IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 888~898
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp888-898

A method for missing values imputation of machine learning datasets

Youssef Hanyf[1,2], Hassan Silkan[3]
[1]Research laboratory in management and decision support, AI Data SEED team, Ibn Zohr University, Dakhla, Morocco
[2] National School of Commerce and Management, Ibn Zohr University, Dakhla, Morocco
[3]LAROSERI laboratory, Faculty of Sciences, Chouaib Doukali University, EL Jadida, Morocco

Article Info

Article history:

Received Apr 7, 2023
Revised Sep 30, 2023
Accepted Nov 15, 2023

Keywords:

Imputation
Machine learning accuracy
Imputation cost
Missing data

ABSTRACT

In machine learning applications, handling missing data is often required in the pre-processing phase of datasets to train and test models. The class center missing value imputation (CCMVI) is among the best imputation literature methods in terms of prediction accuracy and computing cost. The main drawback of this method is that it is inadequate for test datasets as long as it uses class centers to impute incomplete instances because their classes should be assumed as unknown in real-world classification situations. This work aims to extend the CCMVI method to handle missing values of test datasets. To this end, we propose three techniques: the first technique combines the CCMVI with other literature methods, the second technique imputes incomplete test instances based on their nearest class center, and the third technique uses the mean of centers of classes. The comparison of classification accuracies shows that the second and third proposed techniques ensure accuracy close to that of the combination of CCMVI with literature imputation methods, namely k-nearest neighbors (KNN) and mean methods. Moreover, they significantly decrease the time and memory space required for imputing test datasets.

Corresponding Author:

Youssef Hanyf
National School of Commerce and Management of Dakhla, Ibn Zohr University
Dakhla, Morocco
Email: Youssef.hanyf@gmail.com; y.hanyf@uiz.ac.ma

1. INTRODUCTION

In the last decade, machine-learning classification methods have become increasingly required and used in various outstanding technologies such as health care [1], social media, and recommendation systems. In consequence, many machine-learning-related problems have attracted the attention of a large community of researchers. Handling missing data is one of the most severe problems of machine-learning classification because it significantly affects classification accuracy [2], [3]. Although the increasing development of data collection and acquisition technologies, various reasons can lead to losing values in datasets like the breakdown of devices, power cuts, and unanswered form questions [4]. Therefore, datasets often require a preprocessing phase to impute the missing values before training and testing classification models.

The intuitive way to deal with missing data is the deletion of the features or instances containing missing values [5]–[7]. However, this method has risks of losing important information in datasets, and it can significantly impact classification accuracy. Many other methods have been used and proposed in the literature to impute missing data for increasing classification accuracy [8], [9]. These methods can be classified into two principal categories; statistical-based methods, such as mean/mod and least squares (LS), and machine-learning-based methods like k-nearest neighbors (KNN), neural networks (NN), and decision tree (DT) [10].

Hoque *et al.* [5] have compared the imputation accuracy of many machine-learning-based methods. They found that adaboost classifier and linear support vector machine (SVM) are better than logistic regression (LR), and random forest (RF). But this study has been carried out only on one dataset. Thus, these results need to be validated in other datasets.

The machine-learning-based imputation methods are better than statistical methods regarding classification accuracy. But they are computationally expensive due to the training cost of models for each feature that contains missing values [11], [12]. Nevertheless, the statistical imputation methods remain widely used in practice, especially for massive and high-dimensional datasets when the imputation becomes very expensive. Two recent trends appeared for improving the classical imputation methods; hybrid methods and multi-imputation methods. Hybrid methods aim to optimize the trade-off between the imputation cost and the classification accuracy by combining the statistical and machine learning approaches [11]–[13]. Multiple imputation methods aim to increase the imputation's accuracy by imputing missing values with many estimated values and keeping the one that achieves the best accuracy [14].

The class center missing values imputation (CCMVI) is among the hybrid methods that have shown an inexpensive imputation cost and good accuracy at the same time. It computes the center of each class, and then it imputes missing values of each incomplete instance based on its class center. The main drawback of the CCMVI method is that it is applicable just for instances of known classes, whereas that is not the case in real-world applications of machine learning models. Consequently, the employment of the CCMVI method after the model deployment in real-world applications is not possible. As well, instances of test datasets should be assumed unknown to simulate the usage of the model in real-world applications to ensure the credibility of the performance evaluation results. Thus, the CCMVI method can successfully impute missing values of training datasets, but it is not appropriate for test datasets because classes of instances should be assumed unknown to employ the same imputation method that we can use after model deployment.

In this work, we propose an imputation method (CCMVI+) that extends the CCMVI method to handle missing values in test datasets. On one side, we combine the CCMVI method to impute training datasets with statistical or machine-learning-based imputation methods, such as KNN and mean, to impute test datasets. On the other side, we propose two new techniques for datasets missing values imputation based on classes' centers determined in the training dataset imputation. One technique, called the nearest class center method (N_CC), imputes the missing values of instances based on their nearest neighbor among classes' centers. While, the other technique, called in this paper the class centers mean (CC-Mean), is based on the mean of classes' centers. Thus, there are three possible versions of the proposed method by combining the CCMVI with the proposed techniques, namely CCMVI+a_literature_method, CCMVI+N_CC, and CCMVI+CC-Mean. We evaluated the accuracy and the computation time of the proposed method on six datasets of different sizes, dimensionality, and number of classes. Thus, we compared the accuracies of state-of-art classification models trained and tested on datasets imputed by the proposed method. We also evaluated the computation time consumed by the proposed method in test datasets.

The rest of this paper is organized as follows. Section 2 reviews related literature and describes the CCMVI method, and section 3 presents the proposed method. The research method is presented in section 4. While, the results are presented and analyzed in section 5, and section 6 concludes the paper.

2. RELATED WORK

2.1. Methods of missing values imputation

Many works in the literature [11]–[13], [15] categorize imputation methods into two categories: statistical methods and machine-learning-based methods. The Mean method is considered among the simplest statistical method; it replaces each missing value with the average value of the corresponding feature. Other more advanced statistical methods have been proposed in the literature, like the expectation-maximization method (EM) [16], the LR method [17], and the least square method (LS) [18]. Machine-learning-based methods create a model for each feature that contains missing values. The feature of the missing value is regarded as the target of classification/regression models trained based on the remaining data. They predict each missing value by using the corresponding classification/regression model. Among many machine-learning-based imputation methods, SVM, DT [19], KNN [20], and RF are the most popular in the literature [21].

Osman *et al.* [15] categorize the imputation methods into traditional and modern methods. The traditional methods category includes techniques of missing data deletion, and some methods of single imputation such as mean, hot-dock, cold-dock, and regression imputation. The modern methods category contains the multiple-imputation-based methods which analyze multiple imputation choices and adopt the best one, EM method, and machine-learning methods such as KNN [20], [22], NN [23] and DT methods [19].

2.2. CCMVI method

The main idea behind the CCMVI method is to use the class center to impute the missing values of the incomplete instances, belonging to that class. This method consists of two phases. In the first phase, the method determines the center and the threshold of each class. To determine a class center, it calculates the average of the complete instances belonging to the class. On the other hand, to determine the threshold of a class, the method calculates the average of the distance values between the center and the other complete instances of the class.

The second phase utilizes previously computed centers, thresholds, and standard deviation for imputing incompletes instances. It differentiates between two types: instances with only one missing value and instances with many missing values. For instance of the first type, if the distance d(I,C) between an instance I and its class center C exceeds the class threshold, the missing values of I are replaced with the corresponding attributes of its class center C. Otherwise, the method imputes missing values with the sum or difference of the corresponding center attributes and the class standard deviation. Whereas in the case of instances with multiple missing values, the missing values of the instance are imputed with the corresponding attributes of the class center if the distance between the class center and the instance exceeds the threshold. If not, the method generates many imputation propositions by incrementing or decrementing missing values based on the corresponding attributes of the standard deviation. The chosen proposition for the final imputation is the one that minimizes the distance between the center and the instance.

Experiments in [11], [13] have demonstrated that The CCMVI method delivers good classification accuracy for numerical and mixed datasets. It outperforms other methods such as SVM and KNN. Furthermore, CCMVI is a fast imputation method as it calculates the center, the standard deviation, and the threshold of each class just once. Consequently, the imputation of a missing value requires only the identification of the corresponding center.

3. PROPOSED METHOD

The proposed method, CCMVI+, differentiates between training datasets and test datasets. It imputes incomplete instances of training datasets following the same approach as the CCMVI method. However, for imputing test datasets, we propose three different techniques that do not require the identification of the instance class. These techniques can also handle incomplete instances in real-world machine-learning applications. Thus, the CCMVI+ method consists of two parts. The first part deals with training datasets where the classes of instances are known, and the second part handles instances with unknown or assumed unknown classes, mainly test datasets and instances of real-world applications that use machine-learning models. All imputation methods of literature are applicable in the second part. Furthermore, we proposed two algorithms specifically designed for this purpose, which are described in the sub-sections bellow. Figure 1 presents a high-level flow diagram illustrating the CCMVI+ process for imputing data for training, testing, and using machine-learning classification models.

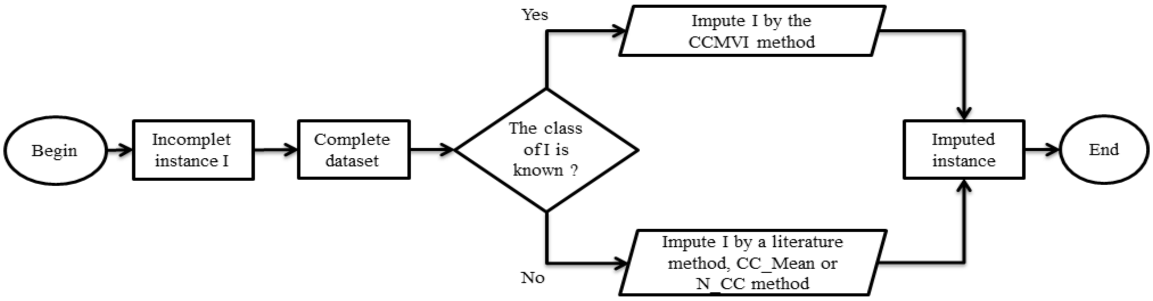
Begin
Incomplet instance I
Complete dataset
The class of I is known ?
Yes
Impute I by the CCMVI method
No
Impute I by a literature method, CC_Mean or N_CC method
Imputed instance
End

Figure 1. High-level diagram flow of the CCMVI+ method imputing process

3.1. Imputation of training datasets

The first part of CCMVI+ uses the same algorithm as the classical CCMVI method [11] to impute training datasets. It starts by identifying the center, the standard deviation, and the threshold of each class. To this end, the algorithm divides the datasets into two groups; complete data and incomplete data. Next, the algorithm calculates the average (*Avg(i))* and the standard deviation (*Std(i))* of each class i. Finally, the threshold of each class i is determined by computing the mean of distances between the class center (*Avg(i)*) and the complete instances of the class. Formally, the threshold a class i is calculated by using the following formula:

$$threshold(i) = \frac{1}{n}\sum_{k=1}^{n} Euclidian\_distance(I_{complete}(k), avg(i)) \quad (1)$$

Such as n is the number of complete instances in the class i and $I_{complete}(k)$ is the kth complete instance of class i. Next, the algorithm distinguishes between two cases to impute the missing values in incomplete instances of each class. The first case is when the instance contains just one missing value. The algorithm, in this case, replaces the missing j^{th} attribute of an incomplete instance $I_{incomplete}$ of class i by the jth attribute of the already calculated average (center) *Avg(i)* of class i; $I_{incomplete}(j)= Avg(i,j)$. Then, if the Euclidian distance between the instance $I_{incomplete}$ and the average of the class *Avg(i)* is superior to the class threshold, the algorithm decreases/increases the missing value by the corresponding attribute of the class standard deviation as follows:

$$I_{incomplete}\,(j) = I_{incomplete}\,(j) + std(i,j)\ \boldsymbol{Or}\ I_{incomplete}\,(j) = I_{incomplete}\,(j) - std(i,j) \quad (2)$$

The second case is when the instance contains more than one missing value. In this case, the algorithm replaces each missing attribute of the incomplete instance $I_{incomplete}$. If the Euclidean distance between the instance and the class center exceeds the class threshold, the algorithm increases or decreases missing values by using the corresponding attributes of the class standard deviation and recalculates the distance to the class center. Finally, the algorithm adopts the instance that minimizes the distance to the center.

Algoritm 1: N_CC Imputation

```
Input:
    C: set of classes'centers
    $I_{incomplete}$ : incomplete instance
Output:
$I_{complete}$: complete instance
Begin
    k=card(C)
    For i=0 to card($I_{incomplete}$)
        if $I_{incomplete}$ (i) is missing value then
            $I_{incomplete}$(i) ← 0
            Add i to the index array of missing values arr_index
        end
    end
    dmin←∞
    For i=0 to k
        C_copy=C(i)
        C_copy(arr_index)=0
        if Euc_distance($I_{incomplete}$, C_copy (i))<dmin then
            NN←C(i)
        end
    end
    $I_{complete}$ = $I_{incomplete}$
    For each i in arr_index
        $I_{complete}$(i)← NN(i)
    end
end
```

3.2. Imputation of test datasets

The second part aims to impute test datasets and instances in real-world machine-learning applications where instances classes are unknown. We propose three techniques for this part. The first technique involves utilizing literature imputation methods such as KNN and Mean to impute incomplete instances. However, this technique may be computationally expensive, especially in the case of using accurate imputation methods for test datasets, because it wastes centers that are computed in the first part.

Besides, we propose two low computational cost techniques based on previously computed centers in messing values imputing in the first part of the CCMVI+ method. The proposed techniques allow a significant reduction of the imputation cost because they deal only with pre-computed centers instead of all training data since they are representatives of all the complete instances of training datasets. One technique identifies the nearest class center, while the other computes the mean of the centers for imputing incomplete instances.

3.2.1. Nearest class center technique

The technique of N_CC imputes the missing values of an instance by using its nearest neighbor among the centers of classes based on Euclidian distance. The Euclidean distance computation between an incomplete instance and centers requires impermanent handling of missing values. Two possible ways to handle the missing values before distance computation; deleting attributes of missing values or replacing the missing values with a constant value. The proposed technique substitutes missing values in incomplete instances and corresponding values in centers with zeros for computing distances (lines 4, 10, and 11 of algorithm 1), which is equivalent to deleting missing values attributes.

Algorithm 1 provides the pseudocode of the proposed technique to impute an incomplete instance Iincomplete based on a set, C, of classes' centers. The algorithm defines the nearest neighbor of the incomplete instance among the classes' centers. To this end, it replaces the missing values with zeros and gets their indexes (lines 2-7). Then, it calculates the Euclidean distances between the instances and centers to identify the nearest neighbor (lines 9-14). Finally, the algorithm replaces the instance missing values with the values of the corresponding attributes of the identified nearest center (lines 16, 17, and 18). Figure 2 illustrates an example of using the nearest center technique for imputing an incomplete instance based on a six-dimensional and four classes dataset.

3.2.2. Mean of classes centers technique

The mean of CC_Mean technique is similar to the classical mean method, but it is computationally less expensive. Instead of the calculation of the average of all data, this proposed technique computes the mean based on classes' centers. For balanced datasets, the average of classes' centers is equal to the average of all instances. Formally, if a dataset contains N complete instances and Nc classes, and |class 1| ≈ |class 2| ≈…≈ |class Nc|, then $\frac{1}{N}\sum_{i=1}^{N} I_{complete}(i) \approx \frac{1}{N_c}\sum_{i=1}^{N_c} Center(i)$, such as $Center(i)$ is the average of the complete instances of class i. Usually, the sizes of classes of training datasets are relatively balanced to achieve good classification accuracy. Thus, one can expect that the CC_mean imputation method will achieve accuracy similar to those obtained by the classical mean imputation method.

Algoritm 2: CC_Mean Imputation

```
Input:
    C: set of classes'centers
    $I_{incomplete}$ : incomplete instance
Output:
    $I_{complete}$: complete instance
Begin
    avg_centers=Average(C)
    $I_{complete} = I_{incomplete}$
    For i=0 to card($I_{incomplete}$)
        if $I_{incomplete}$ (i) is a missing value then
            $I_{complete}$ (i)← avg_centers (i)
        end
    end
end
```

The algorithm of this method (see the pseudocode in algorithm 2) calculates the average of classes' centers. Then it replaces the missing values of the incomplete instance with the corresponding attributes of the Mean of centers. Figure 3 illustrates an example of using the CC_Mean technique for imputing missing values.

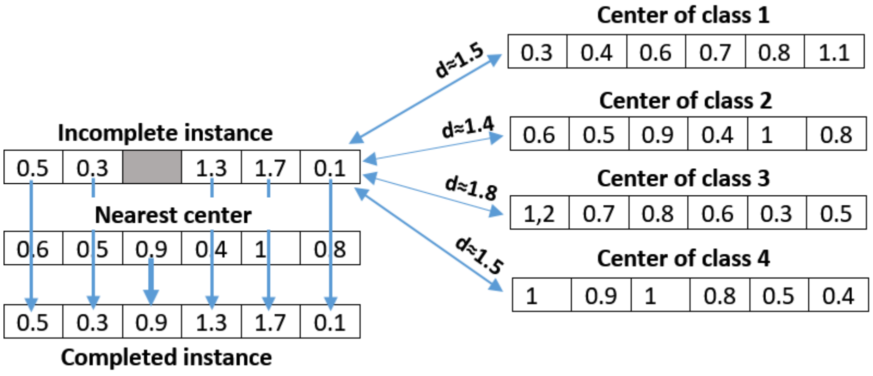

Figure 2. Example of imputing a missing value by using N_CC method

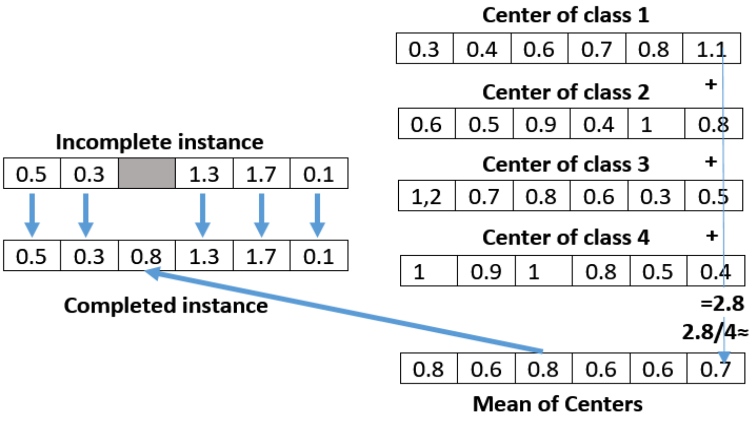

Figure 3. Example of imputing using CC_Mean method

4. RESEARCH METHOD

4.1. Datasets

We used in the experiments six numerical datasets with different numbers of instances, features, and classes to evaluate the performance of the proposed imputation techniques. The classes' numbers are between 2 and 11, the features' number ranges between 11 and 64, and the instances' number ranges between 208 and 19020 in Table 1. The distribution of the Euclidian distance from the centroid of each dataset is presented in Figure. All datasets are downloaded from the University of California Irvine Machine learning repository, except the Texture Dataset which is downloaded from the OpenML repository [24].

Table 1. Datasets characteristics

Datasets	N°. of instances	N°. of features	N°. classes
Sonar [25]	208	60	2
Magic [26]	19,020	11	2
Wall-robot-navigation [27]	5,456	25	4
Segment [28]	2,310	19	7
Optdigits [29]	5,620	64	10
Texture [24]	5,500	41	11

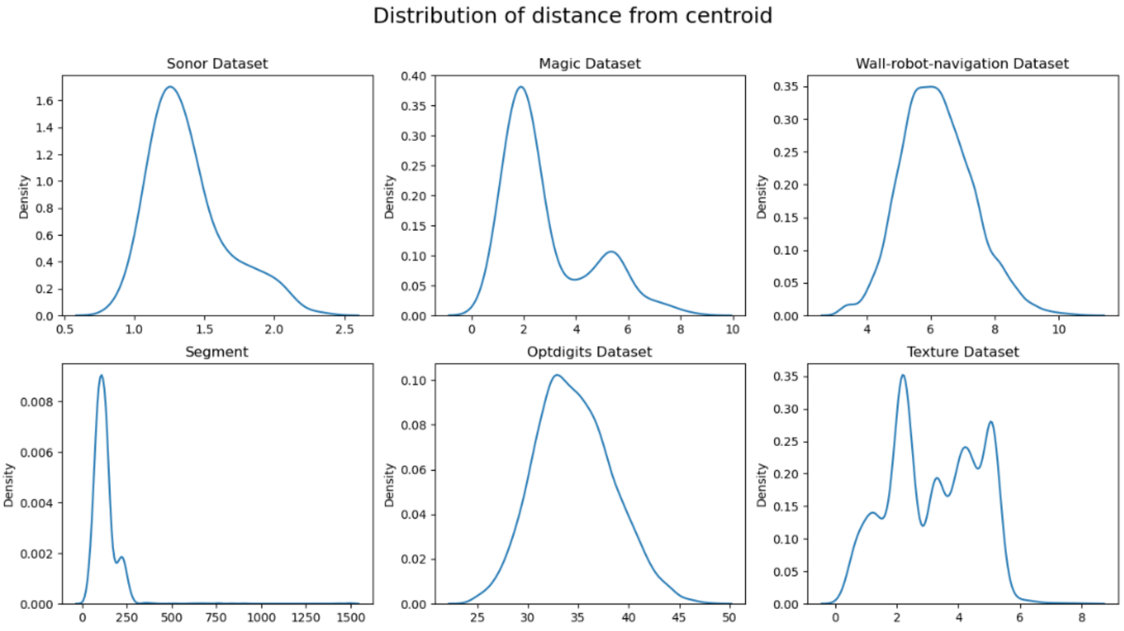

Figure 4. Distribution of Euclidean distances from centroid of experiments datasets

4.2. Experimental design

We evaluated the efficiency of the proposed imputation method by comparing of the accuracies of classification models on datasets imputed by using different proposed methods. We used three machine-learning methods for classification; extreme gradient boosting (XGBOOST), KNN, and multi-layer perceptron (MLP) Classifier. We randomly split the used dataset into a training dataset (75%) and a test dataset (25%) for training and testing each model. For each dataset, the accuracy adopted in the evaluation is the average of the accuracies of these three classification methods.

We performed the experiments on many missing rates of each dataset, namely 10%, 20%, 40%, and 60%. We repeated each experiment 20 times with a random choice of missing values and random splitting of data. The adopted accuracy and computation time is the average of scores obtained in 20 experiences.

We compared the classification accuracies on datasets imputed in five different ways. The first three methods (CCMVI+Constant, CCMVI+Mean, and CCMVI+KNN) combine the CCMVI method for training datasets imputing and the classical imputation techniques such as Constant, the Mean, and the KNN method for test datasets imputing. The remaining two methods (CCMVI+N_CC and CCMVI+CC_Mean) combine the CCMVI method with the proposed N_CC and CC_Mean techniques. We also compared the imputation cost of the proposed methods. Since the cost of training datasets imputing is the same for all used methods, the cost comparison is carried out only on test datasets.

5. RESULTS AND DISCUSSION

5.1. Evaluation of the imputation impact on the classification accuracies

Table 2 shows the accuracy results of the proposed methods on all datasets and all missing rates. We can see that the CCMVI+KNN imputation gives the best average accuracy of all datasets and all missing rates. While the nearest the CCMVI+N_CC method gives the second-best accuracy. We can also observe that the CCMVI+CC_Mean average accuracy approximates that of the CCMVI+Mean. The difference between the average accuracy of CCMVI+Mean and that of CCMVI+CC_Mean imputation is insignificant (0.01). The reason is that the Mean of centers tends to approach the Mean of instances when datasets are nearly balanced, which is the case for the majority of datasets used in experiments. We can also notice that proposed CCMVI+CC_Mean and CCMVI+N_CC are best than CCMVI+constant imputation in terms of accuracy.

Table 2. Average classification accuracies

	Constant	Mean	KNN	CC_Mean	C_NN
Average accuracies	0.653840	0.724957	**0.791701**	0.711585	0.761717

Although the superiority of the CCMVI+KNN accuracy, the average difference between the CCMVI+KNN accuracy with that of the proposed CCMVI+N_CC is insignificant (0.03). The average accuracy of the proposed CCMVI+N_CC outperforms the CCMVI+KNN in the Optdigit dataset (see Table 3) and outperforms all other methods in high missing rates of Optdigit and Segment datasets. We can conclude that the CCMVI+N_CC method gives their best accuracies in datasets of high missing rates and high numbers of classes (Optidigits, Texture, and Segment).

Figure 5 presents the accuracies averages of each method per missing rates on all datasets. For small missing rates (10%), CCMVI+Mean and proposed CCMVI+CC_Mean techniques give almost the same accuracy. Whereas the CCMVI+KNN method performs all others, the difference with the CCMVI+N_CC significantly decreases when we increase the missing rates (40%). Figures 6 to 11 respectively represent the classification accuracies on Sonar, Magic, Wall-robot navigation, Segment, Optdigit, and Texture datasets. The results confirm that the CCMVI+KNN method is more accurate than the other methods in the majority of cases.

Table 3. Average classification accuracies per dataset

	Constant	Mean	KNN	**CC_Mean**	**N_CC**
Sonar	0.646314	0.747997	**0.754679**	0.746715	0.738061
Magic	0.709407	0.728716	**0.731476**	0.667854	0.709911
Wall-robot-navigation	0.635945	0.745595	**0.783822**	0.725406	0.691603
Segment	0.634011	0.653085	**0.771035**	0.655161	0.742085
Optdigits	0.752186	0.715801	0.798746	0.753256	**0.831186**
Texture	0.581564	0.722161	**0.910445**	0.721118	0.857458

5.2. Evaluation of the imputation cost

To evaluate the computation cost of different methods used for imputing test datasets in experiments, we compared the computation times consumed by imputation methods. Figure 12 shows the results of imputation times on each dataset, and Table 4 shows the average imputation time on all datasets. The used missing rate of each dataset is 60%, and all imputations are carried out in the computational environment consisting of Intel(R) Xeon(R) CPU @ 2.20GHz, 2 CPU cores, and 18GB RAM.

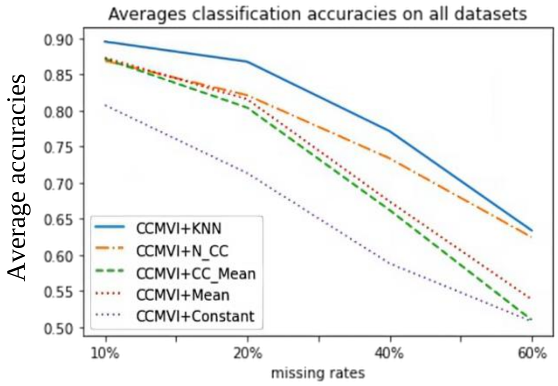

Figure 5. Average accuracies per missing rate

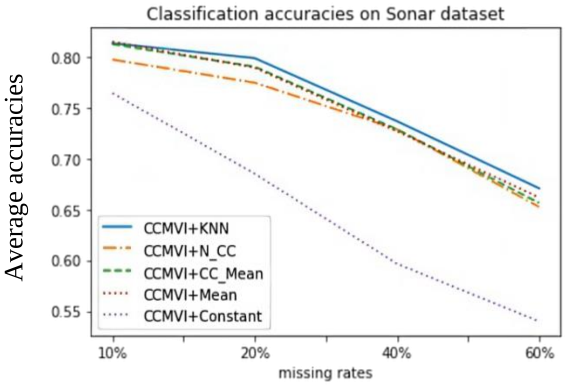

Figure 6. Classification accuracies on Sonar dataset

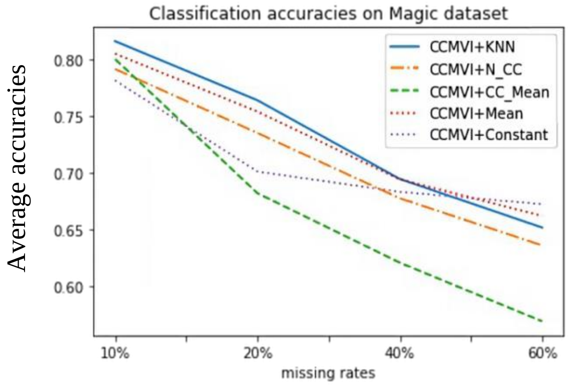

Figure 7. Classification accuracies on Magic dataset

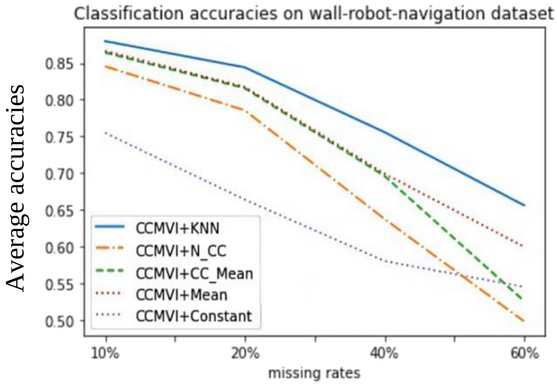

Figure 8. Classification accuracies on Wall-robot navigation dataset

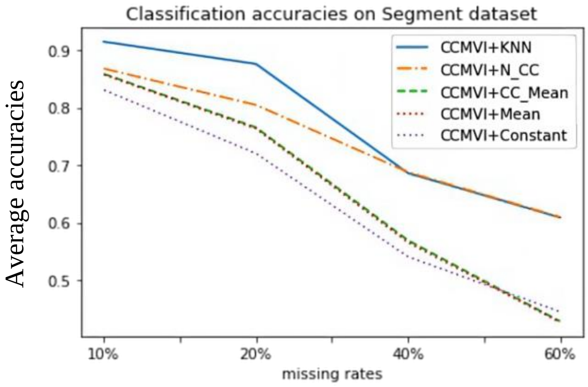

Figure 9. Classification accuracies on segment dataset

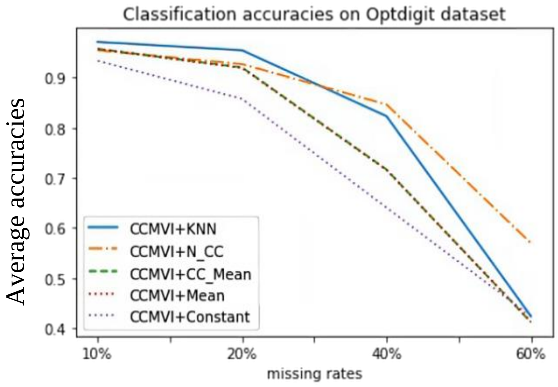

Figure 10. Classification accuracies on optdigits dataset

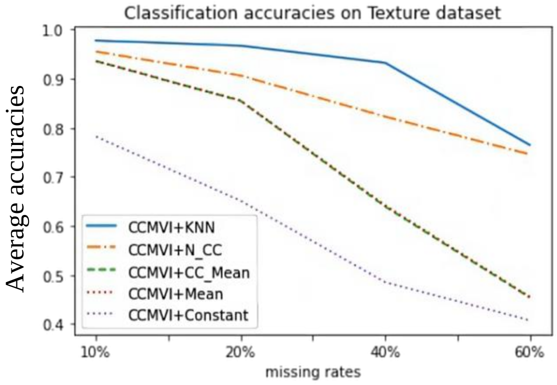

Figure 11. Classification accuracies on texture dataset

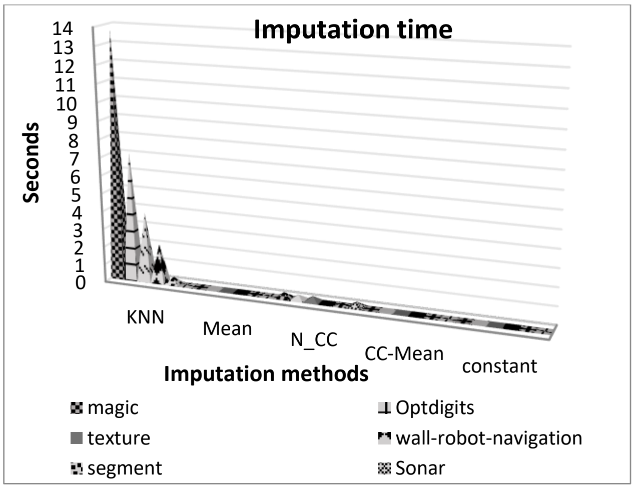

Figure 12. Computation time of different imputation methods

The constant, the Mean, and the CC_Mean methods are significantly faster than the KNN method. The complexity of computing the average of instances in the Mean method is *O(n*m)*, where n is the instances number and m is the features number. On the other hand, the CC_Mean complexity of computing the average of centers is *O(k*m)*, where k is the number of classes. Although the computation of the average by CC_Mean is significantly less expensive than that computed by the classical Mean, one can observe that there is no significant impact on the total imputation time because the computation of the average is done just once to impute all instances.

In these experiments, the proposed N_CC imputation method is 20 times faster than the KNN imputation method. The naïve KNN algorithm requires O(n) distance computation between the incomplete instance and the training datasets instances. The use of data structures such as kd-tree or Ball-tree can reduce the complexity of the KNN imputation to approximately O(log n) distance computations [20], [22], [30]–[35]. While, the naïve N_CC requires only O(k) distance computations, such as k is the number of classes.

Table 4. Average imputation time of different methods

KNN	Mean	Constant	CC_Mean	N_CC
4,5595	0,0009	0,0005	0,0007	0,2288

6. CONCLUSION

In this work, we proposed an extension of the CCMVI imputation method, called CCMVI+, to handle missing values of test datasets. The CCMVI+ method uses the classical CCMVI to impute training datasets and provides three possible techniques for test datasets imputing. The first proposed technique combines the CCMVI with literature imputation methods for test datasets. The second technique identifies the nearest class center for test datasets imputing, whereas the third technique computes the Mean of classes' centers for test datasets imputing. In the experiment, we compared classification accuracies of machine learning methods on datasets imputed by methods that use the proposed techniques, namely CCMVI+Constant, CCMVI+Mean, CCMVI+KNN, CCMVI+CC_Mean, and CCMVI+N_CC imputation methods. The results show that the combination between CCMVI and KNN outperforms the other methods and that the proposed CCMVI+N_CC is the second-best choice in terms of classification accuracy. The results show also that the difference between the proposed second technique and the combination between KNN and N_CC becomes less significant in high missing rates. Moreover, the difference between the proposed technique based on the Mean of classes' centers (CC_Mean) and the CCMVI+Mean is highly insignificant. We also compared the computation time of the proposed methods. The results show that KNN is the most computationally expensive imputation method compared with N_CC and CC_Mean. The computation time of CC_Mean and classical Mean is approximately the same. Thus, the accuracy of the CCMVI+N_CC method is near to that of CCMVI+KNN, and it is significantly less expensive because they treat only classes' centers instead of all datasets. The proposed CCMVI+N_CC and CCMVI+CC_Mean methods significantly save the prediction time and memory space without a high impact on the accuracy, especially for high dimensional, large, and high missing rates datasets.

ACKNOWLEDGEMENTS

We acknowledge the invaluable contribution to this research by the AI Data SEED team belonging to the Research Laboratory in Management and Decision Support. Our gratitude extends to Ibn Zohr University and ENCG of Dakhla for providing essential resources and a conducive research environment.

REFERENCES

[1] D. P. Javale and S. S. Desai, "Machine learning ensemble approach for healthcare data analytics," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, pp. 926–933, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp926-933.

[2] N. H. A. Rahman and M. H. Lee, "Artificial neural network forecasting performance with missing value imputations," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 1, pp. 33–39, Mar. 2020, doi: 10.11591/ijai.v9.i1.pp33-39.

[3] H. A. Saleh, R. A. Sattar, E. M. H. Saeed, and D. S. Abdul-Zahra, "Hybrid features selection method using random forest and meerkat clan algorithm," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 5, pp. 1046–1054, Oct. 2022, doi: 10.12928/TELKOMNIKA.v20i5.23515.

[4] A. Mirzaei, S. R. Carter, A. E. Patanwala, and C. R. Schneider, "Missing data in surveys: key concepts, approaches, and applications," *Research in Social and Administrative Pharmacy*, vol. 18, no. 2, pp. 2308–2316, Feb. 2022, doi: 10.1016/j.sapharm.2021.03.009.

[5] J. M. Z. Hoque, J. Hossen, S. Sayeed, K. Chy Mohammed Tawsif, J. Ganesan, and J. Emerson Raja, "Automatic missing value imputation for cleaning phase of diabetic s readmission prediction model," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 2001–2013, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2001-2013.

[6] A. Desiani, S. Yahdin, A. Kartikasari, and Irmeilyana, "Handling the imbalanced data with missing value elimination smote in the classification of the relevance education background with graduates employment," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 346–354, Jun. 2021, doi: 10.11591/ijai.v10.i2.pp346-354.

[7] F. Ahmad and S. A. M. Rizvi, "Identification of user's credibility on twitter social networks," *Indonesian Journal of Electrical Engineering and Computer Scienc*e, vol. 24, no. 1, pp. 554–563, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp554-563.

[8] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *Journal of Big Data*, vol. 8, no. 1, p. 140, Oct. 2021, doi: 10.1186/s40537-021-00516-9.

[9] P. C. Chiu, A. Selamat, O. Krejcar, K. K. Kuok, S. D. A. Bujang, and H. Fujita, "Missing value imputation designs and methods of nature-inspired metaheuristic techniques: a systematic review," *IEEE Access*, vol. 10, pp. 61544–61566, 2022, doi: 10.1109/ACCESS.2022.3172319.

[10] P. J. García-Laencina, J. L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, Mar. 2010, doi: 10.1007/s00521-009-0295-6.

[11] C. F. Tsai, M. L. Li, and W. C. Lin, "A class center based approach for missing value imputation," Kn*owledge-Based Systems*, vol. 151, pp. 124–135, Jul. 2018, doi: 10.1016/j.knosys.2018.03.026.

[12] I. B. Aydilek and 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*, vol. 233, pp. 25–35, Jun. 2013, doi: 10.1016/j.ins.2013.01.021.

[13] H. Nugroho, N. P. Utama, and K. Surendro, "Class center-based firefly algorithm for handling missing data," *Journal of Big Data*, vol. 8, no. 1, p. 37, Dec. 2021, doi: 10.1186/s40537-021-00424-y.

[14] G. S. Hassan, N. J. Ali, A. K. Abdulsahib, F. J. Mohammed, and H. M. Gheni, "A missing data imputation method based on salp swarm algorithm for diabetes disease," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 3, pp. 1700–1710, Jun. 2023, doi: 10.11591/eei.v12i3.4528.

[15] M. S. Osman, A. M. Abu-Mahfouz, and P. R. Page, "A survey on data imputation techniques: water distribution system as a use case," *IEEE Access*, vol. 6, pp. 63279–63291, 2018, doi: 10.1109/ACCESS.2018.2877269.

[16] L. Malan, C. M. Smuts, J. Baumgartner, and 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*, vol. 75, pp. 67–76, Mar. 2020, doi: 10.1016/j.nutres.2020.01.001.

[17] N. Karmitsa, S. Taheri, A. Bagirov, and P. Makinen, "Missing value imputation via clusterwise linear regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1889–1901, 2020, doi: 10.1109/TKDE.2020.3001694.

[18] Y. Zhang and Y. Liu, "Data imputation using least squares support vector machines in urban arterial streets," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 414–417, May 2009, doi: 10.1109/LSP.2009.2016451.

[19] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. De Carvalho, and A. A. Freitas, "A survey of evolutionary algorithms for decision-tree induction," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 3, pp. 291–312, May 2012, doi: 10.1109/TSMCC.2011.2157494.

[20] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, "A novel KNN algorithm with data-driven k parameter computation," *Pattern Recognition Letters*, vol. 109, pp. 44–54, Jul. 2018, doi: 10.1016/j.patrec.2017.09.036.

[21] W. C. Lin and C. F. Tsai, "Missing value imputation: a review and analysis of the literature (2006–2017)," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1487–1509, Feb. 2020, doi: 10.1007/s10462-019-09709-4.

[22] Y. Hanyf and H. Silkan, "A fast and scalable similarity search in high-dimensional image datasets," *International Journal of Computer Applications in Technology*, vol. 59, no. 1, p. 95, 2019, doi: 10.1504/IJCAT.2019.10018181.

[23] S. J. Choudhury and N. R. Pal, "Imputation of missing data with neural networks for classification," *Knowledge-Based Systems*, vol. 182, p. 104838, Oct. 2019, doi: 10.1016/j.knosys.2019.07.009.

[24] R. G. Mantovani, "Texture," Laboratory of Image Processing and Pattern Recognition (INPG-LTIRF), 2016.

[25] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Networks*, vol. 1, no. 1, pp. 75–89, Jan. 1988, doi: 10.1016/0893-6080(88)90023-8.

[26] R. K. Bock et al., "Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 516, no. 2–3, pp. 511–528, Jan. 2004, doi: 10.1016/j.nima.2003.08.157.

[27] A. L. Freire, G. A. Barreto, M. Veloso, and A. T. Varela, "Short-term memory mechanisms in neural network learning of robot navigation tasks: a case study," in *2009 6th Latin American Robotics Symposium, LARS 2009*, Oct. 2009, pp. 1–6, doi: 10.1109/LARS.2009.5418323.

[28] C. L. Blake and C. J. Merz, "Image segmentation data set," UCI Machine Learning Repository, 1990.

[29] L. Xu, A. Krzyżak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992, doi: 10.1109/21.155943.

[30] Y. Hanyf and H. Silkan, "A queries-based structure for similarity searching in static and dynamic metric spaces," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 188–196, Feb. 2020, doi: 10.1016/j.jksuci.2018.05.004.

[31] Y. Hanyf, H. Silkan, and H. Labani, "An improvable structure for similarity searching in metric spaces: application on image databases," in *Proceedings - Computer Graphics, Imaging and Visualization: New Techniques and Trends*, CGiV 2016, Mar. 2016, pp. 67–72, doi: 10.1109/CGiV.2016.22.

[32] Y. Hanyf, H. Silkan, and H. Labani, "Criteria and technique to choose a good ρ parameter for the D-index," in *2015 Intelligent Systems and Computer Vision, ISCV 2015*, Mar. 2015, pp. 1–6, doi: 10.1109/ISACV.2015.7106169.

[33] Y. Hanyf and H. Silkan, "Fast similarity search in high dimensional image data sets," in *ACM International Conference Proceeding Series*, Mar. 2017, vol. Part F1294, pp. 1–5, doi: 10.1145/3090354.3090426.

[34] Z. Kouahla et al., "A survey on big IoT data indexing: potential solutions, recent advancements, and open issues," *Future Internet*, vol. 14, no. 1, p. 19, Dec. 2022, doi: 10.3390/fi14010019.

[35] M. Zhang, L. Yang, Y. Dong, J. Wang, and Q. Zhang, "Picture semantic similarity search based on bipartite network of picture-tag type," *PLoS ONE*, vol. 16, no. November, p. e0259028, Nov. 2021, doi: 10.1371/journal.pone.0259028.

BIOGRAPHIES OF AUTHORS

Youssef Hanyf holds a PhD degree in computer science from Chouaib Doukkali University, Morocco in 2017. He also received his B.Sc. (Mathematics and informatics) and M.Sc. (Software quality) from the same University, in 2009 and 2011, respectively. Currently, He is a professor of computer science at National School of Commerce and Management of Dakhla, Ibn Zhor University, Dakhla, Morocco. His research includes high-dimensional data processing, Data structure, Image processing, information retrieval, similarity search, machine learning, and recommendation systems. He has published over 13 papers in international journals and conferences. He can be contacted at email: youssef.hanyf@gmail.com or y.hanyf@uiz.ac.ma.

Hassan Silkan receives the PhD in computer sciences from Sidi Mohamed Ben Abdellah University, FSDM, Morocco. Currently, he is a professor in Chouaib Doukkali University, Department of Computer Science, Faculty of sciences El Jadida, Morocco. He published more than 32 papers in international journals and conferences in the fields of Shape Representation and Description, Similarity search, Content based images retrieval, Database Indexing, Multimedia Databases, and others. He can be contacted at email: silkan_h@yahoo.fr or silkan.h@ucd.ac.ma.