp. 1177-1187, 2006.

[17] B. B. Thompson, R. J. Marks, M. A. El-Sharkawi, "On the contractive nature of autoencoders: Application to sensor restoration," *International Joint Conference on Neural Networks*, vol. 4, pp. 3011-3016, 2003.

[18] Jonsson Per, Wohlin Claes, "An Evaluation of K-nearest Neighbours Imputation Using Likert data," Proceedings-10th *International Symposium on Software Metrics*, pp. 108-118, 2004.

[19] Feng Honghai, Chen Guoshun, Yin ChengYang Bingru, Chen Yumei, "A SVM Regression Based Approach to Filling in Missing Values," *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 3683 pp. 581-587, 2005.

[20] W. Ling, F. Dong-Mei, "Estimation of missing values using a weighted knearest neighbors algorithm," *2009 International Conference on Environmental Science and Information Application Technology*, vol. 3, no. 2, pp. 660-663, 2009.

[21] Lane F. Burgette, Jerome P. Reiter, "Multiple imputation for missing data via sequential regression trees," *American Journal of Epidemiology*, vol. 172, no. 9, pp. 1070-1076, 2014.

[22] Yun He, De-chang Pi, "Improving KNN Method Based on Reduced Relational Grade for Microarray Missing Values Imputation," IAENG International Journal of Computer Science, vol. 43, no. 3, pp. 356-362, 2015.

[23] R. Razavi-Far, M. Saif, "Imputation of missing data using fuzzy neighborhood density-based clustering," *IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE), pp. 1834-1841, 2016.

[24] S. Soni, I. Sharma, "An imputation-based method for fuzzy clustering of incomplete data," *International Conference on Communication and Signal Processing* (ICCSP), pp. 0616-0621, 2017.

[25] P. S. Raja, K. Sasirekha, K. Thangavel, "A Novel Fuzzy Rough Clustering Parameter-based missing value imputation," *Neural Computing and Applications*, vol. 32, pp. 10033C10050, 2020.

[26] Saqib Ejaz Awan, Mohammed Bennamoun, Ferdous Sohel, Frank Sanfilippo, Girish Dwivedi, "Imputation of missing data with class imbalance using conditional generative adversarial networks," *Neurocomputing*, vol. 453, pp. 164-171, 2021.

[27] Dan Li, Jitender Deogun, William Spaulding, Bill Shuart, "Towards missing data imputation: A study of fuzzy K-means clustering method," *International Conference on Rough Sets and Current Trends in Computing*, vol. 3066, pp.573-579, 2004.

[28] Wei Qiao, Zhi Gao, R. G. Harley, "Continuous on-line identification of nonlinear plants in power systems with missing sensor measurements," *IEEE International Joint Conference on Neural Networks*, vol. 3, pp.1729-1734, 2005.

[29] M. I. Abdella, T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data in database," *Computing and Informatics*, vol. 24, pp. 1001-1013, 2006.

[30] S. Gajawada, D. Toshniwal, "Missing Value Imputation Method Based on Clusteringand Nearest Neighbours," *International Journal of Future Computer and Communication*, vol. 1, no. 2, pp. 206-208, 2012.

[31] Ibrahim Berkan Aydilek, Ahmet Arslan, "A hybrid method for imputation of missing values using optimized fuzzy cmeans with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25-35, 2013.

[32] Jing Tian, Bing Yu, Dan Yu, Shilong Ma, "Missing data analyses: a hybrid multiple imputation algorithm using gray system theory and entropy based on clustering," *Applied Intelligence*, vol. 40, pp. 376-388, 2014.

[33] Huihui Li, Changbo Zhao, Fengfeng Shao, , Guo-Zheng Li, Xiao Wang, "A hybrid imputation approach for microarray missing value estimation," *BMC Genomics*, vol. 16, pp. 1-11, 2015.

[34] Md. Geaur Rahman, Md. Zahidul Islam, "Missing value imputation using a fuzzy clustering-based EM approach," *Knowledge and Information Systems*, vol. 46, pp. 389-422, 2016.

[35] Lin Qiao, Ran Ran, He Wu, Qiaoni Zhou, Sai Liu, Yunfei Liu, "Imputation Method of Missing Values for Dissolved Gas Analysis Data Based on Iterative KNN and XGBoost," *International Conference on Algorithms, Computing and Artificial Intelligence*, vol. 11, pp. 1-7, 2018.

[36] Aikaterini Karanikola, Sotiris Kotsiantis, "A hybrid method for missing value imputation," *Association for Computing Machinery*, pp. 74-79, 2019.

[37] Sanaz Nikfalazar, Chung-Hsing Yeh, Susan Bedingfield, Hadi A. Khorshidi, "Missing data imputation using decision trees and fuzzy clustering with iterative learning," *Knowledge and Information Systems*, vol. 62, pp. 2419-37, 2020.

[38] P. S. Raja, K. Sasirekha, K. Thangavel, "A Novel Fuzzy Rough Clustering Parameter-based missing value imputation," *Neural Computing and Applications*, vol. 32, no. 14, pp. 10033C10050, 2014.

[39] Agnieszka Dardzinska, Zbigniew W. Ras, "CHASE-2: Rule based chase algorithm for information systems of type lambda," *Active Mining*, vol. 3430, pp. 255-267, 2005.

[40] E. F. Codd, "Further normalization of the data base relational model," IBM Research Report, San Jose, California, RJ 909, 1971.

[41] B. J. Frey, D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, vol. 315, pp. 972-976, 2007.

[42] Yanying Li, Youlong Yang, Jinxing Che, Long Zhang, "Predicting the Number of Nearest Neighbor for kNN Classifier," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 662-669, 2019.

[43] Thorsten Papenbrock, Felix Naumann, "A Hybrid Approach to Functional Dependency Discovery," *SIGMOD*, pp. 821-833, 2016.

[44] P. A. Flach, I. Savnik, "Database dependency discovery: a machine learning approach," *AI Communications*, vol. 12, no. 3, pp. 139-160, 1999.

BIOGRAPHY

Huaiguang Wu received the B.S. degree in industrial electrical automation from the Faculty of Electrical Engineering and Automation, Luoyang Institute of Science and Technology, the M.S degree in computer software and theory from the Faculty of Computer, Zhejiang Normal University, the Ph.D. degrees in software engineering from the Faculty of Computer, Wuhan University, China.

He is currently a Professor of the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, China. His research interests include formal methods, software engineering and algorithms.

Shuaichao Li received B.S. degree in software engineering and is currently a postgraduate student in electronic information from the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, China. His research interests include missing values imputation, entity resolution and conflict resolution.

Wenjun Shi is currently a Lecturer of the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, China. Her research interests include big data analysis and algorithms.

Shaoqing Du received B.S. degree in software engineering and is currently a postgraduate student in electronic information from the Faculty of Computer and Communication Engineering, Zhengzhou University of Light Industry, China. His research interests include data cleaning, conflict resolution.