IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2024, accepted 27 October 2024, date of publication 1 November 2024, date of current version 12 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3490396

RESEARCH ARTICLE

A Perspective on the Missing at Random Problem: Synthetic Generation and Benchmark Analysis

JUAN-FRANCISCO CABRERA-SÁNCHEZ[1], RICARDO CARDOSO PEREIRA[2,3], PEDRO HENRIQUES ABREU[4], (Member, IEEE), AND ESTHER-LYDIA SILVA-RAMÍREZ[1]

[1]Department of Computer Science and Engineering, University of Cádiz, Puerto Real, 11519 Cádiz, Spain
[2]Miguel Torga Institute of Higher Education, 3000-132 Coimbra, Portugal
[3]Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
[4]Department of Informatics Engineering, Centre for Informatics and Systems, University of Coimbra, 3030-790 Coimbra, Portugal

Corresponding author: Juan-Francisco Cabrera-Sánchez (juanfrancisco.cabrera@uca.es)

This work was supported in part by the Ministerio de Ciencia, Innovación y Universidades (MCIN)/Agencia Estatal de Investigación (AEI)/10.13039/501100011033 under Grant PID2022-137646OB-C33, and in part by the European Regional Development Fund (ERDF) A way of making Europe.

ABSTRACT Progressively more advanced and complex models are proposed to address problems related to computer vision, forecasting, Internet of Things, Big Data and so on. However, these disciplines require preprocessing steps to obtain meaningful results. One of the most common problems addressed in this stage is the presence of missing values. Understanding the reason why missingness occurs helps to select data imputation methods that are more adequate to complete these missing values. Missing at Random synthetic generation presents challenges such as achieving extreme missingness rates and preserving the consistency of the mechanism. To address these shortcomings, three new methods that generate synthetic missingness under the Missing at Random mechanism are proposed in this work and compared to a baseline model. This comparison considers a benchmark covering 33 data sets and five missingness rates (10%, 20%, 40%, 60%, 80%). Seven data imputation methods are compared to evaluate the proposals, ranging from traditional methods to deep learning methods. The results demonstrate that the proposals are aligned with the baseline method in terms of the performance and ranking of data imputation methods. Thus, three new feasible and consistent alternatives for synthetic missingness generation under Missing at Random are presented.

INDEX TERMS Data preprocessing, missing data, missing at random, synthetic missingness, deep learning, machine learning, artificial intelligence.

I. INTRODUCTION

Missing data pose a current problem for all types of applications considering the volume of information exchanged every second. This issue arises due to the absence of values registered for certain features. It is occasionally attributed to errors of non-response or inconsistency. Therefore, it must be addressed before any knowledge generation or decision making. In particular, in some sensible contexts, such as medicine, economics, and public policy, missing data problems may have a greater impact. It is valuable not only for regression or classification tasks but also for other preprocessing steps such as feature selection.

Missingness may be present only in certain features and may be related to its values or those of other variables. This behavior, which is referred to as the missingness mechanism, was described by Little and Rubin [1]. Briefly, these mechanisms can be affected by randomness; in Missing Completely at Random (MCAR) and Missing at Random (MAR) mechanisms, missingness is only affected by observed features, or there is no relation at all; and in Missing not Random (MNAR) mechanisms, missingness cannot be explained based on observed values. Typically, the mechanism is intrinsically related to the area in which knowledge data are collected. Medical data tend to be more

The associate editor coordinating the review of this manuscript and approving it for publication was Alex James.

sensitive to MNAR mechanism [2] and MAR [3] while those that are related to sensors are MCAR or MAR [4].

Initially, this problem was ignored by performing List-wise deletion, which removes any instances with missing values. A better approach is to estimate what the real data were from the observed values. This method is referred to as Data Imputation. Traditionally, statistical methods, such as mean, mode, or regression models, have been used to perform this estimation. However, some of these methods may produce bias [5]. In addition, methods based on similarity, such as Hot-Deck imputation, have been widely used. Reference [6] employed this approach to impute geographical coverage data to improve water quality decision-making. Another example is Multiple Imputation by Chained Equations (MICE) which is considered a state-of-the-art method. This approach was used in [7] for substituting anomalous data or missing electricity usage data.

Machine learning methods such as Neural Networks or Decision Trees are a better alternative than statistical methods because they typically provide better results [8]. In [9] Multilayer Perceptrons were used to impute data from diverse areas exposed to perturbation procedures that generate missing data under the MCAR mechanism. The results demonstrated the superiority of the machine learning approach to address missing data problems.

Decision Trees and by extension, Random Forests, are suitable for treating data sets with mixed features. In the same manner, missForest proposed in [10] tackled missingness in both feature types. Results reflect that this method provides robustness in multiple situations. In addition, the authors considered that the proposed method could outperform other imputation methods in extreme cases, such as when more variables than observations.

Deep Learning models have also been increasingly used to address such problems. As an example, Denoising Autoencoders (DAE) have demonstrated good results in reconstructing noisy images [11]. This approach has also been extended to tabular data, where noise represents missing data. Reference [12] used DAE to complete traffic data in a time-series context.

Other deep learning methods have shown good results in different domains. In the case of time series problems, Generative Adversarial Networks (GAN) or Variational Autoencoders (VAE) have been proposed to address missingness. [13] proposed GAN with GRUI cells to impute missing data on KDD data set, and they obtained overall better results than other methods. Also, [14] proposed a new generative model called GAIN, based on GAN networks, to carried out the imputation task. It was compared to other methods, such as MICE and Autoencoders, and achieved better results. The work in [15] proposed VAE with a Gaussian process in different domains, and it demonstrated superiority on image and time-series data sets.

The majority of previously exposed examples were subjected to the MCAR mechanism considering real data. However, when considering complete data sets without missing values, a methodology to simulate the characteristics of different mechanisms is required. In case of missing data under the MCAR mechanism, the approach is straightforward because it is easier to achieve its properties. However, this is not be the case for MAR and MNAR mechanisms, since it is more difficult to generate extreme missingness rates while maintaining the premises of the mechanisms [16].

In [17] an adjusted weight voting method for random forest was proposed for handling missing data. The missingness was only generated for a single feature by dividing the rank of the values of each index by the sum of all ranks of another variable. In addition, the maximum missingness rate considered was a 30%. It should be noted that only a single feature was perturbed; thus, no extreme missingness rates could be achieved.

The multivariate synthetic generation of missingness was considered in [18]. The perturbation procedure first divided the data set into k subsets and computed the weighted sum scores to determine the probability of being marked as missing. A weight was assigned to each feature, and in the case of MAR, the weights were set to zero for features that would be perturbed.

Moreover, in [19] the original data set was also divided into subsets. Missingness was generated in accordance with the desired ratio. In order to achieve this, features were paired using correlation. Subsequently, a probability distribution was employed to determine which instances would undergo perturbation. The maximum rate of missingness considered was 40%.

A benchmark of the three missingness mechanisms was proposed in [20]. The synthetic missingness generation of MAR was based on [21]. Therefore, two percentiles were selected for a given feature, each corresponding to the lowest and highest values. If values of another variable fell in that percentile, they were marked as missing. A maximum missingness rate of 50% was contemplated.

In [22] the missingness was introduced only for independent variables considering a Bernoulli distribution between fully observed features. Only 20% and 50% missingness rates were selected.

The misstastic library was used in [23] for generating missingness under the MAR mechanism using a logistic model. The highest ratio considered is 50%.

The exposed limitations are often caused by the synthetic generation method. For example, in [24] the maximum possible missingness rate was 25%, while in [25] was 50%. These are some reasons why MAR missingness is more challenging than the MCAR mechanism.

In [26] these issues were addressed for the MNAR mechanism. But, for MAR mechanism no similar study has been encountered.

Therefore, in this work, a study on three novel techniques to generate synthetic missing data under MAR mechanism is presented. The first approach uses the Pearson correlation, in which the computed values help to establish which variable will be subjected to have missingness. On the other

hand, Mutual Information (MI) and Minimum Redundancy Maximum Relevance (mRMR) are proposed to carry out the same task. Moreover, each proposal is prepared to handle and generate missingness considering multiple scenarios. So, for each proposed method there are multiple variants. Specifically, these variants allow different masking processes depending on feature type, continuous and categorical. Moreover, multiple causative features are considered for allowing rates from 10% to extreme rates as 80%. In addition, these methods are compared in an exhaustive benchmark using some of the considered state-of-the-art methods for data imputation, such as Autoencoders and MICE, as well as traditional methods such as mean/mode or k-NN. Therefore, using data imputation methods for the benchmark allow to assess the differences between the proposed methods and the baseline method.

The remainder of this paper is organized as follows: an overview of missing data concepts is presented in section II. The experimental setup, including the data sets, preprocessing, and imputation methods, is explained in section III. In section IV, the proposals are described. In section V, the results are presented and discussed. The conclusions and future work are described in section VI.

II. MISSING DATA

Missing data or the absence of values in some observations for one or more features represents a problem when trying to generate new knowledge. Missingness can be caused by multiple reasons, as previously mentioned. The most suitable method to impute missing values depends on the mechanisms and knowledge extraction application. Formally, given a X data set with n observations and p features, a nxp M matrix is defined, representing the missingness can be obtained as follows.

$$M_{i,j} = \begin{cases} 0 \ if \ \text{ missing} \\ 1 \ if \ \text{observed} \end{cases} \tag{1}$$

This matrix M is used as a mask to distinguish missing data patterns. Insights about these patterns can be obtained by sorting the data according to the number of missing values. Two aspects are generally considered, the number of features affected by an absence of value and how missingness appears. In the case of the former aspect, univariate and multivariate patterns can be noticed depending on whether only one or more features are affected. Moreover, a monotone pattern is observed when an absence of value is present for the same variables and observations, whereas a non-monotone is distinguished in any other case.

Missing data can arise from different causes. Reference [1] distinguished three mechanisms considering the relationship between missing values and variables.

- MCAR (missing completely at random). The appearance of missing data is not dependent on any variable nor its values, whether observed or unobserved.

$$P[X_j = miss | X_1, \ldots, X_p] = P[X_j = miss]$$

- MAR (missing at random). Missing data presence depends on the values of other observed variables but not feature values.

$$P[X_j = miss \mid X_1, \ldots, X_p] = P[X_j = miss \mid X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p]$$

- MNAR (missing not at random). The missingness of any variable depends on its own values.

$$P[X_j = miss \mid X_1, \ldots, X_p] = P[X_j = miss \mid X_j]$$

As noted in section I, the most common mechanism is MCAR, regardless of whether it is indicated or not, as it is assumed if not specified. Therefore, three different methods for generating synthetic MAR data are proposed in this work.

III. EXPERIMENTAL SETUP

In this section, data sets and their respective characteristics are described, as well as the preprocessing steps performed. In addition, the imputation methods are described in detail, together with hyperparameters. Finally, the metrics used to assess the imputation quality are explained.

A. DESCRIPTION OF THE DATA

In these experiments, 33 data sets were selected considering various domains and properties, such as the number of categorical and continuous features or the number of instances. The data sets were selected from the OpenML repository [27] covering various situations without performing any previous modification as these data sets does not contain missing data. Table 1 summarizes the characteristics of the selected data sets for the experiments. The division reflects continuous, categorical, or mixed data sets. There are four columns, where n represents the number of instances, p_q and p_c show the number of continuous and categorical features, respectively, and p_t represents the total number of features.

B. PREPROCESSING

Preprocessing is a key step of any machine learning problem. This is how data are treated to be appropriate for use in machine learning models. Considering the different types of data sets selected for the experiments, several transformations were applied.

Continuous features were normalized into the range[0, 1], which facilitated faster convergence in the neural network optimization algorithms.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

For categorical features, a codification based on One-hot Encoding algorithm is performed. This method encodes each category of the feature with a binary variable; thus, given a feature with c categories, c binary variables are added to the data set.

Therefore, model inputs are higher dimensional than initially, as it can be observed in Table 1 with p_t and p_c

TABLE 1. Data sets used in the experiments.

Name	n	p_q	p_c	p_t
Blood-transfusion	748	4	0	4
Bodyfat	252	14	0	14
Mammography	11183	6	0	6
Phoneme	5404	5	0	5
Pollen	3848	5	0	5
Quake	2178	3	0	3
seeds	210	7	0	7
sylvine	5124	20	0	20
Thyroid-new	215	5	0	5
Twonorm	7400	20	0	20
Vehicle	846	18	0	18
Breast	699	0	9	89
Car	1728	0	6	21
HayesRoth	160	0	4	15
Kropt	28056	0	7	40
Lymphography	148	0	18	60
Nursery	12960	0	9	27
PhisingWebsites	11055	0	31	68
Sensory (about wine)	576	0	12	36
Solar	1389	0	7	16
SPECT	267	0	22	44
TicTacToe	958	0	10	27
Abalone	4177	7	1	10
Boston	506	12	1	14
Cardiovascular-Disease	70000	5	6	19
Cmc (Contraceptive)	1473	2	7	24
churn	5000	16	4	33
Eye-movements	10936	24	3	30
Flag	194	10	18	77
PopularKids	478	6	5	22
Saheart	462	8	1	10
SKY	10000	15	1	20
Thoracic-surgery	470	3	13	37

values. A decoding process based on the SoftMax algorithm is performed to obtain a complete data set and proceed to error measurement.

C. IMPUTATION METHODS

Considering the amount of information processed, addressing missing data problems is critical. This problem can arise in multiple disciplines, and several approaches have been used to address it. Aforementioned, from traditional methods, such as Mean/Mode or Hot-Deck, to state-of-the-art methods, such as MICE and SoftImpute. Moreover, Machine Learning methods such as k-NN, and methods based on neural networks have been widely used in the literature. Since an extensive benchmark is required to assess the performance of MAR data, these seven methods have been selected to handling incomplete data. Thereby, a comprehensive comparison between the MAR synthetic generation procedures can be conducted. These methods are described below.

- *Mean/mode substitution (MM)*: This method completes the missing datum with the mean or mode of the observed feature values. If a variable presents several records with missing values, all of them are replaced with the same value, which is computed depending on the data type. Thus, the mean is computed for continuous features and mode for categorical variables.
- *k-Nearest Neighbors (k-NN)*: This method compares individuals with missing data to complete instances. For this purpose, the distance is computed, so that the nearest k records are used to replace incomplete data according to the mean or mode of these k records.
- *SoftImpute (SI)*: This method allows to obtain a matrix in which missing values are replaced iteratively using soft-thresholded Single Value Decomposition (SVD). Therefore, data set is completed through a low-rank solution.
- *Multiple Imputation Chained Equations (MICE)*: This method is based on the multiple imputation approach. MICE uses a succession of regression models to progressively fill in missing values in a feature. Once a feature has been completed, it is used as an additional independent variable for the remaining features. This process is repeated a certain number of times to generate a unique data set.
- *Denoising Autoencoders (DAE)*: This model is a type of fully-connected neural network whose goal is to reproduce its input by learning a latent representation. Although it is usually associated with computer vision problems, its gaining presence for addressing missing data problems due to its noise removal capability. In this context, training is performed on a corrupted version of the data set to reconstruct the original database.
- *Variational Autoencoders (VAE)*: Similarly to DAE, it is capable of reproducing its input to output by learning the data distribution through mean and variance. Using a Gaussian distribution, samples can be drawn to feed the latent space layer in the VAE.
- *GAIN*: A model based on Generative Adversarial Networks, which comprise a discriminator and generator. In the data imputation context, the generator addresses the imputation process, and the discriminator attempts to discern whether the data are actually real or imputed. GAIN model achieves, through a hint mechanism, an imputation that is consistent with the data distribution.

These data imputation methods have either been implemented or public implementations have been used. MICE, k-NN and Mean/Mode are available in Scikit-Learn library. The Muzellec implementation [28] is considered for the SoftImpute method. Autoencoder-based methods have been implemented using the Pytorch-Lightning library. For GAIN, an adaptation of the Pytorch version[1] is used.

In Table 2 a summary of the hyperparameters and their values for each model is provided. Considering the extensive benchmark and high computational cost, careful trials were conducted to determine the optimal hyperparameters and ensure a balance between the methods and data sets.

D. MODEL EVALUATION

To assess the performance of each imputation method, several metrics were considered. The selection depended on three considerations: continuous only, categorical only,

[1] https://github.com/dhanajitb/GAIN-Pytorch

TABLE 2. Models configuration.

Model	Hyperparameter	Value
k-NN	k	5
	Distance	Euclidean
MICE	Estimator	Random Forest
	#Iterations	25
DAE	Structure	2 layers
	Hidden nodes	$\theta \cdot h_{n-1}$
	Latent space	$\theta^2 \cdot p_t$
	Optimizer	AdamW
	Learning rate	0.001
	Regularization (l_2)	0.01
VAE	Structure	2 layers
	Hidden nodes	$\theta \cdot h_{n-1}$
	Latent space	$\theta^2 \cdot p_t$
	Optimizer	AdamW
	Learning rate	0.001
	Regularization (l_2)	0.01

and mixed data sets. In the case of a data set with only continuous features, Weighted Mean Absolute Percentual Error (WMAPE) is computed.

$$WMAPE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|} \tag{3}$$

If the database only contains categorical features, then a decoding process to reverse the One-Hot encoding through the Softmax function is performed before obtaining the error rate.

$$Error_{rate} = \frac{\sum_{i=1}^{n} y_i \neq \hat{y}_i}{n} \tag{4}$$

The selection of these two metrics allow to report errors that are on the same scale.

IV. METHODOLOGY

In this section, three novel procedures to generate synthetic missing data under MAR mechanism are proposed.

Synthetic generation of missing values mimic real-world scenarios to assess the quality of the data imputation models. To simulate MAR situations, the relationship between each independent variable is instead calculated. Thus, most correlated features can be used to mask another feature as described in [25]. This synthetic generation procedure is considered for missing rates of 10%, 20%, 40%, 60%, and 80%.

An example of this strategy can be observed in Figure 1, which represents how correlated variable X_j is to X_i. More intense colors indicate greater correlation or dependence. Thus, feature X_j is perturbed according to the data type of X_i.

In the case of continuous features, the indexes of the lowest or highest values of X_i are used as a mask for X_j. An example of this behavior is shown in Figure 2, in which the intensity of colors correlates with the value. Therefore,

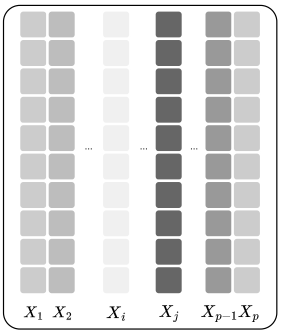


FIGURE 1. Correlation between feature X_i and the others.

lighter cells are equivalent to lower values, whereas darker positions correspond to higher values. Note that feature X_j is more correlated to X_i; thus, it is masked according to the selected approach. In this example, the indexes of the lowest values of X_i are used to generate missingness.

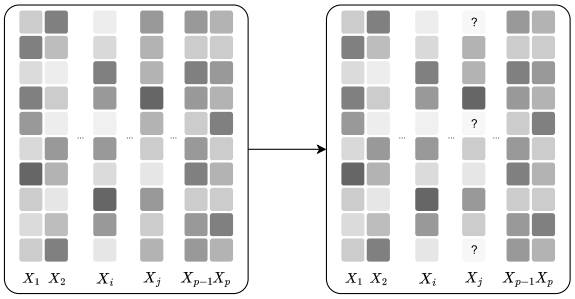


FIGURE 2. Missing values generation procedure.

In contrast, the most or less frequent approaches may be followed if it is a categorical feature. Note that if there are not enough instances of a certain category, random sampling of the next frequent category is performed. Thus, up to four variants can be considered for each of the proposed methods.

Therefore, this process is iteratively repeated by pairing up independent-dependent features until the desired missing rate is achieved. Figure 3, shows an overview of the design of the experiments. Missing indexes in the perturbed data set are represented by a lighter shade. For the imputed data set, the same color intensity approach was used to represent lower or higher values. Thus, only indexes that were missing show different gray scale colors.

It can be noticed that after the perturbation procedure, a hold-out sampling is carried out, obtaining a train and a test set with both subjected to having missing values in the same proportion. The 70% of the instances are selected for the training set while the remaining 30% is used for the test set. Thus, the data imputation method is trained and then evaluated on the test set to obtain the metrics and a complete data set. These experiments are repeated for 30 trials for each missingness rate. Thus, the inherent variability of the sampling process is mitigated.

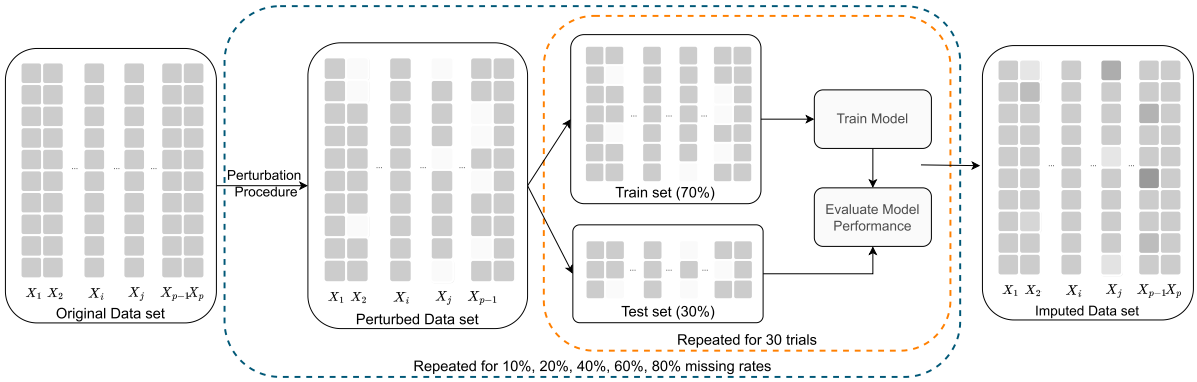


FIGURE 3. Experimental process for each database.

In this study, three novel strategies were considered: Pearson correlation coefficient, Mutual Information (MI), and Minimum Redundancy Maximum Relevance (mRMR). Different aspects can be considered about how synthetic generation is handled. Firstly, these three strategies can have multiple causative features and share a *masking_method* parameter for controlling the masking process. This parameter adjusts the generation of missingness depending on feature type. Lowest or highest values are used in case of continuous features whereas most or less frequent values in case of categorical variables. Thus, the *masking_method* parameter allows to select one of the four aforementioned variants. Algorithm 1 shows the overall process of synthetic missingness generation. The proposed generation methods are benchmarked against previously proposed method for generating missingness under MAR mechanism by Garciarena [29]. This strategy is used as a baseline to compare the performance of the proposals. In order to carry out this benchmark between the synthethic generation procedures the aforementioned data imputation methods are used.

A. BASELINE

The approach proposed by Garciarena [29] involves randomly to sample an unique feature that is used as a causative variable to generate the missingness in the remaining features. Thus, to maintain the restrictions of the MAR mechanism, the indexes of the lowest values of the causative feature are subjected to be missing in other variables. Note that one feature will not be perturbed since the causative variable is not subjected to have missing values; thus, the missingness ratio must be distributed between $p_t - 1$ features. Therefore, only one feature is considered as a causative.

In addition, to be able to handle categorical features, a similar approach of masking the lowest values in other features. So, if the sampled causative variable is categorical, the frequencies of each category are computed and sorted in ascending order. Therefore, indexes of less common categories are those that are subjected to be missing in the remaining features.

B. PEARSON CORRELATION COEFFICIENT

Pearson's correlation coefficient indicates the linear relationship between two variables. This relationship can exhibit strong positive correlation (values near 1), strong negative correlation (values near -1), or no correlation (value of 0).

Therefore, the coefficient is pairwise calculated for each pair of variables in the data set. Then, features that are more correlated with others are used as causative for the masking process.

In Algorithm 1 it can be observed how the missingness is artificially generated after computing the Pearson correlation coefficient between each feature.

Algorithm 1 Pseudocode for Pearson Generation Procedure

Input: *M, missing_rate, masking_method*
Output: *perturbed_dataset*
  $m \leftarrow$ NumberObservations(M)
  $n \leftarrow$ NumberFeatures(M)
  $scores \leftarrow \mathbb{R}^{m \times n}$
  $scores \leftarrow PearsonCorrelation(dataset)$
  $scores \leftarrow abs(scores)$
  $diagonal(scores) \leftarrow 0$
  $miss\_features \leftarrow \{\}$
  while desired missing_rate is not achieved **do**
    $independent, dependent \leftarrow argmax(scores)_{row,column}[0]$
    $miss\_index \leftarrow$ Get $[mr \times m]$ indexes of $M[:, dependent]$ according to *masking_method*
    $perturbed\_dataset[miss\_index, independent] \leftarrow NaN$
    $miss\_features \leftarrow miss\_features \cup dependent$
    for each feature *feat* in *miss_features* **do**
      $scores[feat, :] \leftarrow 0$
      $scores[:, feat] \leftarrow 0$
    end for
  end while

C. MUTUAL INFORMATION (MI)

Mutual Information algorithm is based on information theory and is tied to Shannon entropy. It measures the amount of uncertainty that can be reduced by knowing another feature. This represents the information of a single feature that can be obtained by observing another. Formally, the mutual information of two variables X and Y is defined as follows:

$$I(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

In this definition, $P(x)$ and $P(y)$ are the marginal probabilities, and $P(x, y)$ is the joint probability. Thus, obtaining a nonzero value implies a degree of dependence between the variables. Moreover, achieving a value of 1 means that these features are strongly dependent. However, obtaining a zero value indicates that the variables are completely independent.

Therefore, the Mutual Information is computed between the complete data set and each feature as a target variable. The general procedure can be observed in Algorithm 2.

Algorithm 2 Pseudocode for Mutual Information Generation Procedure

Input: *M, missing_rate, masking_method*
Output: *perturbed_dataset*
  $m \leftarrow$ NumberObservations(M)
  $n \leftarrow$ NumberFeatures(M)
  $scores \leftarrow \mathbb{R}^{m \times n}$
  for each feature f of M **do**
    $target \leftarrow M[:, f]$
    $mi\_scores \leftarrow MutualInformation(M, target)$
    $scores[:, f] \leftarrow mi\_scores$
    $scores[f, f] \leftarrow 0$
  end for
  scores $\leftarrow$ abs(scores)
  $miss\_features \leftarrow \{\}$
  while desired missing_rate is not achieved **do**
    $independent, dependent \leftarrow argmax(scores)_{row,column}[0]$
    $miss\_index \leftarrow$ Get $[mr \times m]$ indexes of $M[:, independent]$ according to *masking_method*
    $perturbed\_dataset[miss\_index, dependent] \leftarrow NaN$
    $miss\_features \leftarrow miss\_features \cup dependent$
    for each feature f in *miss_features* **do**
      $scores[f, :] \leftarrow 0$
      $scores[:, f] \leftarrow 0$
    end for
  end while

Algorithm 3 Pseudocode for MRMR Generation Procedure

Input: *M, missing_rate, masking_method*
Output: *perturbed_dataset*
  $m \leftarrow$ NumberObservations(M)
  $n \leftarrow$ NumberFeatures(M)
  $relevance\_scores \leftarrow \{\}$
  for each feature f of M **do**
    $X \leftarrow M[:, \{1, \ldots, n\} - \{f\}]$
    $target \leftarrow M[:, f]$
    $relevance\_scores \leftarrow relevance\_scores \cup \sum MRMR(X, target)_{relevances}$
  end for
  $independent \leftarrow argmax(relevance\_scores)_{index}[0]$
  $dependent \leftarrow \{1, \ldots, n\} - \{independent\}$
  $miss\_index \leftarrow$ Get $[mr \times m]$ indexes of $M[:, independent]$ according to *masking_method*
  $perturbed\_dataset[miss\_index, dependent] \leftarrow NaN$

D. MINIMUM REDUNDANCY MAXIMUM RELEVANCE (MRMR)

This algorithm is frequently used in feature selection contexts because it selects features with which the target is more correlated. The procedure is based on two principles: selecting features that add information and are not redundant or mere copies, and ensuring that these features are strongly related to the output variable. Thus, this method aims to minimize redundancy R and maximize relevance D of a feature set X to represent target variable y [30]. This method uses the definition of Mutual Information as follows:

$$D = \frac{1}{|X|} \sum_{X_i \in X} I(X_i, y) \qquad R = \frac{1}{|X|} \sum_{X_i, X_j \in X} I(X_i, X_j)$$

where X_i and X_j are two features of the data set and $|X|$ correspond to the total number of features.

In Algorithm 3 can be observed the procedure used to generate missingness with the MRMR technique. In order to select it, the relevance scores of the MRMR method are considered. Then, each feature is considered as a target, and the optimal set of features to represent it is computed. The relevance values are summed up to select the maximum value after all features were considered as target variables. The variable with the maximum value is selected as the causative. Finally, the masking process is carried out in the same manner as in the previous methods. Contrary to the previous method, only one variable is causative, similar to the baseline method.

These experiments have been conducted using the research support supercomputer called Urania from the University of Cádiz. The specifications consist of 20 computing nodes each containing two Intel CPUs Xeon-Platinum 8358 that runs at 2.6GHz and 256 GB of RAM memory.

V. RESULTS AND DISCUSSION

In this study, several methods for generating synthetic missing data under the MAR mechanism are presented. Specifically, the approaches of Pearson, MI and MRMR, were applied considering both feature types: continuous and categorical. Two masking processes are carried out for continuous features, targeting either lower or higher values. In case of categorical features, no significant differences in the results were detected using a most frequent or less frequent approach. Therefore, the masking process for categorical features only involved the use of most frequent values. Thus, for each mechanism two scenarios are considered, masking by lowest or highest values, both with the most frequent approach.

A. BASELINE

First, the results of the baseline generation method are presented. It can be observed in Table 3 how DAE achieves the best results for continuous features. It is closely followed by VAE. In addition, other methods, such as k-NN and MICE, demonstrate slightly worse performance. Mean/Mode and SoftImpute obtain the worst results for continuous features.

In Table 4, the results for categorical features are presented. In this case, differences arise between missingness rates. Autoencoder-based methods outperform the remaining methods in almost all scenarios except for 10% and 40% where there are differences below to 0.5%. MICE presents a worst performance than in continuous features, ranking second to last next to SoftImpute and GAIN.

The results for the Baseline methods show Autoencoder-based methods, DAE and VAE, achieve lower errors than the remaining methods. However, GAIN does not exhibit good performance for any feature type. k-NN or MICE, obtain

TABLE 3. WMAPE of continuous features for Baseline generation.

	10%	20%	40%	60%	80%
Mean/Mode	0.747	0.663	0.591	0.588	0.586
SoftImpute	0.830	0.802	0.790	0.821	0.865
k-NN	**0.477**	0.510	0.487	0.516	0.543
MICE	0.527	0.551	0.529	0.551	0.576
GAIN	0.648	0.656	0.640	0.660	0.670
VAE	0.513	0.467	0.436	0.423	0.430
DAE	0.494	**0.448**	**0.408**	**0.402**	**0.410**

TABLE 4. Error rate of categorical features for baseline generation.

	10%	20%	40%	60%	80%
Mean/Mode	0.343	0.344	**0.343**	0.359	0.370
SoftImpute	0.383	0.372	0.383	0.433	0.480
k-NN	**0.338**	0.338	0.361	0.379	0.396
MICE	0.393	0.396	0.400	0.427	0.420
GAIN	0.384	0.391	0.415	0.433	0.453
VAE	0.345	0.342	0.349	0.358	0.367
DAE	0.341	**0.336**	0.345	**0.347**	**0.360**

heterogeneous results. Although k-NN demonstrates robust behavior for both data types, MICE does not obtain consistent results for categorical variables because it ranks in a worst position compared to continuous features.

B. PEARSON

The results for the Pearson generation methods are discussed below. In Table 5, the WMAPE of each data imputation method for each variant of Pearson generation is presented. DAE outperforms the rest of methods in all missingness rate scenarios when masking with the highest values for continuous features. VAE method achieves a similar and stable performance. Other methods, such as k-NN and MICE, demonstrate deterioration related to an increase in the missingness rate. Conversely, GAIN, Mean/Mode, and SoftImpute achieve the worst results in this scenario. In particular, SoftImpute exhibits the steepest decay at extreme rates. A similar analysis occurs when masking with the lowest values. The only difference is that MICE outperforms the other methods at 10% and 20% of missingness, showing a similar worsening at higher rates.

TABLE 5. WMAPE of continuous features for pearson generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
Pearson ↑	10%	0.484	0.424	0.355	0.325	0.646	0.338	**0.311**
	20%	0.473	0.474	0.378	0.335	0.623	0.336	**0.304**
	40%	0.472	0.582	0.430	0.396	0.609	0.354	**0.314**
	60%	0.484	0.682	0.451	0.452	0.601	0.380	**0.342**
	80%	0.514	0.803	0.496	0.542	0.606	0.401	**0.373**
Pearson ↓	10%	0.817	0.490	0.549	**0.368**	0.601	0.495	0.472
	20%	0.720	0.494	0.515	**0.396**	0.578	0.453	0.421
	40%	0.661	0.542	0.553	0.464	0.566	0.427	**0.386**
	60%	0.647	0.619	0.582	0.511	0.562	0.422	**0.371**
	80%	0.658	0.723	0.645	0.640	0.604	0.421	**0.373**

In the Table 6, the results for data sets with categorical features are shown. While the trend is nearly the same as that observed for continuous features, greater diversity is observed. At lower missingness rates, 10% and 20%, SoftImpute and Mean/Mode achieve the best results when masking with the highest values. Similarly, SoftImpute outperforms the other methods when masking with the lowest values for 10% of missingness. The results for both masking processes, lower and higher, are similar from 40% to 80%. DAE method exhibits the lowest error rate. Other methods such as k-NN and VAE, demonstrate inferior results than DAE at high missingness rates. MICE and SoftImpute manifest the worst performance at extreme missingness rates.

TABLE 6. Error rate of categorical features for pearson generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
Pearson ↑	10%	0.328	**0.311**	0.312	0.380	0.341	0.362	0.339
	20%	**0.317**	**0.317**	0.327	0.374	0.324	0.354	0.328
	40%	0.332	0.365	0.357	0.395	0.345	0.338	**0.311**
	60%	0.358	0.407	0.371	0.413	0.372	0.342	**0.318**
	80%	0.380	0.458	0.395	0.429	0.405	0.347	**0.324**
Pearson ↓	10%	0.318	**0.307**	0.311	0.379	0.329	0.343	0.314
	20%	0.321	0.318	0.324	0.386	0.335	0.330	**0.307**
	40%	0.333	0.382	0.354	0.403	0.342	0.339	**0.313**
	60%	0.358	0.436	0.370	0.427	0.382	0.335	**0.307**
	80%	0.378	0.458	0.388	0.428	0.401	0.346	**0.323**

The results for continuous and categorical features indicate that Machine Learning-based methods achieve a more stable and precise performance than the other methods. In particular those based on Autoencoders, consistently rank among the first positions in all scenarios. Conversely, GAIN method, presents worse results for continuous features.

C. MUTUAL INFORMATION

The results of the Mutual Information generation method are discussed below. First, the performance of the different imputation methods for continuous features is presented in Table 7. In terms of which method rank better when considering a masking with the highest values process a comparable situation is observed. DAE outperforms other methods in all missingness rates. VAE and MICE method achieve slightly worse error than these methods. Other methods such as Mean/Mode or GAIN present the worst performance. Notable differences arise from 40% to 80%. Autoencoders-based methods exhibit robustness while the other demonstrate a deterioration to the increase of missingness rates. In addition, GAIN show a lower error as the rate increases. When masking with the lowest values, higher WMAPE is obtained in general. The analysis is comparable to the highest values masking process. The only remarkable difference is the better performance of the MICE method, which ranks first at the lowest missingness rates, 10% and 20%. Mean/Mode, GAIN, VAE and DAE exhibit a slight decrease in the error at extreme rates compared to the initial level.

The results for the categorical features are presented in Table 8. Excluding MICE, which achieves the worst results

TABLE 7. WMAPE of continuous features for MI generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
MI ↑	10%	0.480	0.425	0.376	0.354	0.675	0.330	**0.315**
	20%	0.470	0.453	0.435	0.350	0.643	0.335	**0.295**
	40%	0.470	0.569	0.454	0.435	0.633	0.346	**0.315**
	60%	0.476	0.671	0.466	0.436	0.607	0.377	**0.335**
	80%	0.498	0.773	0.505	0.506	0.571	0.399	**0.365**
MI ↓	10%	0.697	0.471	0.492	**0.363**	0.642	0.470	0.429
	20%	0.647	0.503	0.512	**0.398**	0.624	0.446	0.410
	40%	0.644	0.558	0.559	0.433	0.584	0.421	**0.360**
	60%	0.624	0.588	0.605	0.508	0.567	0.426	**0.362**
	80%	0.623	0.704	0.613	0.584	0.591	0.421	**0.382**

at 10% of missingness rate, the other methods exhibit a high degree of similarity when masking by the highest values. In particular, SoftImpute, k-NN and DAE obtain the lowest error. GAIN and VAE perform slightly worse than these methods. MICE present the worst performance in this case. The results at 20% present a change since DAE ranks first, followed closely by SoftImpute, with almost no differences. The performance degradation of some methods is observed from 60%. Neural network-based models are more stable to the increase in the missingness rate, whereas other models performs slightly worse in this scenario. Specially, MICE, SoftImpute, and k-NN present the worst results at the highest missingness rates. Considering the masking by the lowest values, differences arise at lower missingness rates. k-NN method outperforms at 10% with a slight difference above SoftImpute. DAE, Mean/Mode, and k-NN exhibit minimal differences in 20% scenario. In addition, as observed with other generation methods, there is a decrease in the performance of machine learning-based models, namely, MICE, k-NN and SoftImpute starting from 40%. DAE and VAE demonstrate robust behavior while maintaining lower error rates than the other methods.

TABLE 8. Error rate of categorical features for MI generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
MI ↑	10%	0.337	**0.310**	0.320	0.373	0.341	0.346	0.332
	20%	0.330	0.306	0.332	0.366	0.310	0.345	**0.304**
	40%	0.380	0.374	0.356	0.409	0.357	0.350	**0.307**
	60%	0.378	0.425	0.386	0.429	0.376	0.336	**0.303**
	80%	0.407	0.447	0.416	0.437	0.416	0.373	**0.339**
MI ↓	10%	0.348	**0.320**	0.319	0.390	0.343	0.363	0.347
	20%	**0.333**	0.321	0.329	0.381	0.326	0.358	0.321
	40%	0.386	0.400	0.386	0.394	0.369	0.366	**0.326**
	60%	0.393	0.400	0.387	0.424	0.374	0.362	**0.320**
	80%	0.383	0.410	0.390	0.421	0.393	0.348	**0.321**

D. MINIMUM REDUNDANCY MAXIMUM RELEVANCE

In this subsection, The results of the MRMR generation methods are presented. In the Table 9 it can be observed the performance of the compared methods considering continuous features. First, when masking with the highest values, a scenario comparable to the Pearson method is observed. These methods, including GAIN and SoftImpute, exhibit increasingly unstable behaviour with rising missingness rates. In contrast, Machine Learning-based methods, including MICE, GAIN, VAE and DAE exhibit less decay to this increase. Particularly, DAE and VAE achieved the best results while MICE and k-NN show poorer performance from 40% missingness rate. Similarly, DAE and VAE outperform the other methods when masking with lowest values. It is noteworthy that methods such as Mean/Mode, VAE, and DAE exhibit an improvement in performance as missingness increases.

TABLE 9. WMAPE of continuous features for MRMR generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
MRMR ↑	10%	0.470	0.628	**0.368**	0.478	0.724	0.381	0.373
	20%	0.454	0.721	0.354	0.462	0.680	0.340	**0.332**
	40%	0.466	0.851	0.407	0.497	0.639	0.357	**0.345**
	60%	0.490	0.936	0.462	0.514	0.620	0.393	**0.375**
	80%	0.505	0.973	0.488	0.517	0.613	0.414	**0.388**
MRMR ↓	10%	0.747	0.830	**0.477**	0.527	0.648	0.513	0.494
	20%	0.663	0.802	0.510	0.551	0.656	0.467	**0.448**
	40%	0.591	0.790	0.487	0.529	0.640	0.436	**0.408**
	60%	0.588	0.821	0.516	0.551	0.660	0.423	**0.402**
	80%	0.586	0.865	0.543	0.576	0.670	0.430	**0.410**

The results of MRMR generation method considering categorical features are presented in Table 10. There are differences with respect to continuous features. Considering the masking by highest values, Mean/Mode presents the lowest error up to 40% of missingness rate. The differences between this method and and the one which ranks in second place are more subtle between 20% and 40%, as DAE and k-NN exhibit a difference of just 1%. From this rate to 80% the difference between DAE and the other widens. In contrast, VAE method presents a stable behavior, particularly with lower missingness rates. GAIN and MICE present the worst performance in almost all missingness rates. If a masking process with the lowest values is instead considered, more differences arise. In this scenario k-NN method clearly achieves the best results at lower missingness rates. Note that this performance decreases as the missingness increases, which demonstrates less robust behavior. From 20% to 80% DAE method outperforms the other methods, with the exception of the case of 40%, where there is hardly any difference.

TABLE 10. Error rate of categorical features for MRMR generation.

		MM	SI	k-NN	MICE	GAIN	VAE	DAE
MRMR ↑	10%	**0.327**	0.374	0.346	0.437	0.423	0.361	0.353
	20%	**0.336**	0.407	0.356	0.435	0.428	0.350	0.343
	40%	**0.347**	0.408	0.357	0.425	0.436	0.358	0.358
	60%	0.364	0.450	0.382	0.441	0.435	0.368	**0.362**
	80%	0.381	0.496	0.404	0.425	0.448	0.370	**0.369**
MRMR ↓	10%	0.343	0.383	**0.338**	0.393	0.384	0.345	0.341
	20%	0.344	0.372	0.338	0.396	0.391	0.342	**0.336**
	40%	**0.343**	0.383	0.361	0.400	0.415	0.349	0.345
	60%	0.359	0.433	0.379	0.427	0.433	0.358	**0.347**
	80%	0.370	0.480	0.396	0.420	0.453	0.367	**0.360**

E. DISCUSSION

A discussion is offered to summarize the results in four scenarios, one for each combination of feature type and

masking process. Thus, for each data set, the obtained metrics are averaged considering the missingness rates. So, the mean and the confidence interval of the 33 data sets are considered for the comparison. First, a summary of the results obtained for the continuous features using the lowest values for the masking process is shown in Figure 4.

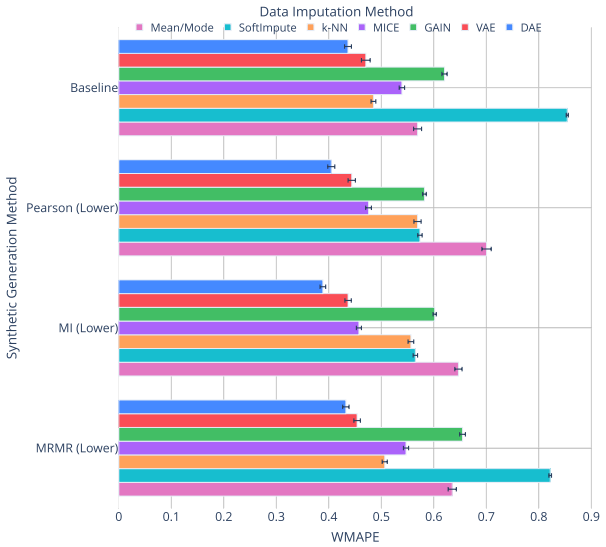


FIGURE 4. Summary of the results of continuous feature and lower masking process.

The results of the proposed methods are consistent, although there are some differences between the performance of some data imputation methods. These results are also similar to those obtained using the baseline method, which exhibits subtle differences in performance for almost all data imputation methods. In particular, the behavior of the GAIN and k-NN methods clearly deviates from that of the proposed methods. Furthermore, there is another difference between k-NN and MICE, in terms of which method ranks third in the comparison. Autoencoder-based methods outperform the remaining methods in terms of the compared generation methods. MRMR exhibit fewer differences to the Baseline method, which is expected given that in both methods only one variable is causative.

A summary of the results for the continuous features using the highest values for the masking process is observed in Figure 5. The results produced by the proposed generation methods are consistent with each other, as shown in the previous example. Note that slightly lower errors occur for these masking processes. The deviation between the proposals and the Baseline method is greater due to opposing extremes in the masking processes. In the proposals, there is a notable decline in the GAIN and SoftImpute methods. The performance gap between the remaining imputation methods and the previous scenario is less pronounced than that of GAIN and SoftImpute.

Below are presented the results for the categorical features. Figure 6 shows the summary of results with masking

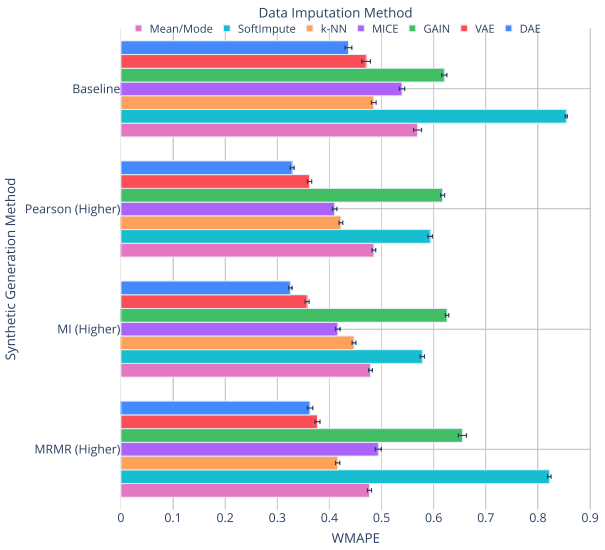


FIGURE 5. Summary of the results of continuous feature and higher masking process.

by lowest values. The proposed methods exhibit fewer differences compared to the Baseline. The improvement in the proposals is clearer in the performance of DAE and GAIN compared to others. The proposals slightly reduce errors without deviating from the baseline method.

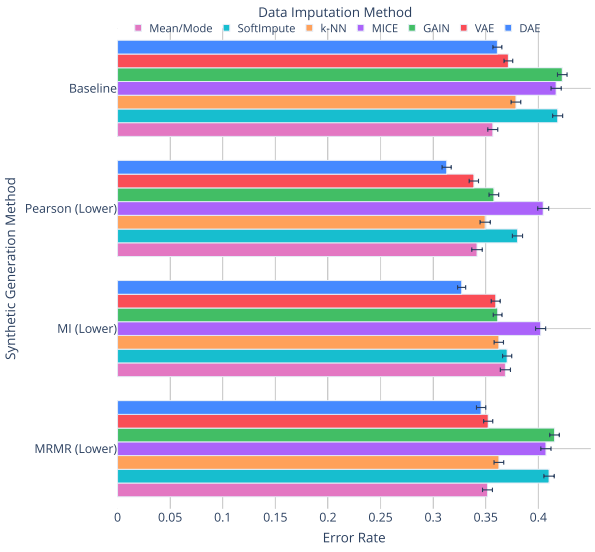


FIGURE 6. Summary of the results of categorical feature and lower masking process.

Figure 7 presents the summary of the results considering the higher values for the masking process. In this scenario, the data imputation models perform similarly to how they did previously. These two comparison highlight the differences in mixed data sets, since categorical features are masked through most frequent approach.

Therefore, as discussed previously, the performance of the proposed methods are aligned in both feature types to the baseline generation method.

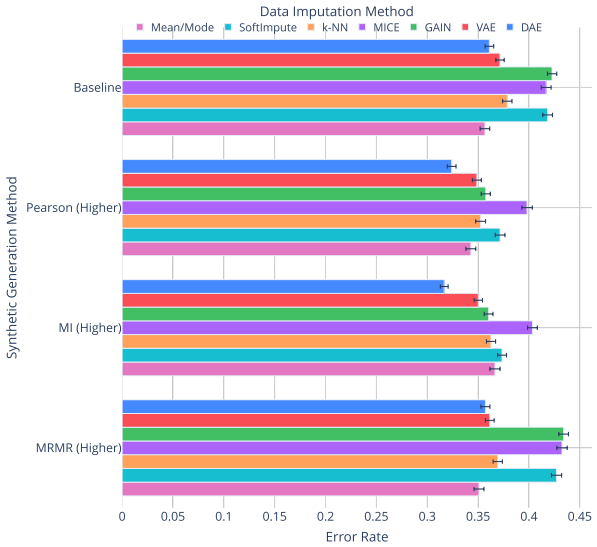


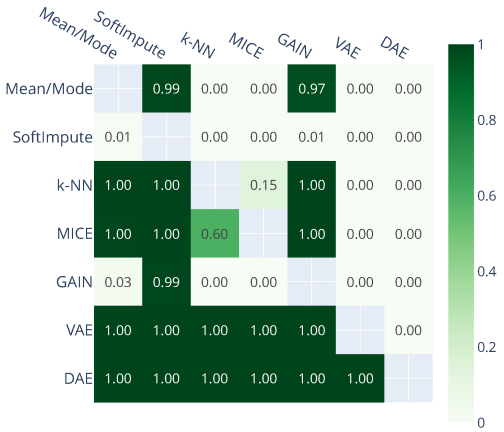
FIGURE 7. Summary of the results of continuous feature and higher masking process.

F. STATISTICAL TESTS

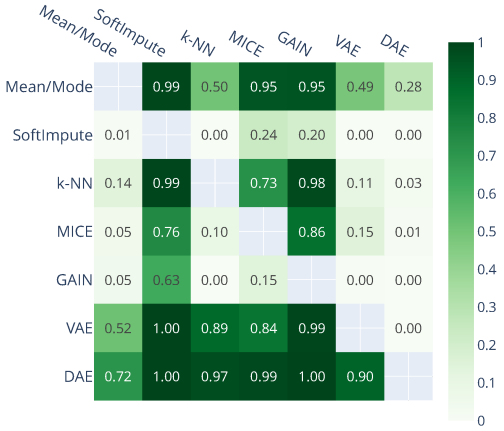
Furthermore, the performance of each data imputation method is compared considering all synthetic generation approaches. This comparison allows to assess which data imputation is more robust to the missingness under MAR mechanism. In order to carry out this comparison several Bayesian statistical tests are conducted [31] by grouping the results by data imputation methods instead of the synthetic generation method. The tests offer advantages like the calculation of probability and setting a threshold. These approaches are versatile enough to be applied to compare multiple models across various data sets.

To provide additional insights, a statistical analysis of models performance was conducted, considering the WMAPE and ErrorRate metrics. In these tests, a threshold is established, and given the chosen metrics, at least 1% of difference is necessary to consider it a significant difference. Thus, the continuous and categorical scores were considered before computing the mean for each data set and data imputation method. In the Figure 8, two heat maps charts are showed. The values represent the probability that the model in row ($Model_A$) outperforms the model in column ($Model_B$). Thus, the greenish the cell, more probable it is that the $Model_A$ is practically better than $Model_B$. For example, MICE is practically superior to k-NN with a probability 60% for continuous features.

Therefore, DAE method achieved the best results for continuous features, followed by VAE and MICE. Between these two methods, VAE outperformed MICE. k-NN method ranks in fourth place because it is practically better than GAIN, Mean/Mode, and SoftImpute. GAIN method only



(a) Statistical test for continuous features



(b) Statistical test for categorical features

FIGURE 8. Statistical tests between data imputation methods.

achieves higher performance than SoftImpute, which it is the worst for continuous features.

In the case of categorical variables, greater diversity is observed. GAIN method achieves similar results to those obtained for continuous features. Therefore, SoftImpute remains the worst method for addressing categorical features. The VAE did not demonstrate significant differences with Mean/Mode since its probabilities are around 50%. This means that it is not possible to conclude which is better. It can be observed that DAE keep showing probabilities above 70% with all methods, which means it was practically better than the other methods. The differences in performance between data imputation models for categorical features are smaller than those for continuous features, in general.

VI. CONCLUSION

This work presents alternative methods for generating synthetic missing data. These three methods employ techniques for evaluating the correlation between features such as Pearson correlation coefficient, Mutual Information and Minimum Redundancy Maximum Relevance. Thus, these

proposals are well fitted to generate missingness under the MAR mechanism. These methods are flexible, since the conditions may vary in terms of the masking process, which allows multiple procedures to be used to select indexes that are subject to missingness.

The proposals were assessed through several data imputation methods and compared with another MAR generation method used as the baseline. The performance is aligned with the baseline in terms of both WMAPE and Error Rate, indicating the validity of the proposals. Regarding the data imputation models used to evaluate the proposals, Autoencoder-based methods and MICE achieved the best results in both variables: continuous and categorical. In particular, the DAE and VAE methods exhibit robust and consistent behavior in the three synthetic missingness generation methods. The superiority of these methods have been confirmed through several Bayesian statistical tests.

Therefore, the conducted benchmark allows giving a perspective on the Missing at Random problem, from synthetic generation to its resolution through data imputation methods. In this sense, the comprehensive comparison of these methods provides insights into which data imputation method is more robust to this missingness mechanism.

In feature research, these methods can be compared to other methods to generate missingness under other mechanisms such as MCAR and MNAR. Thus, an exhaustive benchmark with several missingness generation methods for each mechanism, data imputation methods, and data sets could be proposed.

REFERENCES

[1] R. J. A. Little and D. B. Rubin, *Statistical Analysis With Missing Data*, vol. 32. Hoboken, NJ, USA: Wiley, 1987, p. 70.

[2] R. C. Pereira, P. H. Abreu, and P. P. Rodrigues, "Partial multiple imputation with variational autoencoders: Tackling not at randomness in healthcare data," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 8, pp. 4218–4227, Aug. 2022.

[3] P. J. García-Laencina, P. H. Abreu, M. H. Abreu, and N. Afonoso, "Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values," *Comput. Biol. Med.*, vol. 59, pp. 125–133, Apr. 2015.

[4] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, Nov. 2018.

[5] C. Enders and D. Bandalos, "The relative performance of full information maximum likelihood estimation for missing data in structural equation models," *Struct. Equation Model., A Multidisciplinary J.*, vol. 8, no. 3, pp. 430–457, Jul. 2001.

[6] T. Srebotnjak, G. Carr, A. de Sherbinin, and C. Rickwood, "A global water quality index and hot-deck imputation of missing data," *Ecol. Indicators*, vol. 17, pp. 108–119, Jun. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1470160X1100104X

[7] T. H. Ruggles, D. J. Farnham, D. Tong, and K. Caldeira, "Developing reliable hourly electricity demand data through screening and imputation," *Sci. Data*, vol. 7, no. 1, p. 155, May 2020.

[8] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artif. Intell. Med.*, vol. 50, no. 2, pp. 105–115, Oct. 2010.

[9] E.-L. Silva-Ramírez, R. Pino-Mejías, M. López-Coello, and M.-D. Cubiles-de-la-Vega, "Missing value imputation on missing completely at random data using multilayer perceptrons," *Neural Netw.*, vol. 24, no. 1, pp. 121–129, Jan. 2011.

[10] D. J. Stekhoven and P. Bühlmann, "MissForest—Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, Jan. 2012.

[11] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3911–3919.

[12] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.

[13] Y. Luo, X. Cai, Y. Zhang, J. Xu, and Y. Xiaojie, "Multivariate time series imputation with generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1596–1607. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/96b9bff013acedfb1d140579e2fbeb63-Abstract.html

[14] J. Yoon, J. Jordon, and M. Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5689–5698.

[15] V. Fortuin, D. Baranchuk, G. Ratsch, and S. Mandt, "GP-VAE: Deep probabilistic time series imputation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1651–1661.

[16] M. S. Santos, R. C. Pereira, A. F. Costa, J. P. Soares, J. Santos, and P. H. Abreu, "Generating synthetic missing data: A review by missing mechanism," *IEEE Access*, vol. 7, pp. 11651–11667, 2019.

[17] J. Xia, S. Zhang, G. Cai, L. Li, Q. Pan, J. Yan, and G. Ning, "Adjusted weight voting algorithm for random forests in handling missing values," *Pattern Recognit.*, vol. 69, pp. 52–60, Sep. 2017.

[18] R. M. Schouten, P. Lugtig, and G. Vink, "Generating missing values for simulation purposes: A multivariate amputation procedure," *J. Stat. Comput. Simul.*, vol. 88, no. 15, pp. 2909–2930, Oct. 2018.

[19] D. Wan, R. Razavi-Far, M. Saif, and N. Mozafari, "COLI: Collaborative clustering missing data imputation," *Pattern Recognit. Lett.*, vol. 152, pp. 420–427, Dec. 2021.

[20] S. Jäger, A. Allhorn, and F. Bießmann, "A benchmark for data imputation methods," *Frontiers Big Data*, vol. 4, Jul. 2021, Art. no. 693674. [Online]. Available: https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2021.693674

[21] S. Schelter, T. Rukat, and F. Biessmann, "JENGA—A framework to study the impact of data errors on the predictions of machine learning models," in *Proc. Int. Conf. Extending Database Technol.*, 2021, pp. 529–534. [Online]. Available: https://api.semanticscholar.org/CorpusID:232283615

[22] W. Dong, D. Y. T. Fong, J.-S. Yoon, E. Y. F. Wan, L. E. Bedford, E. H. M. Tang, and C. L. K. Lam, "Generative adversarial networks for imputing missing data for big data clinical research," *BMC Med. Res. Methodol.*, vol. 21, no. 1, p. 78, Dec. 2021.

[23] Y. Sun, J. Li, Y. Xu, T. Zhang, and X. Wang, "Deep learning versus conventional methods for missing data imputation: A review and comparative study," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120201.

[24] B. Zhu, C. He, and P. Liatsis, "A robust missing value imputation method for noisy data," *Appl. Intell.*, vol. 36, no. 1, pp. 61–74, Jan. 2012.

[25] B. Twala, "An empirical comparison of techniques for handling incomplete data using decision trees," *Appl. Artif. Intell.*, vol. 23, no. 5, pp. 373–405, May 2009.

[26] R. C. Pereira, P. H. Abreu, P. P. Rodrigues, and M. A. T. Figueiredo, "Imputation of data missing not at random: Artificial generation and benchmark analysis," *Expert Syst. Appl.*, vol. 249, Sep. 2024, Art. no. 123654.

[27] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked science in machine learning," *ACM SIGKDD Explor. Newslett.*, vol. 15, no. 2, pp. 49–60, 2013, doi: 10.1145/2641190.2641198.

[28] B. Muzellec, J. Josse, C. Boyer, and M. Cuturi, "Missing data imputation using optimal transport," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7130–7140.

[29] U. Garciarena and R. Santana, "An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers," *Expert Syst. Appl.*, vol. 89, pp. 52–65, Dec. 2017.

[30] S. Ramírez-Gallego, I. Lastra, D. Martínez-Rego, V. Bolón-Canedo, J. M. Benítez, F. Herrera, and A. Alonso-Betanzos, "Fast-mRMR: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data," *Int. J. Intell. Syst.*, vol. 32, no. 2, pp. 134–152, Feb. 2017.

[31] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis," *J. Mach. Learn. Res.*, vol. 18, no. 77, pp. 1–36, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-305.html

JUAN-FRANCISCO CABRERA-SÁNCHEZ received the bachelor's and master's degrees in computer science and engineering from the University of Cádiz, where he is currently pursuing the Ph.D. degree. He is also an Invited Assistant Teacher with the University of Cádiz. His major research interests include computational intelligence, data processing, and machine learning-based techniques applied to different areas.

RICARDO CARDOSO PEREIRA received the bachelor's degree in informatics engineering from Coimbra Institute of Engineering and the Ph.D. degree in informatics engineering from the University of Coimbra. He is currently affiliated with the Centre for Informatics and Systems of the University of Coimbra (CISUC). His main research interests include machine learning and data processing, with special emphasis on the missing data field.

PEDRO HENRIQUES ABREU (Member, IEEE) received the Graduate and Ph.D. degrees in informatics engineering from the University of Porto, in 2006 and 2011, respectively. He is currently an Associate Professor with the Department of Informatics Engineering, University of Coimbra; and a Full Member of the Cognitive and Media System Group, Centre for Informatics and Systems, University of Coimbra. He has authored more than 80 publications in highly rated JCR journals and international conferences. His research interests include medical informatics and personal healthcare systems applied to oncology.

ESTHER-LYDIA SILVA-RAMÍREZ received the bachelor's degree in computer science and engineering from the University of Cádiz, and the Engineer and Ph.D. degrees in computer science and engineering from the University of Seville. She is currently an Associate Professor with the Department of Computer Science and Engineering, University of Cádiz. Her major research interests include computational intelligence, data processing, and machine learning-based techniques applied to different areas.