Assessing Adversarial Effects of Noise in Missing Data Imputation

Arthur Dantas Mangussi[1,2][0000−0003−2086−532X], Ricardo Cardoso Pereira[3,4][0000−0003−1735−0771], Pedro Henriques Abreu[4][0000−0002−9278−8194], and Ana Carolina Lorena[1,2][0000−0002−6140−571X]

[1] Computer Science Division, Aeronautics Institute of Techonologies, Praça Marechal Eduardo Gomes, 50, 12228-900, São José dos Campos, Brazil
[2] Science and Technology Institute, Federal University of São Paulo, 12231-280 Talim St. 330, São José dos Campos, Brazil
mangussiarthur@gmail.com,aclorena@ita.br
[3] Miguel Torga Institute of Higher Education, Coimbra, 3000-132, Portugal
[4] Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, University of Coimbra, Coimbra, 3030-290, Portugal
rdpereira@dei.uc.pt,pha@dei.uc.pt

Abstract. In real-world scenarios, a wide variety of datasets contain inconsistencies. One example of such inconsistency is missing data (MD), which refers to the absence of information in one or more variables. Missing imputation strategies emerged as a possible solution for addressing this problem, which can replace the missing values based on mean, median, or Machine Learning (ML) techniques. The performance of such strategies depends on multiple factors. One factor that influences the missing value imputation (MVI) methods is the presence of noisy instances, described as anything that obscures the relationship between the features of an instance and its class, having an adversarial effect. However, the interaction between MD and noisy instances has received little attention in the literature. This work fills this gap by investigating missing and noisy data interplay. Our experimental setup begins with generating missingness under the Missing Not at Random (MNAR) mechanism in a multivariate scenario and performing imputation using seven state-of-the-art MVI methods. Our methodology involves applying a noise filter before performing the imputation task and evaluating the quality of the imputation directly. Additionally, we measure the classification performance with the new estimates. This approach is applied to both synthetic data and 11 real-world datasets. The effects of noise filtering before imputation are evaluated. The results show that noise preprocessing before the imputation task improves the imputation quality and the classification performance for imputed datasets.

Keywords: Missing data imputation · noise filtering.

1 Introduction

Real-world data often presents multiple problems, which can jeopardize the performance of Machine Learning (ML) classifiers [12]. Renggli et al. [16] advocate that data quality issues influence several stages of the ML pipeline. One common type of inconsistency is missing data (MD), which can be described as the absence of values in the data observations [17]. The literature categorizes MD mechanisms into the following categories [14]:

- *Missing Completely at Random* (MCAR): missingness occurs randomly without any dependency on specific features within the dataset;
- *Missing at Random* (MAR): a dependency between existent features determines the missingness nature;
- *Missing Not at Random* (MNAR): the missing values depend on observed and/or other unobserved data (i.e., features not available on the dataset).

To address the MD issue, the literature presents different missing value imputation (MVI) methods, from basic, such as mean, median, and mode, to more sophisticated strategies, including using ML methods to estimate the missing values [15]. Santos et al. [17] describe the classical experimental setup for evaluating MD imputation algorithms with four main steps. The first concerns data collection, where a complete dataset (i.e., without missing values) is considered. Subsequently, the next step is amputation, which introduces artificial missing values, following the characteristics of MCAR, MAR, or MNAR mechanisms. Then, the MVI algorithms need to be selected and applied. Finally, the last step is evaluation. In general, studies in the MD field often evaluate new MVI techniques by measuring the difference between the original (i.e., the ground truth) and imputed data, a process known as direct evaluation. Conversely, there is indirect evaluation, which measures the classification performance using the imputed datasets. However, the literature rarely focuses on evaluating MVI methods using both approaches [9, 6].

Another data quality issue is the presence of noisy instances on the dataset. According to Zhu and Wu [26], there are two distinct types of noise: attribute and class noise. The former type of noise affects input features, while the latter affects the labels registered for the observations. Both are present in real-world scenarios, emphasizing that aggregate noise identification strategies should be considered for both. Nonetheless, the label noise is potentially more harmful than attribute noise [22, 26]. The literature outlines several techniques for identifying potential noise and addressing it to build more reliable ML models from data [3]. Our work will focus on noise filters (NFs), which scan the training data for potentially noisy instances.

Regarding the interplay of MD and noise inconsistencies, when initial noisy data is used to extract patterns for missing data imputation, whether through simple statistics or more sophisticated strategies, the harmful effects of noise can propagate to other instances. Nonetheless, this interaction has received little attention in the missing data literature, and only Zhu et al. [25], Fangfang et al.

[8], and Hulse and Khoshgoftaar [23] have investigated the interaction between these two data inconsistencies. None of these works analyze the application of noise pre-processing before the imputation task. Thus, this work investigates the interaction between missing and noisy data inconsistencies, considering pre-processing on noisy instances before the imputation task and investigating how noisy data impacts the results of MVI methods.

The remainder of this work is organized as follows: Section 2 presents related work on the MD field, and the interplay of missing values and noise data. Section 3 describes the methodology and the experimental setup. The results are presented in Section 4. Section 5 outlines the conclusions and future directions of this work.

2 Related work

This section presents a literature review of imputation techniques, noise in ML, and the interplay of both data inconsistencies.

2.1 Imputation techniques

There are several strategies for handling MD. Replacing the missing values with a predetermined estimate (i.e., imputing the missing values) is a common approach in the literature. The simplest way to perform the imputation task is through a single imputation, where missing values in quantitative features can be replaced with the mean or median of all available non-missing values. In contrast, missing values in qualitative attributes are replaced with the mode [9].

To solve the limitations of single imputation, the literature provides multiple imputation strategies, which employ approximate values that reflect the uncertainty around the actual value from the observed data [2]. The Multivariate Imputation by Chained Equation (MICE) is the most widely used multiple imputation technique [1]. The MICE algorithm is a Gibbs sampler that estimates the posterior distribution of a vector of unknown parameters by sampling iteratively from conditional distributions [20].

Another way to impute MD is using matrix completion methods, such as *Soft-Impute*. The key idea is to perform imputation by approximating the original data with a low-rank matrix. This process typically involves matrix decomposition to identify latent features that best describe the available values [14].

Random Forests (RF) are used for MVI in the *missForest* algorithm [19]. The *missForest* algorithm is an iterative imputation scheme that trains an RF on observed values in the first step, predicts the missing values and proceeds iteratively. Another common ML algorithm used in data imputation is the K-Nearest Neighbor (KNN), which finds the nearest neighbors of instances with missing values and uses them for imputation [6].

The MVI community has recently been using deep learning (DL) methods for MVI [10]. An example is the Generative Adversarial Imputation Nets (GAIN)

[24], which has two main components: generator and discriminator. The generator component imputes the missing values on the observed data. It outputs a complete vector, and the discriminator attempts to validate which element in the output vector is imputed [20, 18]. Autoencoders (AEs) are another example of DL-based methods employed in MVI. AEs are a neural network architecture that learns from incomplete data (input layer) and tries reproducing this input at the output layer, generating new plausible values for imputation [13, 14].

With the development of new MVI techniques, it is essential to evaluate their effectiveness. The literature presents an interesting behavior wherein most works only use the direct evaluation for MVI techniques, where the imputed values are compared to real values in datasets where some values are amputed. But it is also important to evaluate how the imputed values influence classification performance, in an indirect evaluation. Very few works use both approaches.

Pereira et al. [14] evaluate the Siamese Autoencoder-Based Approach for Missing Data Imputation (SAEI) for imputation tasks in direct and indirect ways. For the direct evaluation, the authors used the Mean Absolute Error (MAE) metric, and for the indirect evaluation, they measured the F1-score performance with three different classifiers: KNN, RF, and eXtreme Gradient Boosting (XGB). This was done for 14 datasets. The results show that SAEI outperforms other state-of-the-art MVI methods under MNAR assumption and induces the best classification results, improving the F1-scores for 50% of the used datasets.

Luengo, García, and Herrera [11] analyze the behavior of 23 classification methods and 14 different MVI approaches in an indirect evaluation. Moreover, this methodology was applied to 21 real-world datasets, and all of them have their proper MD. They found that using certain MVI techniques could improve the accuracy obtained for the classification methods, facilitating an explanation of how imputation may be a helpful tool to overcome the negative impact of MD.

2.2 Noise

Noise is frequent in real-world datasets and can harm the predictive performance of ML classifiers. Although most ML algorithms have some internal mechanisms to avoid focusing on noisy instances (e.g. pruning mechanisms in Decision Trees), cleansing such instances can be beneficial [4].

The literature shows various methods to identify and address attribute and label noise. According to Saez et al. [21], Noise filter (NF) techniques are widely used in a data pre-processing step for cleansing the training data [3]. Their strategy consists of identifying potential noise and removing these unreliable examples. However, these inconsistencies can also be corrected [5].

NFs can be divided into two main categories: similarity-based and ensemble-based filters [21]. The similarity-based or distance-based filters employ the KNN algorithm to evaluate whether an example is closest to others within its class; otherwise, it is an unreliable and potentially noisy instance. Various KNN-based methods have emerged in the literature [3]. A well-known example is the Edited

Nearest Neighbor (ENN) NF, which eliminates samples whose class differs from most of its K nearest neighbors [5, 21].

Our focus in this work will be using the ENN algorithm to identify potential noisy instances. These noisy instances will be disregarded during the imputation process to avoid noise from propagating to the imputed data.

2.3 Interplay of missing and noisy data

A few works have have investigated the relationship between noise and MD. Hulse and Khoshgoftaar [23] evaluated the impact of noise in software measurement data on the imputation process. The authors have used five imputation methods using real-world software measurement datasets. The amputation process was made only in the dependent variable covering the MCAR, MAR, and NI (non-ignorable) mechanisms until they achieved 5%, 10%, 15%, and 20% of missing rates. They used five imputation methods on real-world software measurement datasets. The amputation process was applied only to the dependent variable, covering the MCAR, MAR, and NI (non-ignorable) mechanisms, until they achieved missing rates of 5%, 10%, 15%, and 20%. They also considered four different noisy scenarios: inherent noise only (i.e., noise present in the original dataset [23]), no noisy instances, and inherent noise with an additional 5% and 10% of injected noise. For each experimental setup and combination of factors, they conducted five independent random selections and compared the imputation accuracy of the five imputation techniques. Their experiments demonstrated that data quality plays a crucial role in the effectiveness of imputation techniques. Moreover, for the four missing rates used in their experimental setup, an increase in missing rate was not found to be significant for all imputation techniques.

Robust Imputation based on the Group Method of Data Handling (RIBG) is an MVI method proposed by Zhu et al. [25] for predicting missing values in noisy environments. The authors use the Group Method of Data Handling (GMDH), a heuristic self-organizing data mining technique known for its noise immunity, to develop the RIBG method. RIBG operates as follows: given an incomplete dataset, it first performs a preliminary imputation using the mean for numerical features and mode for categorical features to create an initial complete dataset. Then, RIBG applies the GMDH mechanism to iteratively predict and update these initial missing value estimates. To evaluate the effectiveness of the RIBG method, the authors tested it on nine datasets from the UCI repository, with missing rates of 5%, 10%, and 20%, under varying noise levels. The missing data was generated under MCAR, MAR, and MNAR assumptions. The results indicate that noise significantly impacts MVI methods, particularly at high noise levels.

Li et al. [8] present a Noise-Aware Missing Data Multiple Imputation (NPMI) algorithm designed to handle missing data in noisy environments. The NPMI algorithm uses the Random Sample Consensus (RANSAC) method to estimate the initial parameters of the multiple imputation algorithm. This approach enhances the robustness of multiple imputations and ensures accuracy even when noise is present. The proposed method was validated on four datasets: two real and

two synthetic. For the synthetic data, random Gaussian noise was simulated at different noise levels: 5%, 10%, 20%, 30%, 40%, and 50%. For the real data, some values were randomly designated as missing, with missing rates corresponding to the noise level percentages. The accuracy of imputation was evaluated using the Root Mean Square Error (RMSE). The experimental results demonstrated that noise significantly affects the data quality of the entire dataset, with higher noise levels leading to a greater degradation in data quality.

Therefore, previous work has shown that the quality of observed data significantly impacts the imputation task and needs to be addressed. As suggested in studies by [25] and [8], one option is to use algorithms that are robust to noise for the imputation task. However, those works do not analyze if a simple pre-processing for treating noisy instances beforehand impacts missing value imputation. Overall, a comprehensive exploratory analysis of NF use before the imputation task is still needed. Thus, this work aims to investigate the interaction of missing and noisy data inconsistencies and to what extent noisy data impacts the results of MVI methods. From the authors' knowledge, this constitutes the first work that has tried employing an NF before the imputation, investigating the direct and indirect impact of this procedure on imputation quality.

3 Methodology

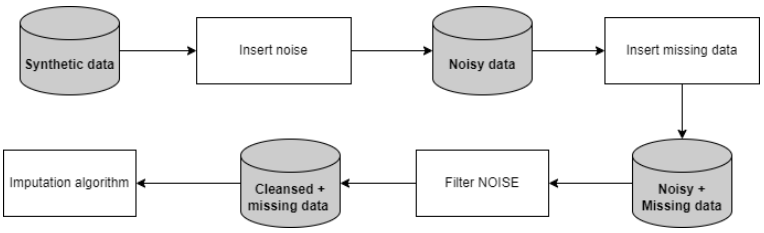

Fig. 1. Methodology overview illustrating the sequential steps followed in the experimental design in the case of synthetic datasets.

The methodology for this work consists of evaluating the interplay between noise and missing data in two points: the imputation quality (direct evaluation) and the classification performance for imputed datasets (indirect evaluation). We perform two types of experiments to achieve our goal: using synthetic data and real-world datasets.

Firstly, we use the synthetic datasets to conduct an initial analysis and have greater control over the experiments. As shown in Figure 1, we generate synthetic data and introduce artificial noise into the attributes or labels. Afterward, we insert missing data in an amputation process (i.e., generate artificial missing values). Then, with a dataset that contains both data inconsistencies, we

use an NF and perform the imputation of the missing values, disregarding the potentially noisy instances. The next section describes how these datasets were generated. A direct evaluation of the imputation quality in these datasets with and without noise filtering is done.

Once the impact of noise on the imputation results of missing values in synthetic data has been assessed, the experiments are extended to real-world datasets currently employed in the related literature, assuming that real-world data already contains some noise level [8]. Therefore, in this case, the artificial insertion of noise at the beginning of Figure 1 is disregarded. Here, both direct and indirect evaluations of the imputation results are assessed.

We have selected eleven real-world heterogeneous benchmark datasets that are currently employed in the MD field and are available on the University of California Irvine Machine Learning Repository[5] and Kaggle[6]. Table 1 overviews dataset characteristics. Each dataset is identified by its acronym name and information on the number of instances, types of features, and classes (i.e., the number of possible output variable values). Categorical features were converted to quantitative values with a one-hot encoding when needed.

As seen in Table 1, the eleven real-world datasets are all binary classification problems. We selected this type of problem due to the methodology employed in this work. We split each dataset according to the two classes to ensure that both classes have the same amount of missing values and to prevent the noise filter from removing an entire class. Section 3.1 will describe this process in more detail.

Table 1. Overview of datasets characteristics.

Dataset	**Instances**	**Features**		**Classes**
		Continuous	**Categorical**	
Wiscosin	569	30	0	2
Pima diabetes	768	8	0	2
Indian liver	583	9	1	2
Parkinsons	195	22	0	2
Mammographic masses	830	1	3	2
Thoracic surgery	470	3	13	2
Diabetic retinopathy	1151	3	16	2
BC Coimbra	116	4	0	2
Thyroid recurrence	383	1	15	2
Blood transfusion	748	4	0	2
Law school	20798	6	6	2

For the implementation, we used Python version 3.11 and several additional libraries: Pandas, Numpy, Scikit-Learn, mdatagen, and Imbalance-Learn. All the experiments were conducted on a machine with 60GB RAM, GPU NVIDIA GeForce RTX 4090 24GB, and Linux, Ubuntu version 22.04.4.

[5] https://archive.ics.uci.edu/datasets
[6] https://www.kaggle.com/datasets

3.1 Experimental setup

As outlined in Figure 1, the first step in our methodology was to create synthetic data. We have used the "make_classification" function from Scikit-Learn to generate the synthetic data. Our base dataset consists of 500 observations with 5 numerical input features and two classes. To introduce attribute noise, Gaussian noise is added to each feature. This noise is generated using the mean and standard deviation of each feature, ensuring the values remain within the feature's minimum and maximum range. Label noise is simulated by flipping the labels. The following rates are tested: 5%, 10%, and 20%. We introduce noise to the entire dataset to obtain a more realistic dataset, as real-world data already contains some noise.

Once noise is inserted, we used a stratified cross-validation strategy with five folds to perform the amputation process (i.e., artificially generating missing values in the dataset) and imputation task. We used the Python library *mdatagen*[7] for each fold to generate artificial MD under MNAR mechanism in a multivariate scenario. To ensure that both classes would receive MD, we split the training set by the outcome and conducted independent procedures for training and testing sets to keep the same missing rates for both. The multivariate MNAR strategy deleted the lowest values for dataset features more related to the classes up to 5%, 10% and 20%. These missing rates were selected from [25].

Using such corrupted datasets, we employ the Edited Nearest Neighbor (ENN) algorithm to filter the potentially noisy instances in the training sets, with $k = 5$. The ENN algorithm cleans the dataset by deleting samples that are close elements from other classes. This tends to remove data in overlapping, borderline and noisy areas of the dataset [7]. In this work, we used a more conservative approach to undersample the majority class, where most of the neighbors must belong to the same class as the examined sample for it to be retained, and the default distance metric in Imbalanced-learn package. The identified noisy instances are not used in data imputation afterwards.

For the imputation task, we chose seven state-of-the-art MVI to address the generated MD. The algorithms chosen are the mean of each feature, KNN, MICE, PMIVAE, missForest, SoftImpute, and GAIN. The KNN, imputation by the mean, missForest, and MICE were used directly from the Scikit-learn library. The remaining algorithms are available in different GitHub repositories[8]. The KNN was used with $K = 5$ and the Euclidean distance, MICE was run with 100 iterations, and the parameterization of the architecture of deep learning methods followed the authors' recommendations from the original articles. As aforementioned, we saved the imputed test data for each fold in the stratified cross-validation strategy. We measured the imputation quality with the Mean Absolute Error (MAE) between the predicted values and the ground truth for the multivariate scenario in the test sets. Next, at the end of the cross-validation

[7] https://pypi.org/project/mdatagen/

[8] https://github.com/travisbrady/py-soft-impute,
https://github.com/jsyoon0823/GAIN
https://github.com/ricardodcpereira/PMIVAE

process, we combined all folds, obtaining an imputed dataset with original data dimensions without bias.

We selected an RF classifier to measure the classification performance of the new datasets. The experiments used the same cross-validation strategy for amputation and imputation processes. We tuned the RF hyperparameters by Randomized Search in the training sets using the F1 Score.

4 Results

This section presents our analysis of the effect of employing an NF based on similarity before the imputation process under MNAR mechanism. We begin by investigating the overall impact of ENN in synthetic data. Subsequently, we analyze the effect of ENN filtering on the imputation task for real-world datasets in direct and indirect evaluations. We also discuss if the pattern found in the experiments using synthetic data is verified in real-world datasets.

4.1 Impact of NF in imputation

Table 2 presents differences in MAE imputation results for synthetic data under the MNAR mechanism. The differences are taken from the baseline, where no noise filtering is applied. Therefore, positive values represent better estimates of the missing values after NF, while negative values denote the opposite. The negative values are boldfaced in the table. The datasets named Att$X$ contain X% of noise in the attributes, where X is a noise rate, while datasets named Label$X$ have X% of label noise. For each one of them, missing rates are also varied.

In all cases, except some specific scenarios with the SoftImpute method, using an NF has improved the MAE results in imputation. Although the differences are small, this demonstrates that even by employing a simple noise filter as ENN before imputation, the imputation quality can be improved. For the mean, MICE, PMIVAE and missForest imputers, the results are improved more for increased missing rates, indicating that a reliable dataset is especially needed for better imputation results when there are large missing rates. In other cases, there is no clear tendency. There is also no clear tendency on which type of noise or noise ratio affects more the MVI results.

Regardless of the characteristics of the SoftImpute method, its primary goal is to find a low-rank matrix that approximates the original dataset [14]. However, the data distribution is altered when we apply an NF that removes specific observations based on the ENN criterion. These new estimates may no longer accurately represent the observed data, leading to the deterioration in the MVI results shown in Tables 2 and 3.

Based on the findings in Table 2, we extend our analysis to real-world datasets, assuming they already have noisy instances.

Table 2. Differences between the average MAE for the baseline where no NF is applied and the application of an NF before imputation under MNAR multivariate conditions for the synthetic datasets. 'Att' is an acronym for attribute noise and 'Label' for label noise. 'Att05' means a 5% noise level introduced as attribute noise, and the same applies for 'Label 05'.

Dataset	Missing Rate	Mean	KNN	MICE	PMIVAE	SoftImpute	GAIN	missForest
Att05	5	0.007	0.055	0.014	0.007	0.001	0.103	0.036
	10	0.011	0.053	0.018	0.011	0.001	0.052	0.035
	20	0.021	0.040	0.024	0.021	0.009	0.062	0.046
Att10	5	0.008	0.047	0.016	0.008	**-0.025**	0.075	0.031
	10	0.012	0.040	0.022	0.012	**-0.032**	0.065	0.032
	20	0.022	0.032	0.029	0.023	**-0.009**	0.041	0.038
Att20	5	0.007	0.044	0.017	0.008	0.039	0.119	0.014
	10	0.013	0.033	0.027	0.012	0.036	0.041	0.024
	20	0.022	0.031	0.028	0.023	0.001	0.076	0.032
Label05	5	0.008	0.027	0.013	0.006	**-0.012**	0.059	0.029
	10	0.012	0.039	0.014	0.012	**-0.012**	0.015	0.016
	20	0.021	0.042	0.016	0.023	0.016	0.030	0.028
Label10	5	0.006	0.024	0.010	0.006	0.000	0.035	0.020
	10	0.011	0.043	0.014	0.009	0.024	0.009	0.020
	20	0.021	0.045	0.019	0.020	0.008	0.017	0.030
Label20	5	0.006	0.042	0.011	0.006	**-0.007**	0.036	0.006
	10	0.013	0.040	0.014	0.013	**-0.007**	0.063	0.010
	20	0.021	0.040	0.020	0.021	**-0.003**	0.029	0.027

4.2 Imputation results for real-world datasets

Table 3 demonstrates the differences in MAE results of employing the previous methodology in real-world datasets under the MNAR mechanism in a multivariate scenario. Again, the differences are taken from the baseline of not employing an NF. Negative differences are boldfaced and indicate the estimates of the missing values are worse when NF is applied. Again, this happens mostly for the SoftImpute algorithm. But now GAIN, missForest, MICE and kNN were also impaired in some specific cases. In most cases, there are improvements, which tend to be higher for larger missing rates. The SoftImpute method shows worse MAE for 44.12% instances and GAIN for 41.18%. For the remaining methods, the percentage of worse results is less than 10%. Therefore, the applied methodology enhances the imputation quality for most MVI techniques on these real-world datasets under the MNAR mechanism. As observed for synthetic data, when the missing rate is increased, the results tend to be improved more when ENN is applied as a pre-processing step.

Table 4 gives a general overview of how each imputation method performs in the datasets by averaging the MAE results per imputation method and missing rate. The best results are boldfaced. The missForest algorithm outperforms the remaining methods, followed by KNN and MICE. The GAIN method was

Table 3. Differences between the average MAE for the baseline where no NF is applied and the application of an NF before imputation under MNAR multivariate conditions for the real-world datasets.

Dataset	Missing Rate	Mean	KNN	MICE	PMIVAE	SoftImpute	GAIN	missForest
wiscosin	5	0.006	0.009	0.001	0.003	**-0.009**	**-0.006**	0.008
	10	0.012	0.014	0.002	0.012	**-0.012**	0.003	0.001
	20	0.022	0.021	0.009	0.021	**-0.008**	0.002	0
pima	5	0.011	0.005	0.01	0.013	**-0.009**	**-0.011**	0.008
	10	0.015	0.007	0.007	0.014	0	**-0.031**	0.017
	20	0.025	0.018	0.018	0.026	**-0.011**	0.035	0.02
indian_liver	5	0.005	0.012	0.005	0.002	**-0.023**	**-0.024**	0.001
	10	0.01	0.016	0.011	0.005	0.032	0.02	0.019
	20	0.023	0.034	0.027	0.015	**-0.002**	0.002	0.029
parkinsons	5	0.004	0.012	0.003	0.002	**-0.008**	0.02	**-0.001**
	10	0.011	0.023	0.007	0.01	0.017	0.046	0.015
	20	0.023	0.026	0.033	0.021	0.031	0.065	0.017
mammographic_masses	5	0.007	0.01	0.003	0.002	0.004	0.065	0.004
	10	0.012	0.033	**-0.006**	0.105	**-0.015**	**-0.015**	0.008
	20	0.028	0.086	0.004	0.063	0.016	0.07	0.015
thoracic_surgery	5	0.006	0.018	**-0.001**	0.001	0	**-0.004**	0.018
	10	0.011	0.02	0.005	0.003	0.015	**-0.006**	0.009
	20	0.022	0.028	0.018	0.006	0.008	0.015	0.011
diabetic_retionapaty	5	0.003	0.013	0.003	**-0.001**	0.008	0.012	**-0.019**
	10	0.006	0.026	0.005	0.005	0.012	0.034	0.013
	20	0.016	0.024	0.012	0.012	0.016	**-0.02**	**-0.009**
bc_coimbra	5	0.007	0.017	0.007	0.006	0.04	**-0.041**	0.025
	10	0.012	0.013	0.005	0.002	**-0.005**	0.026	0.028
	20	0.024	0.029	0.019	0.015	0.012	0.114	0.037
thyroid$_{recurrence}$	5	0.007	0.008	0.018	0.007	**-0.005**	0.058	0.022
	10	0.014	0.017	0.011	0.018	**-0.005**	**-0.046**	0.014
	20	0.027	0.025	0.024	0.018	0.004	**-0.011**	0.016
blood_transfusion	5	0.004	**-0.015**	0	0.002	0.006	**-0.115**	0.004
	10	0.01	0.015	0.012	0.009	**-0.004**	0.111	0.009
	20	0.02	0.03	0.027	0.019	**-0.003**	**-0.017**	0.041
law	5	0.008	0.022	0.004	0.004	0.012	**-0.058**	0.008
	10	0.017	0.056	0.007	0.008	**-0.012**	0.071	0.025
	20	0.033	0.091	0.074	0.077	0.006	0.023	0.086

the worst method in this experimental setup, and the PMIVAE only surpassed the imputation results by mean. The complexity of those methods may justify the need for larger datasets to obtain better results in the experiments. Since the ENN algorithm further reduces the datasets in imputation, this might have degraded their performance more.

Taking the three best imputation methods, missForest, KNN, and MICE, respectively, we also analyze the average difference between applying or not the methodology of this work, for different missing levels. The results are shown in Table 5. The average difference increases for higher missing rates, confirming the results observed for synthetic data.

Indirectly evaluating the imputation quality, we trained an RF classifier for the newly imputed data and compared its F1 score with the baseline (i.e., the F1-score for original datasets, without amputed values). Figure 2 illustrates the overall average results across all imputed real-world datasets and missing rates, where the average F1-score achieved for the original complete datasets is shown

Table 4. Average MAE results obtained for each imputation method, grouped by missing rate. The highlighted bold results present the best MAE results for each missing rate.

Missing Rate	Mean	KNN	MICE	PMIVAE	SoftImpute	GAIN	missForest
5%	0.233	0.142	0.145	0.221	0.168	0.481	**0.139**
10%	0.224	0.146	0.147	0.209	0.164	0.442	**0.130**
20%	0.211	0.158	0.148	0.196	0.162	0.467	**0.132**

Table 5. Average differences of MAE grouped by missing rate for the three best-performing imputation methods.

Missing Rate	KNN	MICE	missForest
5%	0.010	0.005	0.007
10%	0.022	0.006	0.014
20%	0.037	0.024	0.024

as a dashed red line. MICE outperformed the remaining MVI methods and got closer average to the baseline, followed by KNN and missForest. Those methods were the best three imputation algorithms in our previous evaluation and benefited more from noise filtering. Therefore, we can observe an interplay with better quality of imputation and classification performances. In contrast, the PMIVAE method presents a worse F1-score than the other methods. Also, PMIVAE only outperforms the mean and GAIN in MAE imputation quality. We believe that the nature of ENN's undersampling process, combined with the high dependency on parameterization for deep learning methods such as PMIVAE and GAIN, explains the results observed in our study. Therefore, we plan to conduct an optimization search to identify the best parameters for these methods in the future. Additionally, due to its specific characteristics, the PMIVAE imputation method may not yield good results with the RF classifier. We intend to extend our methodology to include other classification algorithms to address this.

In conclusion, most imputation strategies benefited from noise filtering, estimating new values with higher quality. However, the results are impaired for some specific algorithms. MAE is reduced for SoftImpute, GAIN, and PMIVAE. PMIVAE is especially impaired in the indirect evaluation. These algorithms may require more data to obtain a proper fit of values to be imputed. Since noise filtering reduces the dataset, this can explain the observed behavior.

5 Conclusions

This study investigated the interaction between missing and noisy data, focusing on pre-processing noisy instances before imputation and how this procedure impacts MVI methods. This is the first comprehensive analysis of its kind, addressing a gap in the related literature. We used the Edited Nearest Neighbors (ENN) noise filter and seven imputation methods: mean, KNN, MICE, PMIVAE,

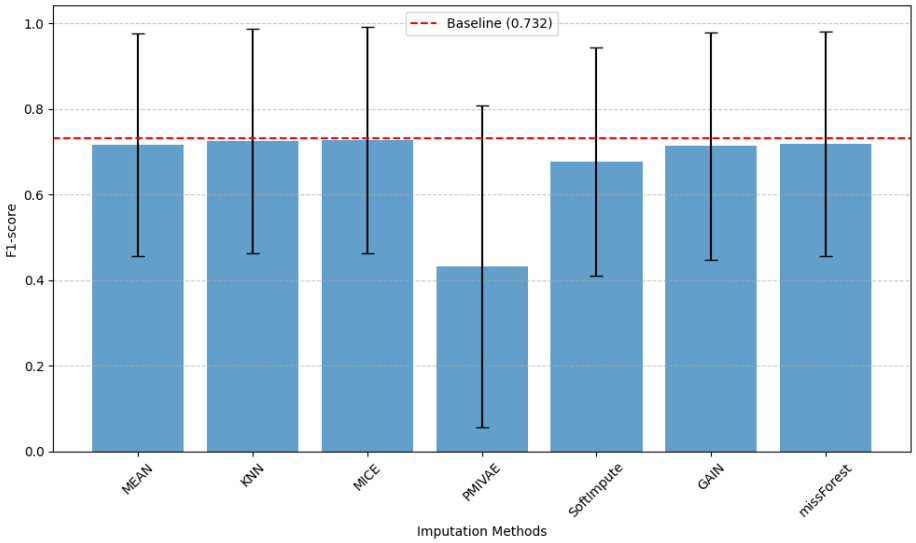

Fig. 2. Overall F1 score of all imputation methods for Random Forest classification model.

SoftImpute, GAIN, and missForest. The filtering takes place before imputation and the hypothesis is that disregarding the unreliable instances from the imputation process can be beneficial, as it prevents noise from propagating to the new estimated values.

Applying the noise filter has generally improved estimates of the missing values for both synthetic and real-world datasets. The imputation methods that benefited the most from this approach were missForest, KNN, and MICE. And the results are improved more for datasets with higher missing rates. Nonetheless, the reduction in the datasets used for adjusting the models seems to have impaired the results for more complex techniques. However, this must be confirmed for larger datasets with more instances.

For future considerations of this work, we want to extend the methodology for the MAR multivariate mechanism and cover other classifiers and noise filters more complex than ENN.

Acknowledgments. The authors gratefully acknowledge the Brazilian funding agencies FAPESP (Fundação Amparo à Pesquisa do Estado de São Paulo) under grants 2022/10553 -6, 2023/13688-2 and 2021/06870-3. This research was also supported by the Portuguese Recovery and Resilience Plan (PRR) through project C645008882-00000055 - Center for Responsable AI.

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

References

1. Buuren, S., Groothuis-Oudshoorn, C.: Mice: Multivariate imputation by chained equations in r. Journal of Statistical Software **45** (12 2011). https://doi.org/10.18637/jss.v045.i03
2. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., Tabona, O.: A survey on missing data in machine learning. Journal of Big Data **8**(1) (Dec 2021). https://doi.org/10.1186/s40537-021-00516-9, funding Information: This work received a grant from the Botswana International University of Science and Technology. Publisher Copyright: © 2021, The Author(s).
3. Frenay, B., Verleysen, M.: Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems **25**(5), 845–869 (2014). https://doi.org/10.1109/TNNLS.2013.2292894
4. Garcia, L.P., de Carvalho, A.C., Lorena, A.C.: Effect of label noise in the complexity of classification problems. Neurocomputing **160**, 108–119 (2015)
5. Garcia, L.P., Lehmann, J., de Carvalho, A.C., Lorena, A.C.: New label noise injection methods for the evaluation of noise filters. Knowledge-Based Systems **163**, 693–704 (2019). https://doi.org/https://doi.org/10.1016/j.knosys.2018.09.031, https://www.sciencedirect.com/science/article/pii/S0950705118304829
6. Hasan, M.K., Alam, M.A., Roy, S., Dutta, A., Jawad, M.T., Das, S.: Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). Informatics in Medicine Unlocked **27** (11 2021). https://doi.org/10.1016/j.imu.2021.100799
7. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research **18**(17), 1–5 (2017), http://jmlr.org/papers/v18/16-365
8. Li, F., Sun, H., Gu, Y., Yu, G.: A noise-aware multiple imputation algorithm for missing data. Mathematics **11**, 73 (12 2022). https://doi.org/10.3390/math11010073
9. Lin, W.C., Tsai, C.F.: Missing value imputation: a review and analysis of the literature (2006–2017). Artificial Intelligence Review **53**, 1487–1509 (02 2020). https://doi.org/10.1007/s10462-019-09709-4
10. Liu, M., Li, S., Yuan, H., Ong, M.E.H., Ning, Y., Xie, F., Saffari, S.E., Shang, Y., Volovici, V., Chakraborty, B., Liu, N.: Handling missing values in healthcare data: A systematic review of deep learning-based imputation techniques. Artificial Intelligence in Medicine **142**, 102587 (2023). https://doi.org/https://doi.org/10.1016/j.artmed.2023.102587, https://www.sciencedirect.com/science/article/pii/S093336572300101X
11. Luengo, J., García, S., Herrera, F.: On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowledge and Information Systems **32**, 77–108 (2012). https://doi.org/https://doi.org/10.1007/s10115-011-0424-2
12. Nakhaei, A., Sepehri, M.M., khatibi, t.: A promising method for correcting class noise in the presence of attribute noise. International Journal of Hospital Research **12**(1), – (2023). https://doi.org/LBL_COMMENTED_AT/ijhr.2023.383118.1535, https://ijhr.iums.ac.ir/article_171438.html
13. Pereira, R.C., Abreu, P.H., Rodrigues, P.P.: Siamese autoencoder-based approach for missing data imputation. In: International Conference on Computational Science. pp. 33–46. Springer (2023)

14. Pereira, R.C., Abreu, P.H., Rodrigues, P.P.: Siamese autoencoder architecture for the imputation of data missing not at random. Journal of Computational Science **78**, 102269 (2024). https://doi.org/https://doi.org/10.1016/j.jocs.2024.102269, https://www.sciencedirect.com/science/article/pii/S1877750324000620
15. Pereira, R.C., Rodrigues, P.P., Figueiredo, M.A.T., Abreu, P.H.: Automatic delta-adjustment method applied to missing not at random imputation. Computational Science – ICCS 2023 pp. 481–493 (2023). https://doi.org/10.1007/978-3-031-35995-8_34
16. Renggli, C., Rimanic, L., Gürel, N., Karlaš, B., Wu, W., Zhang, C.: A data quality-driven view of mlops. IEEE Transactions on Knowledge and Data Engineering (03 2021)
17. Santos, M.S., Pereira, R.C., Costa, A.F., Soares, J.P., Santos, J., Abreu, P.H.: Generating synthetic missing data: A review by missing mechanism. IEEE Access **7**, 11651–11667 (2019)
18. Shahbazian, R., Greco, S.: Generative adversarial networks assist missing data imputation: A comprehensive survey and evaluation. IEEE Access **11**, 88908–88928 (2023). https://doi.org/10.1109/ACCESS.2023.3306721
19. Stekhoven, D., Bühlmann, P.: Missforest?non-parametric missing value imputation for mixed-type data. Bioinformatics (Oxford, England) **28**, 112–8 (01 2012). https://doi.org/10.1093/bioinformatics/btr597
20. Sun, Y., Li, J., Xu, Y., Zhang, T., Wang, X.: Deep learning versus conventional methods for missing data imputation: A review and comparative study. Expert Systems with Applications **227**, 120201 (04 2023). https://doi.org/10.1016/j.eswa.2023.120201
21. Sáez, J.A.: Noise models in classification: Unified nomenclature, extended taxonomy and pragmatic categorization. Mathematics **10**(20) (2022). https://doi.org/10.3390/math10203736, https://www.mdpi.com/2227-7390/10/20/3736
22. Sáez, J.A., Galar, M., Luengo, J., Herrera, F.: Analyzing the presence of noise in multi-class problems: Alleviating its influence with the one-vs-one decomposition. Knowledge and Information Systems **38**, 179–206 (01 2014). https://doi.org/10.1007/s10115-012-0570-1
23. Van Hulse, J., Khoshgoftaar, T.: A comprehensive empirical evaluation of missing value imputation in noisy software measurement data. Journal of Systems and Software **81**, 691–708 (05 2008). https://doi.org/10.1016/j.jss.2007.07.043
24. Yoon, J., Jordon, J., van der Schaar, M.: Gain: Missing data imputation using generative adversarial nets. In: International Conference on Machine Learning (ICML). pp. 5689—-5698 (2018)
25. Zhu, B., He, C., Liatsis, P.: A robust missing value imputation method for noisy data. Appl. Intell. **36**, 61–74 (01 2012). https://doi.org/10.1007/s10489-010-0244-1
26. Zhu, X., Wu, X.: Class noise vs. attribute noise: A quantitative study. Artif. Intell. Rev. **22**, 177–210 (11 2004). https://doi.org/10.1007/s10462-004-0751-8