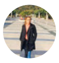
Missing value imputation through shorter interval selection driven by Fuzzy C-Means clustering

Article *in* Computers & Electrical Engineering · July 2021

DOI: 10.1016/j.compeleceng.2021.107230

3 authors:



Hufsa Khan
Shenzhen University
5 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Xi-Zhao Wang
Shenzhen University
334 PUBLICATIONS **9,425** CITATIONS
SEE PROFILE

Han Liu
Shenzhen University
114 PUBLICATIONS **1,552** CITATIONS
SEE PROFILE

Highlights

Missing value imputation through shorter interval selection driven by Fuzzy C-Means clustering

Hufsa Khan,Xizhao Wang,Han Liu

- A missing data imputation technique called SISFCM is proposed for numeric features.
- The SISFCM approach improves the imputation performance in comparison with other competitive state-of-the-art imputation methods.
- The method shows its robustness to the change of the percentage of missing values.

Missing value imputation through shorter interval selection driven by Fuzzy C-Means clustering

Hufsa Khan[a], Xizhao Wang[a,b] and Han Liu[a,*]

[a] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*

[b] *College of Computer Science and Software Engineering, Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, Guangdong, China*

ARTICLE INFO

Keywords:
Incomplete data processing
Missing value handling
Missing value imputation
Fuzzy C-Means clustering

ABSTRACT

The presence of missing data is a common and pivotal issue, which generally leads to a serious decrease of data quality and thus indicates the necessity to effectively handle missing data. In this paper, we propose a missing value imputation approach driven by Fuzzy C-Mean clustering to improve the classification accuracy by referring only to the known feature values of some selected instances. In particular, the missing values for each instance are imputed by selecting a shorter interval based on the cluster membership value within the certain threshold limit of each feature, while using a short interval is considered to improve the imputation effectiveness and get more accurate estimation of the values in comparison with using a long interval. Our method is evaluated through comparing with state-of-the-art imputation methods on UCI datasets. The experimental results demonstrate that the proposed approach performs closely to or better than those state-of-the-art imputation methods.

1. Introduction

Imputation of missing values is an important and extensively used technique when dealing with incomplete data. A data set is considered complete if it contains no missing value, otherwise it is incomplete. There are multiple reasons for the presence of missing data, such as, loss of files, improper record of data, imperfect manual data entry procedures, equipment errors, incorrect measurement and forgetting or refusing to answer a survey question [1]. Different studies indicate that 5% or more data values can often be missed (lost) unless extraordinary care is taken by the organization [2]. In such a situation, when data have imperfection, a pre-processing stage is required in which data are cleaned and prepared. In case of automatic classification, a system should have its own mechanism of pre-processing to handle the missing data. In the pre-processing stage, the easiest approach of dealing with missing data is to remove all the instances that have missing values. However, this strategy is suitable only when a data set holds a considerably smaller number of instances comprising with missing data and when the inference analysis of complete instances (data) will not lead to serious bias [3].

Using poor quality data, with incorrect and missing values, can lead to an erroneous and non-sensible conclusion, making useless the entire process of data collection and analysis for the users. Therefore, in order to deal with missing values, three general approaches are mainly used, namely, ignorance, deletion and imputation [4]. In the first approach, incomplete observations are usually ignored but the drawback is the loss of effectiveness due to ignorance of incomplete observation and bias estimates in a systematic way [5]. The second approach is designed to delete all the records having missing values [6]. This approach is simple but it has a serious disadvantage that, while using small-sized data, the correctness of statistical analysis will be reduced [7]. The third approach for handling missing data is referred to as imputation, which aims to fill the missing values by some estimation methods [4, 5].

Currently for missing data handling, imputation is the most widely used approach [3], where various imputation methods have been proposed [4, 8, 9]. However, the performance of imputation depends mostly on the suitability of the selected method and the characteristics of data [10], i.e., for missing value imputation, different techniques generally involve different strategies of processing, which show different degrees of suitability for different characteristics of data. Specifically, imputation methods are divided into two categories, namely, statistical techniques and machine

[*] Corresponding author.

hufsakhan@email.szu.edu.cn (H. Khan); xizhaowang@ieee.org (X. Wang); han.liu@szu.edu.cn (H. Liu)
ORCID(s): 0000-0002-0037-1448 (H. Khan)

learning techniques [11]. In the setting of statistical techniques, missing values are usually estimated by considering statistical measures, such as mean/mode and covariance of data. Imputing missing values based on the mean/mode of the observed values from an instance or attribute is a good approach in some but not all cases, as sometimes the mean/mode of the observed values is not close to the true estimate of the missing value, which affects the imputation performance.

In this situation, to make the missing value imputation more precise and accurate, it is necessary to select only those values which are closer to the true estimates of the missing values by means of selecting a shorter interval of observed values. On the other hand, in the setting of machine learning techniques, the most effective estimate to impute the missing value is usually achieved by using the most similar patterns or features. Among these techniques, clustering is considered as a proper tool for missing data handling due to its ability to divide the data into different clusters and to find the relevant patterns which are considered an appropriate set of donors for imputation. In this paper, motivated by the idea that an estimation to missing data coming from a shorter interval will be more accurate than that from a longer interval, we propose a new approach by selecting a shorter interval within which missing data are.

In particular, we use a powerful and well known clustering algorithm which is referred to as Fuzzy C-Mean clustering [8] to achieve the above-mentioned shorter interval selection to improve the classification accuracy with the help of imputed data. This powerful clustering algorithm is not only useful for missing data handling but also has significant usefulness in other applications, e.g., it has been used in image segmentation [12] to obtain the higher segmentation precision, and in gene expression data to solve the gene space or sample space problem [13]. Furthermore, nowadays to address the classification issues for complex data sets (i.e., facial recognition data, biological data and handwritten digital images), the focus on clustering algorithms based on density strategy is also increased [14]. Inspired from the wide applications of Fuzzy C-Mean clustering in various areas, we adopt this algorithm to drive our proposed approach of missing value imputation. The experimental results show that our approach leads to more effective imputation of missing values based on the observed values within a shorter interval in comparison with using a longer interval. The main contributions of this study include:

- A new effective imputation technique, called shorter interval selection driven by Fuzzy C-Means (SISFCM), is proposed.
- Our proposed imputation approach has been evaluated theoretically and experimentally, and the results indicate that the imputation of missing data based on the instances from a shorter interval is more accurate than that from a longer interval.
- It is experimentally validated that our proposed approach shows better performance on 12 data sets from the UCI machine learning repository [15], in comparison with those state-of-the-art methods.

The rest of the paper is organized as follows: Section 2 reviews related work about missing data imputation. Our proposed approach (SISFCM) of missing data imputation is presented in Section 3 . Section 4 reports the experimental results and analyses the effectiveness of the proposed approach by comparing its performance with that of nine state-of-the-art imputation techniques. Finally, the concluding remarks are drawn in Section 5.

2. Related work

This section provides an overview of missing data imputation techniques. Recently, several missing value imputation techniques have been proposed [16, 17, 18]. Some of the most commonly used imputation approaches are K-nearest neighbour imputation (KNNI) [19, 20], expectation maximization imputation (EMI) [1, 21], regression imputation (RI) [22, 23], knowledge-based imputation [24, 25], and fuzzy C-means imputation (FCMI) [26, 27].

In [19, 20], an intuitive imputation method for missing values (KNNI) is presented. In particular, KNNI is designed to find the k value (representing the number of nearest neighbors) and the missing value of a feature of an instance is replaced with the mean value computed on the basis of the values of the feature of the selected nearest neighbors [28]. The KNNI method can perform well when the data size is small. However, while the sample size of training data becomes larger, it will cause more computational cost to find the k nearest neighbors from the large training set. Based on KNNI, another missing value imputation approach referred to as weighted k-nearest neighbor imputation (WKNNI) is proposed by Troyanskaya et al. [29]. In this method, for better estimation of missing values, weights are assigned to nearest neighbours based on Euclidean distance between the instance with missing values and each of the nearest neighbours. The estimation results showed that the performance of WKNNI is better than that of the original KNNI method.

According to [21] about expectation maximization imputation (EMI), maximum likelihood estimation (MLE) is the most critical step [30, 31]. Therefore, a well-known probability distribution of MLE is used to estimate the missing data, and when the change of estimated data stopped, the estimation task is completed. [1] also proposed a fuzzy clustering based expectation maximization (FEMI) technique for imputation of both categorical and numerical data. However, this approach is suitable only for the case of missing completely at random (MAR), and parameters estimation through MLE is a main step of the EMI approach. Another interesting thing to explore is how to manage an appropriate assumption of the estimated parameters through MLE.

An additional type of imputation techniques is knowledge based imputation. In this context, Qi et al. [24] highlighted that additional knowledge was not sufficiently used for existing imputation approaches. However, in the setting of knowledge based imputation, missing values can be handled more effectively by taking advantage of public knowledge [32]. Although missing values can be filled by involving human intelligence in knowledge based imputation, some other drawbacks exist in this kind of methods. The first one of the drawbacks is the mismatching of missing data that would affect the estimation effectiveness and efficiency. The second one is the absence of potential knowledge on missing value, leading to low effectiveness in the imputation procedure.

Moreover, clustering is also one of the most commonly used 17 imputation methods, e.g., K-mean and Fuzzy K-mean (C-mean or Fuzzy C-mean). According to [33], the key step of K-mean clustering is the estimation of the centroid position of cluster (c_1,c_2,...,c_k). In [1], the Fuzzy K-means approach is utilized for missing value estimation. In this approach, each of the instances is associated with all clusters with different membership degrees. Theoretically, the main challenge for using Fuzzy K-mean is determination of the number of clusters and the membership degree to which an instance belongs to each of the clusters [34]. It is worth noting that Fuzzy C-Mean splits the data set into different clusters (c_1,c_2,...,c_k), which make the imputation more effective by means of imputing missing values of an instance based on the known values of those similar instances within the same cluster. However, how we can make imputation more effective by selecting some instances within a specific cluster is another aspect of further investigation.

Fuzzy C-Mean has also been used together with support vector regression (SVR) and genetic algorithm (GA) to impute missing values. In particular, the parameters of Fuzzy C-Mean are optimized with GA, i.e., optimizing the cluster size and the weighting factor. It is concluded in [8] that when parameters are optimized then the performance of Fuzzy C-Mean becomes better. However, Fuzzy C-Mean clustering involves only one iterative step to impute the missing values, without checking how effectively the missing values in a data set can be imputed. Therefore, this paper aims to improve the effectiveness of imputing missing values through shorter interval selection based on Fuzzy C-Means clustering, so that the imputed value is more accurate as compared to longer interval.

All of these previous works on missing data imputation indicate that different approaches are suitable for different kinds of problems. Furthermore, Fuzzy C-Mean have the capability to achieve good performance based on the membership values (i.e., an association). Although Fuzzy C-Mean can make imputation more effective by taking advantages of the higher similarity between the instance with missing values and each of the other clean instances within the same cluster, Fuzzy C-Mean imputes the missing data once in one time and its interval is related to the cluster size (the number of instances in each cluster). Therefore, on the basis of the Fuzzy C-Mean clustering method, we propose a shorter interval selection method to achieve more effective imputation of missing data towards improving the classification accuracy. We will present in Section 3 the procedure of the proposed approach in details and show that this shorter interval selection driven by Fuzzy C-Mean clustering is more effective for missing value imputation in comparison with selecting a longer interval.

3. Proposed missing value imputation technique

3.1. Research framework

Before a detailed discussion about the proposed methodology, we firstly introduce the overall framework, our contribution and the basic concept of this research. In Fig. 1, it can be seen that the research framework involves 5 steps.

In Step 1, data set D_f is partitioned into three parts: the first one is observed/clean data (D_{obs}), the second one is missing data (D_{mis}) and the last one is test data (D_{test}) (see Fig. 1). In this research, we artificially created missing

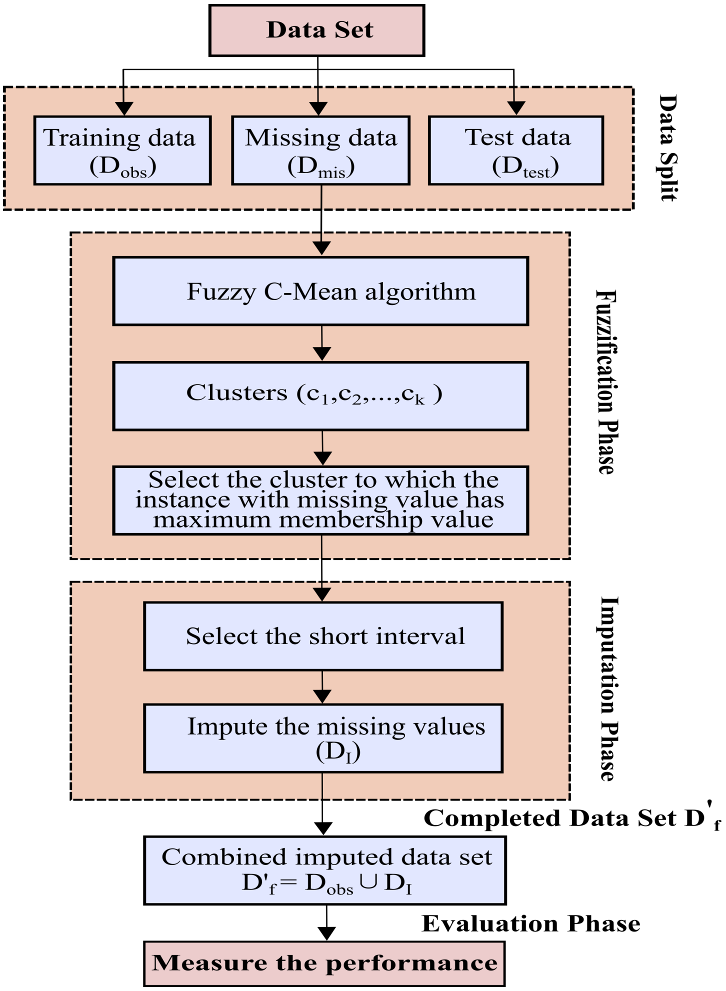

Data Set
Data Split
Training data (D_{obs})
Missing data (D_{mis})
Test data (D_{test})
Fuzzification Phase
Fuzzy C-Mean algorithm
Clusters (c1,c2,...,ck)
Select the cluster to which the instance with missing value has maximum membership value
Imputation Phase
Select the short interval
Impute the missing values (DI)
Completed Data Set D'f
Combined imputed data set D'f= Dobs ∪ DI
Evaluation Phase
Measure the performance

Figure 1: The overall work flow of the proposed SISFCM imputation technique.

values with the ratio of 20% in (D_{mis}) and these missing values are initially filled with zeros. In Step 2, the Fuzzy C-Mean clustering algorithm is applied to create a number of clusters alongside their centroids and each of the instances is assigned a membership degree to each specific cluster (see Algorithm 1). In Step 3, we apply our SISFCM method to impute missing values (see Algorithm 2). In Step 4, we combine the clean data and the imputed data to form a complete training data set D'_f. In Step 5, the imputation performance is measured by using different classifiers i.e., Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Gaussian Naive Bayes (NB), and Support Vector Machines (SVM).

3.2. Fuzzy C-Means clustering algorithm

The Fuzzy C-Mean clustering algorithm has been used in a wide range of clustering related tasks, which is one of the most effective clustering algorithms. The main objective of clustering is to divide the given data set into different groups based on the similarity and dissimilarity of the instances. There are two types of clustering methods. The first one is hard clustering and the second one is soft (fuzzy) clustering. In the first type of clustering, from a data set D_f a record R_i belongs to one and only one cluster, i.e., each record has the membership value of 1 to one cluster and the membership value of 0 to each of the other clusters. However, in the second type of clustering, the record R_i has a certain association (membership value $mv \in [0, 1]$) to each cluster. In this study, we consider the second type of clustering, where the membership value of a record R_i varies between 0 and 1.

Let R= { $R_1, R_2, \dots , R_n$ } represents a set of records and A= { $A_1, A_2, \dots , A_m$ } represents a set of attributes of data set D_f. Moreover, for k clusters, the record R_i has k centroids (i.e.,v= $\{v_1, v_2, \dots, v_k\}$) and k membership values (i.e., $\mu = \{ \mu_{i1},\mu_{i2},...,\mu_{ik} \}$). In general, for each instance, the sum of membership values to the created clusters is equal to 1 (as shown in Eq.(1)).

$$\sum_{j=1}^{k} \mu_{ij} = 1, \forall j = 1, 2, ...k \tag{1}$$

The centroid of a cluster is calculated based on the formula shown in Eq. (2) .

$$v_k = \frac{\sum_{i=1}^{n} (\mu_{ij})^m R_i}{\sum_{i=1}^{n} (\mu_{ij})^m}, 1 \leq m \leq \infty \tag{2}$$

Here, m (m>1) is a fuzzification coefficient, and in this study we consider its value to be 2 (m=2), where μ_{ij} represents the membership value of R_i in the j-th cluster and n is the number of data points. The Euclidean distance formula is used to calculate the distance between two points, which is denoted as;

$$d_{ij} = \|R_i - v_j\|^2 = \sqrt{\sum_{l=1}^{k} (R_{il} - v_{jl})^2} \tag{3}$$

$\| R_i - v_j \|$ represents the Euclidean distance between the i-th instance and the centroid of the j-th cluster. After calculating the Euclidean distance, membership value μ_{ik} is updated with the formula:

$$\mu_{ik} = \frac{1}{\sum_{i=1}^{n} \frac{\|R_i - v_j\|^{2/m-1}}{\|R_i - v_j\|}} \tag{4}$$

where i is the index of the iteration step and n is the number of data points. In Fuzzy C-Mean clustering, the process of calculating the membership and the centroid of each cluster is iterative, i.e., it continues until a termination condition is met.

Algorithm 1: Fuzzy C-Mean

```
Input : Data set $D_f$ having | $R$ | records and | $A$ | attributes.
Output : Calculated membership value and centroid of cluster.
Step 1:
Begin:
    Initialize termination criteria $\varepsilon > 0$
    Fuzzification parameter $m$ >1
    Initialize the cluster number ($k$=n, $2 \leq k \leq$ n)
end
Step 2:
Begin:
    Initialize the centroid v= $\{v_1, v_2, \dots, v_k\}$
    V= { }
    Repeat
    V= $v'$
end
 Step 3:
Begin:
    Calculate the centroid v= $\{v_1, v_2, \dots, v_k\}$ of $k$ clusters according to Eq.( 2).
    Calculate the membership degrees $\mu_{ik}$ of each record $R_n$ to each cluster $c_k$ according to Eq.( 4).
    Until $\|V - v'\| < \varepsilon$
    Return $v'$
end
```

Algorithm 2: Shorter Interval Selection Driven By Fuzzy C-Means Clustering

```
Input : Data set $D_f$ containing $|R|$ records and $|A|$ attributes.
Output : An imputed dataset $D_f^{'}$ containing $|R|$ records and $|A|$ attributes.
Step 1:
Begin:
    Divide data set $D_f$ into $D_{obs}$ (having no missing value) and $D_{mis}$ (having missing value) data set.
end
Step 2:
Begin:
    Apply Fuzzy C-Mean clustering algorithm on $D_{obs}$ and $D_{mis}$ data set to find centroid v= $\{v_1, v_2, \dots, v_k\}$
      and membership values µ =$\{\mu_{1k}, \mu_{2k}, \dots, \mu_{ik}\}$ of k-clusters.
end
Step 3:
Begin:
    Select the highest membership value and cluster number of missing attribute.
    For each record $R_i \in D_{mis}$ do
        For each attribute $A_k \in A$ do
            If $A_k$ is missing then
                If $A_k$ is numerical then
                    Impute $R_{ik}$ with the calculated average mean:
                        $R_{ik}$ with the calculated average mean (using Eq.( 10))    /* against missing Value
                          check its cluster number and take average of all the values having same cluster
                          number in $D_{obs}$ */
                    end
                end
            end
        end
    end
end
Step 4:
Begin:
    Completed dataset $D_f^{'}$ =$D_{obs} \cup D_I$
    Return $D_f^{'}$;
end
```

3.3. Problem formulation

In this paper, we use the shorter interval selection approach driven by Fuzzy C-Mean clustering in order to effectively impute the missing values. In particular, this clustering algorithm (Fuzzy C-Mean) is used to divide the complete data set into different clusters and calculate the membership value (μ_{ik}) and the centroid of each cluster (v_k), where the objective is to make imputation of missing values more effective.

In the setting of our proposed imputation strategy, two types of data are involved, where one is missing data and the other one is observed clean data. This can be defined as having a data set D_f with i rows and j columns, which indicate that D_f has i records and each record has j attributes. The data set D_f contains n instances/records (denoted in Eq. (5)) and m attributes (denoted in Eq. (6)).

$$D_f = (R_1, R_2, \dots, R_n) \tag{5}$$

where $R_i (1 \leq i \leq n)$ is i^{th} record of data set D_f.

$$A = (A_1, A_2, \dots, A_m) s.t A_j (1 \leq j \leq m) \tag{6}$$

Suppose we have a score matrix where $X_{i,j}(1 \leq i \leq n, 1 \leq j \leq m)$ represents the value of the j^{th} attribute of the i^{th} record of the data set D_f. A subset of D_f is referred to as either clean data (observed data) or missing data on the basis of Eq. (7).

$$Y_z(A_j) = \begin{cases} X_{i,j} & \{1 \leq i \leq n, 1 \leq j \leq m : \forall_{i,j} \in N\} \\ nan & \{1 \leq i \leq n, 1 \leq j \leq m : \forall_{i,j} \in N\} \end{cases} \tag{7}$$

In this equation, $Y_z(A_j)$ represents the j^{th} attribute of the i^{th} record of D_f. Furthermore, $Y_z(A_j)$= *nan* indicates that the value of the j^{th} attribute of the i^{th} record is missing otherwise it is complete. Mathematically we can describe the above-mentioned value in such a way as:

$$P_i^{miss} = \{1 \leq i \leq n, 1 \leq j \leq m, Y_z(A_j) | \exists Y_z(A_j) = nan; \forall_{i,j} \in N\} \tag{8}$$

$$Q_i^{obs} = \{1 \leq i \leq n, 1 \leq j \leq m, Y_z(A_j) | \forall Y_z(A_j) \neq nan; \forall_{i,j} \in N\} \tag{9}$$

At the beginning, an initial guess is made for the missing value imputation, let D= $\{R_1, R_2, \dots, R_n\}$ be a data set and A= $\{A_1, A_2, \dots, A_m\}$ be a set of features to be processed. Let A_j be the feature with missing values for some instances. Furthermore, suppose feature A_j in the missing data set needs to be imputed and the imputed value is estimated based on the clean data X= $\{x_1, x_2, \dots, x_t\}$. The Fuzzy C-Means clustering algorithm is applied on both kinds of data sets (i.e., the observed data (D_{obs}) and the missing data (D_{miss})) (see Algorithm 1), and the membership values (μ_{ik}) of each instance to various clusters are computed by using Eq. (4). The cluster (c_k) to which the instance (with missing value on A_j) has the highest membership value is selected to impute the missing value of of A_j.

On the basis of membership values and clusters, for an instance R_i= $\{a_{i1}, a_{i2}, \dots, a_{im}\}$, each feature A_j with a missing value is handled by imputing the weighted average of the values of those selected clean instances in the same cluster c_k, based on Eq. (10), where the selected clean instances form a shorter interval of values for feature A_j, in comparison with the length of the original interval (domain) of feature A_j.

$$y_{kmean} = \frac{1}{t} * \sum_{i=1}^{t} x_{pq} \tag{10}$$

Where t is the number of records in X and x_{ij} is the value of the j^{th} attribute of the i^{th} record. The imputation procedure is continued until the stopping criteria is met. The Pseudo code of Algorithm 2 represents the procedure of the SISFCM imputation method. After imputing the missing value, the performance is evaluated.

3.4. Theoretical analysis

The proposed method is motivated by the idea that an estimation to missing data coming from a shorter interval will be more accurate than that from a longer interval. In particular, we assume that shorter interval based missing value imputation is considered more accurate as compared to longer interval based one. In this section, we prove this concept (a shorter interval is better than a longer interval) with the use of Bayes Theorem which is stated as below:

$$p(y \mid x_1, \dots, x_n) = \frac{p(y)p(x_1|y)p(x_2|y)\dots p(x_n|y)}{p(x_1)p(x_2)\dots p(x_n)} \tag{11}$$

It can be expressed as:

$$p(y \mid x_1, \dots, x_n) = \frac{p(y)\Pi_{i=1}^{n} p(x_i|y)}{p(x_1)p(x_2)\dots p(x_n)} \tag{12}$$

If the denominator remains constant for a given input, then we can remove the term:

$$p(y \mid x_1, \dots, x_n) \propto p(y)\Pi_{i=1}^{n} p(x_i \mid y) \tag{13}$$

Suppose, for a missing value, we have an imputed value ($Q^{imputed}$) to replace it and have a true value (P^{true}) to judge whether the imputed value is accurate. We want to analyse the effectiveness of $Q^{imputed}$, i.e., how effectively the missing value is imputed in comparison to the true value (P^{true}). The effectiveness analysis is performed by using a shorter interval (X), and a longer interval (Y), respectively.

Let us make the following denotations:

X denotes the length of the shorter interval.

Y denotes the length of the longer interval.

T denotes the case of correct imputation.

F denotes the case of incorrect imputation with a high error degree, i.e., the imputed value is far away from the acceptable range.

T/F denotes the case of incorrect imputation with a low error degree, i.e., the imputed value is not exactly the same as the true value (P^{true}), but the imputed value is within an acceptable range, e.g., it differs from P^{true} by two upward value (P^{true}+2) or two downward value (P^{true}-2).

F/T denotes the case of incorrect imputation with a medium error degree, i.e., the imputed value is outside the above-mentioned acceptable range, but it is close to the upper or lower bound of the range.

It is worth noting that the above-defined acceptable range can be adjusted based on the characteristic of specific data. In our case, we assume that the acceptable range of imputed values involves P^{true}+2, P^{true}+1, P^{true}, P^{true}+1, P^{true}+2. Suppose, $Q^{imputed}$ is exactly the same as the true value (P^{true}) then the probability of T is $1/X$, whereas T/F probability (the chance of differing from the true value by P^{true}-2 or P^{true} +2) is $4/X$. Likewise, the probability of F/T is $4/X$, as shown in Table 2 and the probability of F is $X - 9/X$.

Table 1
Naive Bayes Theorem based calculated values.

Accepted	Rejected
$9/X$	$(X\text{-}9)/X$

Thus, we can see that if the imputed value is within the acceptable range [P^{true}-2, $P^{observed}$+2], then $T, T/F, F/T$ conditions are satisfied. Otherwise, condition F will be satisfied (see Table 2). It indicates that in a larger interval the chance of satisfying the F condition will be greater than that in a shorter interval. As the range of the possible imputation values is increased, the chance of $Q^{imputed}$ being selected will be decreased and vice versa. Therefore, the probability of imputing the exactly true value or a value in the acceptable range within a shorter interval is higher as compared to that within a longer interval. Table 1 shows the total probability of getting an accepted or rejected imputed value from an interval and Table 2 is the specialization of Table 1.

We use the following formula to calculate the highest probability of getting an imputed value within the defined acceptable range, based on a longer or shorter interval.

$$P(\text{imputed value} \in \text{acceptable range}) = p(T) + p(TF) + p(FT) \tag{14}$$

Based on Eq. (14), we can derive the following:

$$P(\text{imputed value} \in \text{acceptable range}) \propto 9/X \tag{15}$$

Table 2

T	F	T/F	F/T
$1/X$	$(X\text{-}9)/X$	$4/X$	$4/X$

It can be deduced from the above equation that the probability of getting the correctly imputed value is inversely proportional to the range of the imputable values, i.e., the smaller the value in the denominator is, the higher the probability would be. In the case that a shorter interval with the length of X is replaced with a longer interval with the length of Y, as shown in Eq. (16), the probability of getting an imputed value within the defined acceptable range is reduced.

$$P(imputedvalue \in acceptablerange) \propto 9/Y \tag{16}$$

4. Experiments

4.1. Experimental framework

In this section, we present the experimental results to validate the proposed imputation approach. The experiments are conducted on 12 UCI data sets [15] for verifying how each imputation method affects the classification accuracy. In particular, we compare the imputation performance of our technique SISFCM with that of the nine state-of-the-art imputation methods, i.e., Nearest Neighbors Imputation (NNI) [35], Multiple Imputation by Chained Equations (MICE) [36], Softimpute [37], MissForest [38], Matrix Completion (Matrix) [39], Weighted K-Nearest Neighbors Imputation (WKNNI) [29], Singular Value Decomposition (SVD) [29], SvrFcmGa imputation [8], and one deep learning [40, 41] based method called Generative Adversarial Imputation Nets (GAIN) [9]. Furthermore, we evaluate the performance of our imputation method in different settings (i.e., having data with different missing rates). The experiment on each data set is run 10 times and then the average accuracy of classification (including the standard deviation) is taken for the final evaluation of the imputation performance. The performance evaluation is particularly done by using six supervised learning algorithms (LR, LDA, KNN, CART, NB, and SVM) for training classifiers on each data set that contains a mixture of clean data and imputed data. The missing values in each data set are created artificially by randomly removing 20% of the attribute values of all the data points.

4.2. Datasets description

The details about the characteristics of the 12 data sets, such as the number of data instances, attributes and classes, can be seen in Table 3. In this table, we can see that there are some data sets (i.e., autoMPG and ozone) that already have missing values. In our experiments, clean data sets are required for evaluating the effectiveness of missing value imputation. Therefore, in the pre-processing stage, we remove all the records (instances) that have missing values. Instead, we artificially create missing values in these clean instances, which will be imputed later on using SISFCM and those other imputation techniques selected for comparison. In this setting, it is straightforward to evaluate the performance of the imputation techniques. In other words, the original values of the artificially created missing data are known, so it is straightforward to evaluate how the imputation of missing values impacts on the classification performance in comparison with the performance obtained on the clean data, while the same learning algorithm is used for training classifiers. In contrast, if a data set with missing values is used for evaluation, then it is difficult to evaluate how effectively our imputation method can lead to a positive impact on the classification performance.

4.3. Experimental setting

For our imputation approach SISFCM, it is required to impute suitable values in some experimental settings through hyper-parameter optimization. In particular, the k value for Fuzzy C-Mean clustering is initially set in the range of 3-25 for all data sets and the best value of k leading to the highest validation accuracy is selected. The fuzzification parameters m=2, iteration=200, and stopping criteria epsilon (ϵ)=0.001 are selected in this study. Furthermore, to get more optimized performance we narrow down the range of the membership values towards making the imputed

Table 3
Details of UCI data sets used in the experiments.

Data Set	Records	No. Attr.	Classes	Missing Values
Iris (Ir)	150	4	3	No
Wine (Wi)	178	13	3	No
Yeast (Ye)	1484	8	10	No
Winequality (WQ)	4898	12	10	No
AutoMPG (MPG)	398	8	3	Yes
Seeds (Se)	210	8	3	No
Cleveland (Cl)	303	14	4	No
Ozone (Oz)	2536	73	2	Yes
Pima (Pi)	768	8	2	No
Pen digits (PD)	10992	16	10	No
ImageSegmentation (IS)	2310	19	7	No
Optdigits (OD)	5620	64	10	No

value more accurate. In particular, we select those instances having membership values closer to each other so that the imputed value can be more precise and optimal.

4.4. Justification of K

For our proposed technique, since k clusters are required for the fuzzy clustering (Fuzzy C-Mean) stage, the k value is selected automatically following the procedure specified below:

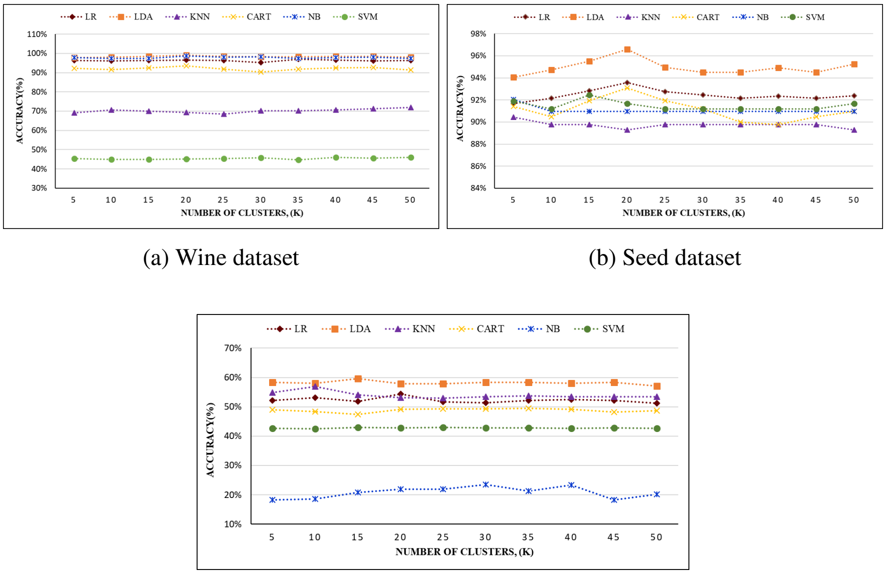
Initially, we define different ranges (i.e., 3-5, 5-10, ..., 45-50) for the selection of the best value of k. In order to select a finally suitable range for k, we first evaluate the performance of SISFCM for different data sets (i.e., wine, seed and yeast) by adjusting the range of k values as shown in Fig. 2, and the best value of k is then selected from this defined range on the basis of the highest membership value. The advantage of this approach is that the best k value is determined by evaluating the imputation performance on the validation data rather than evaluating the clustering performance for each k value. As we can see from Fig. 2, across all the defined k values, the accuracy scores are close to each other, which indicate that our method is less sensitive to the selection of the k value.

4.5. Performance evaluation matrix

In this paper, the imputation performance is evaluated by using six well known supervised learning algorithms, i.e., LR, LDA, KNN, CART, NB, and SVM. We select these algorithms to indicate the effect of imputed data on the classification performance. Table 4 demonstrates that the performance of classification is considerably improved using our proposed approach in most of the cases in comparison with other imputation methods. It is worth noting that the aim of the experiments is to analyse the effect of missing value imputation on the classification performance. Therefore, it is not a suitable choice to use the RMSE, MAE, etc., for performance evaluation, i.e., a lower RMSE or MAE cannot directly indicate a higher score of classification accuracy. That's why we selected different algorithms of classifiers training to evaluate the classification accuracy in the presence of missing values and in the absence of missing values. Therefore, to show the effect of missing value imputation, the classification accuracy is measured.

We evaluate the performance of SISFCM from two aspects. First, we want to identify how close the imputed values provided by SISFCM are to the true values. Second, we want to evaluate how well our method performs in comparison with the other methods. Therefore, to evaluate our method we split the original data set D_f into 3 parts, clean data (D_{obs}), missing data (D_{mis}) and test data with the ratio of 60%, 20%, 20%, respectively as shown in Fig. 3. The imputed data (D_I) and clean data (D_{obs}) are merged $D_f^{'} = D_{obs} \cup D_I$ and a complete data set $D_f^{'}$ is obtained. Our performance evaluation procedure involves two steps as follows:

In Step 1, we train classifiers on clean data using six algorithms (LR, LDA, KNN, CART, NB, and SVM.) and evaluate the performance of each classifier on test data. In Step 2, we train classifiers on the data set that contains a mixture of clean data and imputed data $D_f^{'}$ ($D_f^{'} = D_{obs}$ (60%) $\cup$ D_I (20%)) and test these classifiers on test data as shown in Fig. 3. The results obtained in these two steps are shown in Table 4.



(a) Wine dataset

(b) Seed dataset

(c) Yeast dataset

Figure 2: Performance of SISFCM imputation method at different classifiers for different number of clusters. The x-axis represents different ranges for cluster k and y-axis represents the accuracy (%) at different cluster range.

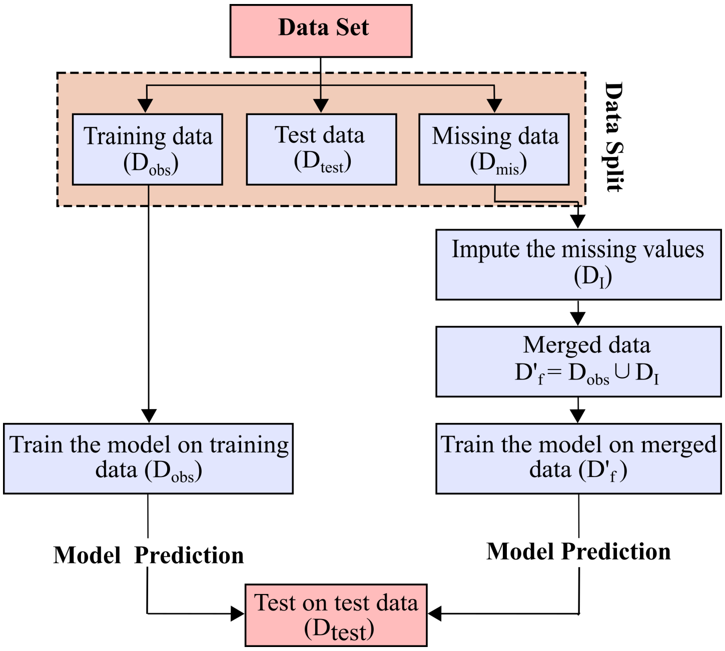


Figure 3: Work flow of performance measure for model evaluation of the proposed SISFCM technique.

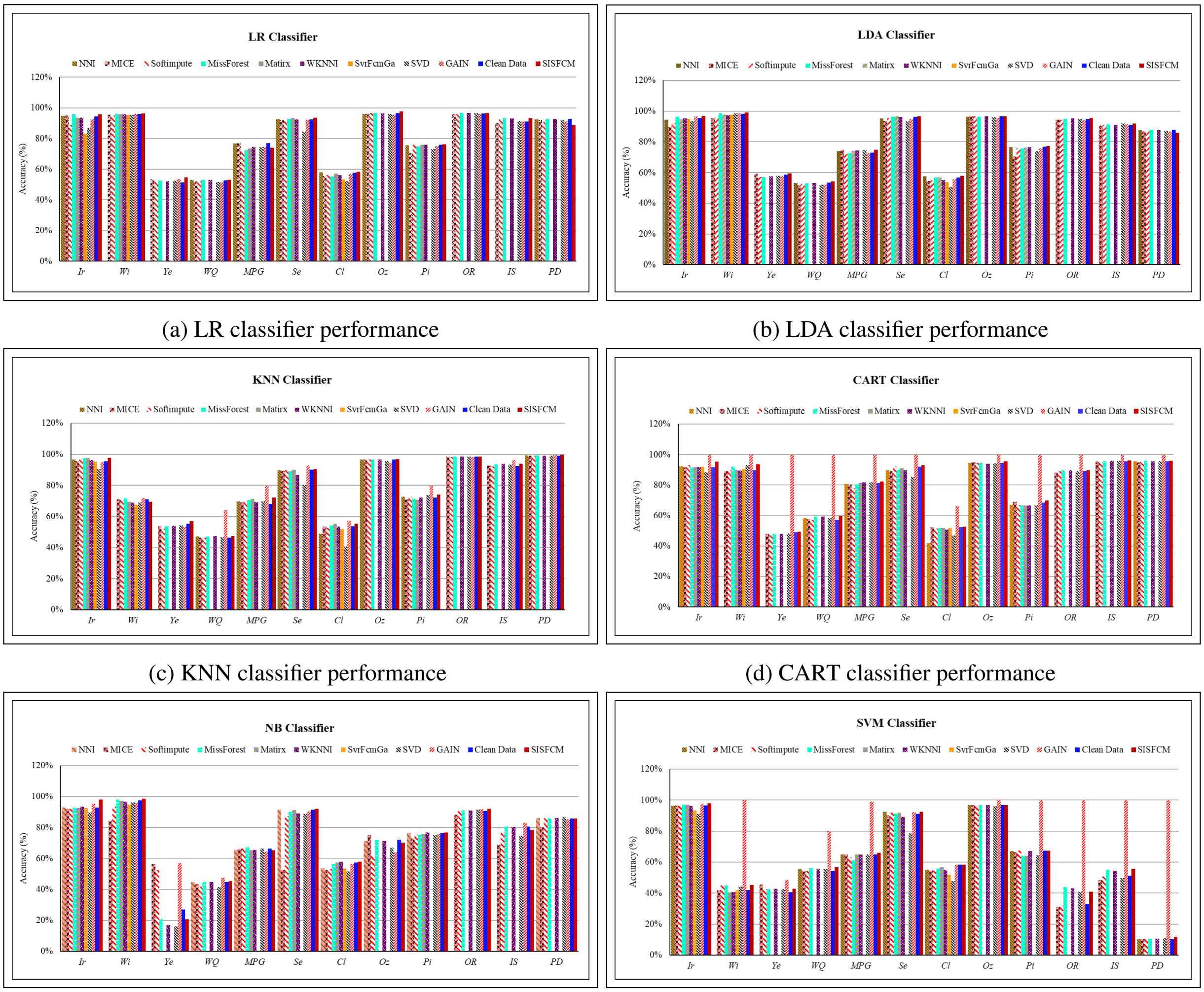
It can be seen from Table 4 that the accuracy rate of SISFCM is higher than that of the comparison imputation methods in most cases. Furthermore, we can claim that on a large amount of data sets the performance of SISFCM is encouraging in comparison with that of the other imputation methods. The results also show that the classification performance is improved with the help of imputation through shorter interval selection.

Table 4
Accuracy (%) of NNI, MICE, WKNNI, SVD, SvrFcmGa, MissForest, softimpute, Matrix, GAIN,Clean Data and SISFCM imputation methods against different classifiers (with LR, LDA, KNN, CART, NB, and SVM) for 12 real world UCI data sets.

Dataset	Classifiers	NNI	MICE	Softimpute	MissForest	Matirx	WKNNI	SvrFcmGa	SVD	GAIN	Clean Data	SISFCM
Ir	LR	94.666±3.399	95.000±2.687	89.000±7.000	95.666±3.000	93.666±3.785	93.666±3.378	83.000±6.403	87.000±3.144	92.573±3.576	94.333±3.000	**95.888±3.103**
	LDA	94.333±4.484	89.666±5.259	92.333±4.955	96.333±3.480	94.666±4.000	95.333±3.711	95.000±4.772	93.666±1.000	96.504±2.705	95.666±3.611	**97.000±2.768**
	KNN	96.666±3.333	95.666±5.174	96.666±3.333	97.333±2.494	97.666±2.603	96.331±3.785	95.333±4.000	90.333±1.000	94.980±2.967	95.666±5.121	**97.711±2.000**
	CART	92.333±3.958	91.666±1.666	93.000±2.768	91.666±3.073	91.666±3.415	91.666±2.612	92.000±3.055	88.333±2.687	**100.000±0.000**	91.666±3.000	**95.333±3.055**
	NB	93.000±4.068	92.000±6.182	92.000±4.526	93.000±3.415	92.666±3.887	93.333±3.312	92.666±4.422	89.666±2.768	95.364±3.290	93.000±3.715	**98.000±2.211**
	SVM	96.333±3.144	96.666±3.333	96.333±2.768	97.000±2.768	97.000±2.768	96.333±3.444	93.333±2.981	91.000±1.527	97.629±2.615	96.666±3.333	**98.000±2.211**
Wi	LR	-	95.555±3.333	95.000±3.239	96.111±3.333	95.833±3.560	95.711±3.499	95.25±3.812	95.511±1.580	96.011±3.818	96.111±2.832	**96.383±3.298**
	LDA	-	95.277±4.487	95.833±3.344	98.611±1.863	97.777±2.078	97.500±2.286	97.755±2.732	98.622±1.689	98.566±2.187	98.333±3.333	**99.111±1.388**
	KNN	-	71.111±1.192	70.555±9.060	**71.666±8.127**	69.444±7.657	68.922±5.063	67.611±6.121	68.966±0.000	**72.022±8.246**	71.111±1.192	69.444±9.781
	CART	-	88.888±4.120	89.722±5.833	91.944±4.882	89.722±4.656	89.644±5.163	90.555±4.331	93.104±4.080	**100.000±0.000**	89.722±5.422	**93.611±4.312**
	NB	-	84.444±8.164	93.333±3.967	98.055±2.500	97.222±2.777	96.788±2.966	94.733±3.912	96.200±1.034	95.999±3.821	97.500±2.620	**98.777±2.991**
	SVM	-	41.944±9.658	45.000±9.765	45.000±9.765	40.277±1.119	40.711±6.813	41.899±6.742	44.133±2.580	**100.00±0.000**	41.944±1.020	**45.211±9.365**
Ye	LR	-	53.199±2.693	51.600±3.299	52.421±2.735	-	52.266±2.731	-	52.493±2.640	53.535±2.872	51.382±2.551	**54.835±2.943**
	LDA	-	59.267±1.010	56.991±1.594	57.111±1.544	-	57.611±3.212	-	57.845±1.855	57.544±2.818	58.656±1.774	**59.555±1.862**
	KNN	-	53.871±2.356	52.161±2.510	53.877±2.352	-	54.077±3.421	-	54.243±1.925	53.421±2.416	55.374±1.957	**56.988±1.800**
	CART	-	47.976±4.545	46.886±3.017	48.181±1.610	-	48.111±2.551	-	48.283±2.885	**99.991±0.001**	49.101±2.233	**49.300±3.505**
	NB	-	**56.222±1.683**	52.411±3.017	20.944±5.246	-	17.070±3.312	-	15.991±1.518	**57.166±2.966**	27.100±2.515	20.791±4.879
	SVM	-	**45.622±2.861**	42.655±2.929	42.691±3.594	-	42.844±2.714	-	42.696±3.594	**48.600±3.841**	40.541±3.371	42.800±3.544
WQ	LR	53.051± 1.582	51.969±1.309	52.261±1.349	52.923±1.451	-	53.145±1.360	-	51.644±1.472	51.396±1.431	52.791±1.321	**53.177±1.350**
	LDA	53.102± 1.369	51.928±1.653	52.580±1.525	53.010±1.425	-	53.212±1.221	-	52.033±1.472	52.200±1.409	53.371±1.492	**54.365±1.616**
	KNN	47.255± 0.812	46.398±1.050	46.511±1.237	47.322±0.830	-	47.525±0.944	-	46.820±0.944	**64.404±1.592**	46.398±1.001	**47.599±0.833**
	CART	58.316± 1.563	57.602±1.010	57.161±1.510	59.455±1.890	-	59.257±1.224	-	58.371±1.705	**99.800±0.191**	57.056±1.521	**59.557±0.610**
	NB	44.816± 1.463	43.500±3.602	41.791±4.045	44.825±1.514	-	44.814±1.441	-	41.671±2.210	**47.628±1.783**	44.876±1.399	**45.266±1.716**
	SVM	55.551± 1.078	54.326±1.084	54.356±0.900	56.267±1.580	-	55.681±0.900	-	55.601±1.399	**79.981±1.211**	54.314±1.111	**56.865±1.313**
MPG	LR	**76.702±3.430**	76.831±3.670	71.846±5.366	72.450±3.343	73.335±3.821	74.686±4.012	-	74.301±4.217	74.444±5.521	77.144±4.312	74.056±5.012
	LDA	74.055±3.932	74.814±3.644	71.974±7.917	72.781±5.496	73.963±4.621	74.433±3.981	-	74.556±4.124	72.701±5.324	73.077±3.853	**74.846±3.854**
	KNN	69.745±6.708	69.114±6.154	70.000±5.576	70.491±4.459	71.422±6.323	69.242±7.012	-	69.741±6.175	**79.860±2.424**	68.251±7.001	**72.196±4.744**
	CART	80.755±2.643	80.375±3.164	77.896±6.831	80.321±5.131	81.581±4.527	81.770±2.571	-	81.644±3.553	**100.000±0.000**	80.315±4.333	**82.276±3.300**
	NB	65.691±4.649	66.201±3.351	66.974±7.096	**67.377±6.737**	65.399±6.600	65.565±5.121	-	66.455±4.610	64.376±5.611	66.500±4.422	65.188±5.111
	SVM	65.822±5.400	65.691±5.443	63.682±6.006	61.801±6.308	65.711±6.001	65.822±5.201	-	65.821±5.201	**99.841±0.400**	65.555±6.333	**66.496±5.010**
Se	LR	92.856±4.312	91.900±3.814	91.952±3.625	92.644±4.988	93.399±4.000	92.610±4.001	-	84.523±1.111	92.250±6.775	92.611±4.321	**93.576±0.400**
	LDA	95.233±3.001	93.577±2.122	95.601±2.368	96.476±3.663	96.565±2.811	96.191±1.932	-	93.333±1.411	94.800±3.010	96.422±2.422	**96.611±0.401**
	KNN	89.765±4.210	89.281±2.111	89.750±3.414	89.111±4.583	90.233±4.171	86.900±3.201	-	80.231±1.077	**92.875±5.989**	90.000±4.378	**90.433±4.185**
	CART	89.765±3.070	89.044±3.511	92.431±4.012	90.000±5.277	91.194±5.011	89.766±2.666	-	85.474±3.000	**100.000±0.000**	91.900±3.873	**93.099±4.185**
	NB	91.421±5.311	52.611±17.701	87.566±5.622	90.290±5.777	91.190±5.363	89.044±3.410	-	88.800±1.578	90.524±4.396	91.424±5.345	**92.044±5.128**
	SVM	92.388±5.551	90.000±3.655	91.955±3.620	91.476±4.019	91.900±4.152	89.285±3.071	-	78.572±0.000	92.195±5.119	91.199±4.877	**92.446±4.130**
Cl	LR	58.000±5.066	55.511±7.893	56.033±8.169	55.161±5.921	57.166±5.918	56.041±6.417	53.265±3.151	52.166±1.674	56.819±5.428	57.833±6.666	**58.333±5.120**
	LDA	57.665±4.358	54.651±7.902	54.822±7.546	56.661±7.124	57.000±7.295	55.000±7.168	53.781±0.300	50.833±2.500	55.708±5.710	56.666±7.453	**57.867±5.775**
	KNN	49.000±4.229	53.440±9.348	52.581±7.642	54.331±5.112	55.335±6.112	53.541±5.594	51.911±2.114	40.833±2.713	**57.354±6.251**	53.666±6.699	**55.503±5.112**
	CART	41.831±6.767	52.411±8.382	51.552±7.533	52.000±5.652	52.000±6.227	50.833±6.194	51.883±8.242	47.000±3.785	**66.106±4.429**	52.463±7.000	**52.661±6.081**
	NB	53.832±7.418	52.751±8.239	53.275±7.766	56.661±5.941	57.333±5.587	57.916±5.951	53.665±0.200	51.666±1.490	56.631±6.774	57.500±6.677	**58.001±7.295**
	SVM	55.165±4.856	54.481±8.865	54.655±7.865	56.001±6.771	56.833±4.740	55.208±6.404	51.731±3.912	47.833±0.763	58.493±4.530	58.337±5.830	**58.516±4.273**
Oz	LR	96.027±0.342	96.351±0.922	96.405±1.081	96.486±0.841	-	96.486±0.849	-	96.135±0.383	95.514±3.581	96.405±1.081	**97.646±0.785**
	LDA	96.432±0.108	96.594±0.856	96.594±0.983	96.299±0.952	-	96.542±0.809	-	96.000±0.108	95.501±3.254	96.651±0.809	**96.677±0.863**
	KNN	96.652±0.000	96.729±0.883	96.726±0.879	96.729±0.883	-	96.756±0.879	-	95.945±0.000	94.537±2.808	96.677±0.883	**96.867±0.805**
	CART	94.540±4.491	94.945±0.640	94.594±1.226	94.402±1.451	-	94.081±1.293	-	94.216±0.675	**100.000±0.000**	94.544±1.260	**95.701±0.875**
	NB	71.405±0.264	**75.189±4.450**	61.378±13.695	71.831±2.843	-	71.378±2.843	-	66.945±0.725	64.045±10.513	72.243±3.017	70.297±2.956
	SVM	96.756±0.000	96.783±0.875	96.683±0.875	96.783±0.875	-	96.783±0.875	-	95.945±0.000	**100.000±0.000**	96.783±0.875	**96.856±0.871**
Pi	LR	75.777±4.401	70.641±2.451	75.841±3.841	74.700±3.335	75.911±3.074	75.972±4.353	-	73.244±0.984	75.199±3.071	76.033±4.341	**76.233±4.088**
	LDA	76.620±4.153	70.455±3.271	75.771±3.955	75.711±3.262	76.410±3.255	76.554±4.231	-	73.891±1.161	75.763±3.074	76.946±4.141	**77.335±4.484**
	KNN	72.721±3.969	71.234±3.263	72.885±3.671	71.515±3.813	70.833±3.008	72.141±3.205	-	73.866±1.991	**79.941±3.396**	72.201±4.363	**73.986±3.909**
	CART	67.010±2.596	69.091±3.941	66.880±4.385	66.555±5.163	66.500±3.885	66.491±2.912	-	66.940±3.017	**100.000±0.000**	68.375±3.832	**69.896±2.976**
	NB	76.441±4.739	72.923±2.767	75.453±5.334	75.457±3.371	75.911±1.878	76.686±4.839	-	75.191±0.962	75.757±1.841	76.422±4.545	**76.767±4.024**
	SVM	67.205±5.050	66.421±2.886	67.271±5.031	64.115±4.018	64.160±3.634	67.141±4.996	-	64.283±0.000	**99.825±0.563**	67.277±5.000	**67.476±4.972**
OD	LR	-	96.189±0.312	95.731±0.501	96.582±0.441	-	96.566±0.421	-	96.490±0.444	96.088±1.263	96.400±0.323	**96.766±0.891**
	LDA	-	94.351±0.744	94.500±0.981	95.101±9.862	-	95.155±0.965	-	95.022±0.800	94.265±1.469	95.085±1.060	**95.678±1.054**
	KNN	-	98.446±0.211	98.261±0.363	98.577±0.255	-	98.485±0.255	-	98.490±0.380	98.188±0.550	98.446±0.335	**98.665±0.463**
	CART	-	88.036±1.065	88.931±1.052	89.705±1.274	-	89.761±1.090	-	88.924±1.090	**100.000±0.000**	89.233±1.0756	**89.835±1.041**
	NB	-	88.291±0.721	90.776±1.263	91.346±0.943	-	91.074±0.631	-	91.645±0.880	91.712±2.181	90.712±0.924	**92.221±0.916**
	SVM	-	31.371±2.252	32.474±2.615	**44.235±6.931**	-	43.666±7.316	-	41.151±7.443	**100.000±0.000**	33.500±0.039	41.331±5.084
IS	LR	-	90.081±1.231	92.854±1.441	93.185±1.552	-	93.044±1.455	-	91.344±0.300	91.230±1.374	91.185±1.432	**93.404±2.019**
	LDA	-	90.575±1.353	91.040±1.344	91.400±1.444	-	91.264±1.223	-	91.931±0.841	91.499±1.332	91.233±1.254	**91.991±1.405**
	KNN	-	92.750±0.640	92.283±0.550	93.700±0.916	-	93.811±0.911	-	93.480±0.341	**96.480±1.356**	92.694±0.751	**93.885±1.106**
	CART	-	95.411±1.021	94.510±0.940	95.715±0.806	-	95.988±0.801	-	95.965±0.512	**99.865±0.231**	95.685±0.861	**96.222±1.106**
	NB	-	68.810±2.251	76.985±2.681	**80.905±2.501**	-	80.410±2.535	-	74.715±1.463	**82.993±3.313**	80.575±2.614	78.378±2.163
	SVM	-	48.655±3.312	50.777±3.613	55.240±4.591	-	54.251±4.122	-	49.530±1.060	99.811±0.210	51.313±3.121	**55.615±4.700**
PD	LR	92.600±0.301	92.101±0.300	90.933±0.512	92.813±0.231	-	92.811±0.313	-	92.000±0.216	91.103±0.641	**92.845±0.301**	89.866±0.803
	LDA	87.565±0.615	86.622±0.432	87.188±0.774	87.733±0.752	-	87.711±0.741	-	87.195±0.532	86.644±0.622	**87.744±0.514**	86.000±0.717
	KNN	99.222±0.141	99.090±0.232	99.100±0.114	99.235±0.202	-	99.202±0.244	-	99.145±0.261	99.500±0.116	99.091±0.202	**99.680±0.224**
	CART	95.631±0.431	95.022±0.441	95.411±0.406	95.800±0.391	-	95.753±0.375	-	95.642±0.363	**100.000±0.000**	95.631±0.514	**95.984±0.411**
	NB	86.151±0.833	79.983±0.116	85.955±0.737	86.167±0.906	-	86.125±0.981	-	86.641±0.736	85.343±1.070	85.855±0.902	**86.755±0.362**
	SVM	10.511±0.741	10.510±0.461	10.611±0.444	10.623±0.763	-	10.587±0.711	-	10.892±1.010	**100.000±0.000**	10.475±0.406	**11.646±1.064**

4.6. Result comparison

In this section, we evaluate the performance of different methods using 12 real data sets shown in Table 3. The average accuracy (in %) and the standard deviation of the imputation methods are reported in Table 4. In the table, the effects of all comparison methods are summarized and the best accuracy scores are represented with bold face red color. In Table 4 we can see that the performance of our imputation method is higher than that of the other imputation methods in most of the cases. It shows that the classification accuracy obtained on the data imputed by SISFCM is better than the one obtained on clean data, which means that the imputed values for missing data are more



(a) LR classifier performance

(b) LDA classifier performance

(c) KNN classifier performance

(d) CART classifier performance

(e) NB classifier performance

(f) SVM classifier performance

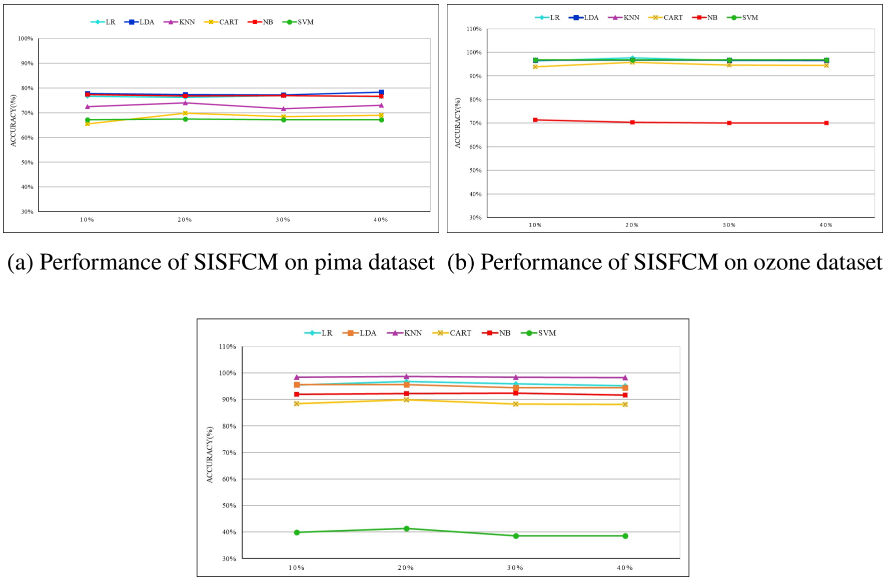
Figure 4: Performance analysis of six classifiers for nine state-of-the-art imputation methods in comparison with proposed SISFCM approach.

approximate to their actual values after using our new proposed shorter interval selection approach. It can be seen that SISFCM is effective in terms of missing value imputation not only for small data sets but also for large data sets. The effectiveness and feasibility of our approach can be verified with different missing ratios as shown in Fig. 5. We also conduct statistical analysis through Wilcoxon signed-rank test for identifying whether the degree to which our proposed approach outperforms each of the other ones is statistically significant and the results will be presented in Table 6.

The experimental results in Table 4 demonstrate that our method preforms sufficiently well for almost all the data sets across multiple classifiers. It can be observed in most of the cases for all classifiers our method shows better performance than the other methods. However, for some other classifiers trained on some data sets, the results show that our imputation method performs marginally worse. For example, the average accuracy of the LR classifier trained on the autoMPG data set imputed by the NNI method is a little bit higher than the one produced by SISFCM. On the other side, SISFCM performs better while using 5 out of 6 classifiers. Furthermore, the MICE method shows better performance of missing value imputation, while using NB and SVM to train classifiers on the ozone and yeast data sets, respectively. MissForest shows better performance on the wine data set while using KNN, but the use of NB leads to better performance of MissForest on the autoMPG, image segmentation and optdigits data sets and the use of SVM results in better performance on the optdigits data set.

Table 5
Accuracy for varying the percentages of missing values.

Data Set	Classifiers	10%	20%	30%	40%
Pima	LR	76.666±5.269	76.233±4.088	76.910±5.040	76.479±5.282
	LDA	77.723±5.395	77.335±4.484	77.235±5.538	78.292±4.851
	KNN	72.439±4.518	73.986±3.909	71.707±5.341	73.089±4.897
	CART	65.528±5.079	69.896±2.976	68.455±2.993	69.024±5.287
	NB±std	77.398±5.665	76.767±4.024	76.991±5.863	76.667±5.381
	SVM	67.154±5.383	67.476±4.972	67.154±5.383	67.235±5.417
Ozone	LR	96.418±1.059	97.646±0.785	96.520±1.100	96.587±0.523
	LDA	96.655±0.921	96.677±0.863	96.655±0.867	96.587±0.747
	KNN	96.824±0.710	96.867±0.805	96.790±0.679	96.790±0.679
	CART	93.851±1.435	95.701±0.875	94.662±1.168	94.425±1.607
	NB	71.418±2.551	70.290±2.956	70.033±2.878	70.000±2.841
	SVM	96.824±0.000	96.856±0.871	96.824±0.802	96.824±0.802
Optdigits	LR	95.667±0.731	96.766±0.891	95.347±0.754	95.204±0.694
	LDA	94.653±0.998	95.678±1.054	94.501±0.914	94.421±1.018
	KNN	98.345±0.323	98.665±0.463	98.354±0.225	98.282±0.221
	CART	88.371±1.109	89.835±1.041	88.300±1.297	88.194±1.057
	NB	91.921±0.807	92.221±0.916	92.451±0.945	91.628±0.968
	SVM	39.843±6.855	41.331±5.084	38.594±6.780	38.621±6.788



(a) Performance of SISFCM on pima dataset (b) Performance of SISFCM on ozone dataset

(c) Performance of SISFCM on optdigit dataset

Figure 5: Performance of SISFCM imputation method at different missing rates (10%,20%,30%,40%). The x-axis represents the missing ratio and y-axis represents the accuracy (%).

The main reason that those classifiers trained with the help of some other imputation methods perform better on some specific data sets is likely to be due to the suitability of some specific learning algorithms for the data sets pre-processed by specific imputation methods. For example, NB has the characteristic of the conditional independence assumption, when a data set holds this characteristic, the learning by NB converges more quickly, so fewer training instances are required and better performance can be obtained. On the other side, the LDA algorithm does not work well if the data distribution is not balanced (i.e. the sample sizes for various classes are different). Otherwise, it is more likely to work well. Similarly, KNN shows better performance when it selects an optimal number of neighbours, and

Table 6
Wilcoxon signed-ranks test for pairwise one tail comparison at p<0.005.

Methods	p-value (one tailed)	Null Hypothesis
NNI	0.0000	Reject
MICE	0.0042	Reject
Softimpute	0.0000	Reject
MissForest	0.0001	Reject
Matirx	0.0000	Reject
WKNNI	0.0000	Reject
SvrFcmGa	0.0012	Reject
SVD	0.0000	Reject
GAIN	0.0992	Accept
Clean data	0.0326	Reject

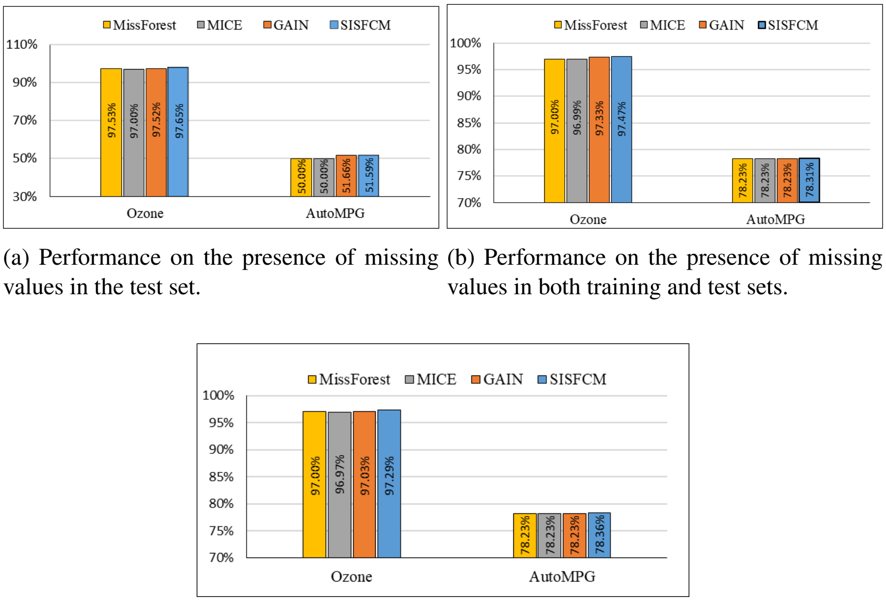
it is very sensitive to outliers, due to the selection of the nearest neighbours based on distance measures. Therefore, it is more probable that KNN is more suitable for training classifiers on the wine data set and performs better in missing value imputation. In addition, SVM works effectively when there is a clear separation margin between different classes of instances, but the performance may become worse when it is difficult to find a clear separation margin based on a specific distribution of different classes of instances.

The Matrix and SvrFcmGa methods are not suitable for large data sets. In the case of the matrix method, when we run it on CPU for large data sets, it appears to have the memory issue due to increasing the matrix size and the dimensionality. Furthermore, we run it on GPU Tesla V100-PCIE but we could not find it suitable for large data sets, so the Matrix method is considered more effective for small data sets. SvrFcmGa is more effective for a low percentage of missing values (1.5% of missing value), i.e., the presence of a high percentage (greater than 1.5%) of missing values results in the difficulty in meeting the error convergence criteria for SvrFcmGa. In particular, when the error convergence value is less than one, its error rate is not converging on some data sets either small or large. Therefore, this method is suitable for a small missing ratio (less than 20%) but with the increase of the missing ratio (greater than 1.5%) it does not work effectively anymore. However, the issue can be solved by increasing the error convergence value (if error>1 then the value is predicted). Table 4 depicts the data sets which are not used for evaluation with the corresponding methods (represented with '-'). Fig. 4 presents the graphical representation of our results more clearly and intuitively against different imputation methods with the use of six different algorithms for classifiers training. The x-axis represents the data sets, and the y-axis represents the average accuracy of each classifier obtained using these data sets.

In Table 4, the imputation performance of the GAIN method is marked in bold face blue color if GAIN performs the best. Also, the performance of our proposed method (SISFCM) is marked in bold face red color if SISFCM performs better than all the other methods apart from GAIN. In general, we can see that SISFCM outperforms GAIN on some data sets. However, while LR and LDA are used for training classifiers, SISFCM performs better than GAIN on the majority of these data sets. Moreover, on almost all the data sets, we can find some of the supervised learning algorithms that produce better performing classifiers on the data imputed by SISFCM than on the one imputed by GAIN. Considering that the GAIN method was designed in the setting of deep learning, SISFCM only involves learning strategies in a traditional setting but shows similarly competitive performance in comparison with GAIN, which indicates the effectiveness of the imputation strategies involved in SISFCM.

4.7. Statistical analysis

We used the Wilcoxon signed rank test to analyse the validity of the performance of the imputation methods, by comparing SISFCM with all the other methods separately at the statistical significance level p<0.05. Table 6 shows the pairwise comparison between SISFCM and each of the other imputation methods. The results indicate that our algorithm is significantly better than each of the other ones (except for the GAIN method).



(a) Performance on the presence of missing values in the test set.

(b) Performance on the presence of missing values in both training and test sets.

(c) Performance on the presence of missing values in the training set.

Figure 6: Performance of SISFCM imputation method with others on different cases. The x-axis represents the data sets and y-axis represents the accuracy (%).

4.8. Result analysis for all data sets

We can see in Table 5 the performance of our method with varying percentages of missing values in each data set and Fig. 5 represents a line graph to demonstrate the behaviour of SISFCM for some data sets (pima, ozone, optdigits) with different ratios of missing values, i.e., 10%, 20%, 30%, and 40%. We selected 3 different scales (small, medium and large) of data sets to show the SISFCM performance with varying missing ratios. The x-axis represents the percentage of missing values and the y-axis represents the accuracy (in %) against varying missing ratios. Fig. 5 depicts that our imputation technique has stable performance. The imputation accuracy does not vary significantly with the missing rate in the data set, and it indicates that with the increase of the missing rate in the data set the performance of this technique will not become worse.

Furthermore, we apply our method on two real-life data sets that involve real challenges, i.e., ozone and autoMPG (see Table 3), which have naturally missing values with the ratio of 27% and less than 10%, respectively. It brings challenges on judging whether the imputation of missing values is correct due to the fact that the true values (expected to be imputed) are unknown. The autoMPG data set consists of both continuous and discrete attributes, so in the pre-processing stage we removed the features having discrete values and involved only those continuous features in our experiment. The performance of SISFCM and three other methods (i.e., MissForest, MICE and GAIN) are evaluated on these two data sets (i.e., ozone and autoMPG) by assuming three possible cases of the natural presence of missing values in the experimental settings. In the first case, we put all the instances with missing values in the test set and train the KNN classifier on a clean training set for evaluating the imputation performance of each selected method on the test set (see Fig.6). Second, we randomly add each of the instances with missing values in either the training set or the test set, and the KNN classifier is trained for evaluating the imputation performance of various methods (see Fig.6). Thirdly, we put all the instances with missing values in the training set and train the KNN classifier for evaluating the performance of various imputation methods on the clean test set (see Fig.6).

Fig. 6 shows the results obtained for the three possible cases of the natural presence of missing values, which indicate that our method performs better than others on the two data sets, with an exception that the performance of our method is a little bit worse than that of the GAIN method on the autoMPG data set with the presence of missing values in the test set. Given that the autoMPG data set has all the missing values in the same column (feature), our method can directly handle such a situation without the need of other actions, but other methods do not work properly, i.e., they require at least one available value in the column. Therefore, for comparison of the experimental results, the value of zero is artificially added to fill in the missing values in that column for two randomly selected instances and the results are obtained in this setting for other methods. From Fig. 6, we can see that our method is suitable and effective in handling missing values naturally present in data sets and is very competitive to the other methods even when the values of a feature are missed for all instances.

Overall, the experimental results indicate that our method shows its effectiveness and strength in the case of natural presence of missing values in real world data sets, even when the ratio of missing values is high.

5. Conclusion

In this paper, we have proposed a shorter interval selection approach for estimating and imputing the missing values, which is driven by the Fuzzy C-Mean clustering technique. Our method has been evaluated on 12 well known data sets based on optimized parameters, i.e., the cluster size (the value of k) and the fuzzy parameter m are optimized according to the corresponding data set. The experimental results demonstrate that the effectiveness of missing value imputation can be improved through shortening the interval to which the missing value belongs. The experimental results also indicate that the proposed approach is robust to the change of the percentage of missing values by means of its stable performance on the same data set with different percentages of missing values. Moreover, our proposed approach shows its effectiveness of missing value imputation leading to satisfactory performance of classification, even for data with a high percentage of missing values, in comparison with the performance obtained using the clean data. We have also conducted statistical analysis through Wilcoxon signed-rank test for identifying the degree of the performance difference between our approach and each of the other ones. The results show that the proposed imputation method outperforms the other ones significantly. In the future, we will investigate how the proposed approach can be adjusted to impute effectively the missing values in the target column in the setting of semi-supervised learning and deep learning.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China, Guangdong province (No.2018A 0303130026) and National Natural Science Foundation of China (Grants 61976141 and 61732011).

References

[1] M. G. Rahman, M. Z. Islam, Missing value imputation using a fuzzy clustering-based em approach, Knowledge and Information Systems 46 (2) (2016) 389–422.

[2] H. Wang, S. Wang, Mining incomplete survey data through classification, Knowledge and information systems 24 (2) (2010) 221–233.

[3] R. J. Little, D. B. Rubin, Statistical analysis with missing data, Vol. 793, John Wiley & Sons, 2019.

[4] I. Myrtveit, E. Stensrud, U. H. Olsson, Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods, IEEE Transactions on Software Engineering 27 (11) (2001) 999–1013.

[5] K. Pelckmans, J. De Brabanter, J. A. Suykens, B. De Moor, Handling missing values in support vector machine classifiers, Neural Networks 18 (5-6) (2005) 684–692.

[6] W. Young, G. Weckman, W. Holland, A survey of methodologies for the treatment of missing values within datasets: Limitations and benefits, Theoretical Issues in Ergonomics Science 12 (1) (2011) 15–43.

[7] S. Nakagawa, R. P. Freckleton, Missing inaction: the dangers of ignoring missing data, Trends in ecology & evolution 23 (11) (2008) 592–596.

[8] I. B. Aydilek, A. Arslan, A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm, Information Sciences 233 (2013) 25–35.

[9] J. Yoon, J. Jordon, M. Van Der Schaar, Gain: Missing data imputation using generative adversarial nets, arXiv preprint arXiv:1806.02920 (2018).

[10] S. Zhang, Z. Jin, X. Zhu, Missing data imputation by utilizing information within incomplete instances, Journal of Systems and Software 84 (3) (2011) 452–459.

[11] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, Pattern classification with missing data: a review.

[12] G. Xu, J. Zhou, J. Dong, C. P. Chen, T. Zhang, L. Chen, S. Han, L. Wang, Y. Chen, Multivariate morphological reconstruction based fuzzy clustering with a weighting multi-channel guided image filter for color image segmentation, International Journal of Machine Learning and Cybernetics 11 (12) (2020) 2793–2806.

[13] A. K. Alok, P. Gupta, S. Saha, V. Sharma, Simultaneous feature selection and clustering of micro-array and rna-sequence gene expression data using multiobjective optimization, International Journal of Machine Learning and Cybernetics 11 (2020) 2541–2563.

[14] L. Sun, X. Qin, W. Ding, J. Xu, S. Zhang, Density peaks clustering based on k-nearest neighbors and self-recommendation, International Journal of Machine Learning and Cybernetics (2021) 1–26.

[15] D. Dua, C. Graff, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/, [Online; accessed 19-July-2020] (2020).
[16] X. Chen, Z. Wei, Z. Li, J. Liang, Y. Cai, B. Zhang, Ensemble correlation-based low-rank matrix completion with applications to traffic data imputation, Knowledge-Based Systems 132 (2017) 249–262.
[17] J. Huang, B. Mao, Y. Bai, T. Zhang, C. Miao, An integrated fuzzy c-means method for missing data imputation using taxi gps data, Sensors 20 (7) (2020) 1992.
[18] X. Lai, L. Zhang, X. Liu, Takagi-sugeno modeling of incomplete data for missing value imputation with the use of alternate learning, IEEE Access 8 (2020) 83633–83644.
[19] R. K. Bania, A. Halder, R-ensembler: A greedy rough set based ensemble attribute selection algorithm with knn imputation for classification of medical data, Computer Methods and Programs in Biomedicine 184 (2020) 105122.
[20] S. Zhang, Nearest neighbor selection for iteratively knn imputation, Journal of Systems and Software 85 (11) (2012) 2541–2552.
[21] L. Malan, C. M. Smuts, J. Baumgartner, C. Ricci, Missing data imputation via the expectation-maximization algorithm can improve principal component analysis aimed at deriving biomarker profiles and dietary patterns, Nutrition Research 75 (2020) 67–76.
[22] U. Garciarena, R. Santana, An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers, Expert Systems with Applications 89 (2017) 52–65.
[23] H. Shahbazi, S. Karimi, V. Hosseini, D. Yazgi, S. Torbatian, A novel regression imputation framework for tehran air pollution monitoring network using outputs from wrf and camx models, Atmospheric Environment 187 (2018) 24–33.
[24] Z. Qi, H. Wang, J. Li, H. Gao, Frog: Inference from knowledge base for missing value imputation, Knowledge-Based Systems 145 (2018) 77–90.
[25] C.-F. Tsai, M.-L. Li, W.-C. Lin, A class center based approach for missing value imputation, Knowledge-Based Systems 151 (2018) 124–135.
[26] A. M. Sefidian, N. Daneshpour, Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model, Expert Systems with Applications 115 (2019) 68–94.
[27] L. Zhang, W. Lu, X. Liu, W. Pedrycz, C. Zhong, Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values, Knowledge-Based Systems 99 (2016) 51–70.
[28] G. E. Batista, M. C. Monard, An analysis of four missing data treatment methods for supervised learning, Applied artificial intelligence 17 (5-6) (2003) 519–533.
[29] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R. B. Altman, Missing value estimation methods for dna microarrays, Bioinformatics 17 (6) (2001) 520–525.
[30] H. Glanz, L. Carvalho, An expectation–maximization algorithm for the matrix normal distribution with an application in remote sensing, Journal of Multivariate Analysis 167 (2018) 31–48.
[31] H. Jia, Z. Zhang, H. Liu, F. Dai, Y. Liu, J. Leng, An approach based on expectation-maximization algorithm for parameter estimation of lamb wave signals, Mechanical Systems and Signal Processing 120 (2019) 341–355.
[32] R. Razavi-Far, B. Cheng, M. Saif, M. Ahmadi, Similarity-learning information-fusion schemes for missing data imputation, Knowledge-Based Systems 187 (2020) 104805.
[33] M. Amiri, R. Jensen, Missing data imputation using fuzzy-rough methods, Neurocomputing 205 (2016) 152–164.
[34] P. D. Pantula, S. S. Miriyala, K. Mitra, An evolutionary neuro-fuzzy c-means clustering technique, Engineering Applications of Artificial Intelligence 89 (2020) 103435.
[35] L. Beretta, A. Santaniello, Nearest neighbor imputation algorithms: a critical evaluation, BMC medical informatics and decision making 16 (3) (2016) 74.
[36] X. Xu, W. Chong, S. Li, A. Arabo, J. Xiao, Miaec: Missing data imputation based on the evidence chain, IEEE Access 6 (2018) 12983–12992.
[37] R. Mazumder, T. Hastie, R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, The Journal of Machine Learning Research 11 (2010) 2287–2322.
[38] D. J. Stekhoven, P. Bühlmann, Missforest—non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (1) (2012) 112–118.
[39] E. J. Candès, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational mathematics 9 (6) (2009) 717.
[40] H. Lu, R. Yang, Z. Deng, Y. Zhang, G. Gao, R. Lan, Chinese image captioning via fuzzy attention-based densenet-bilstm, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 17 (1s) (2021) 1–18.
[41] H. Lu, M. Zhang, X. Xu, Y. Li, H. T. Shen, Deep fuzzy hashing network for efficient image retrieval, IEEE Transactions on Fuzzy Systems (2020).

Hufsa Khan is currently pursuing PhD degree in the School of Computing Science and Software Engineering, Shenzhen University,Guangdong,China under the supervision of Prof. Xizhao Wang. She received the BSc (Honors) and MSc degree in Computer Science from Lahore College For Women University, Lahore, Pakistan and The University of Lahore, Lahore, Pakistan, respectively. Her research interests include machine learning, missing data handling, data cleansing/preprocessing and semi supervised learning.

Xizhao WANG is a professor in Big Data Institute of ShenZhen University. He is a CAAI Fellow, an IEEE Fellow, the previous BoG member of IEEE SMC society, the chair of IEEE SMC Technical Committee on Computational Intelligence, the Chief Editor of Machine Learning and Cybernetics Journal, and associate editors for a couple of journals such as IEEE Transactions on Fuzzy Systems, on Cybernetics, etc.

Han Liu is currently an Associate Researcher in Machine Learning in the College of Computer Science and Software Engineering at the Shenzhen University. He has previously been a Research Associate in Data Science in the School of Computer Science and Informatics at the Cardiff University and has also been a Research Associate in Computational Intelligence in the School of Computing at the University of Portsmouth.