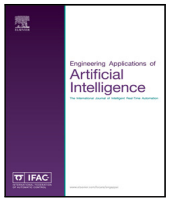
Engineering Applications of Artificial Intelligence 111 (2022) 104791

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



# The impact of heterogeneous distance functions on missing data imputation and classification performance

Miriam Seoane Santos [a,*], Pedro Henriques Abreu [a], Alberto Fernández [b], Julián Luengo [b], João Santos [c,d]

[a] University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, Coimbra, Portugal
[b] Department of Computer Science and Artificial Intelligence, University of Granada, Spain
[c] Instituto de Ciências Biomédicas Abel Salazar da Universidade do Porto, Portugal
[d] IPO-Porto Research Centre (CI-IPOP), Porto, Portugal

ARTICLE INFO

*Keywords:*
Missing data
Data imputation
kNN
Distance functions
Heterogeneous data

ABSTRACT

This work performs an in-depth study of the impact of distance functions on K-Nearest Neighbours imputation of heterogeneous datasets. Missing data is generated at several percentages, on a large benchmark of 150 datasets (50 continuous, 50 categorical and 50 heterogeneous datasets) and data imputation is performed using different distance functions (HEOM, HEOM-R, HVDM, HVDM-R, HVDM-S, MDE and SIMDIST) and $k$ values (1, 3, 5 and 7). The impact of distance functions on kNN imputation is then evaluated in terms of classification performance, through the analysis of a classifier learned from the imputed data, and in terms of imputation quality, where the quality of the reconstruction of the original values is assessed. By analysing the properties of heterogeneous distance functions over continuous and categorical datasets individually, we then study their behaviour over heterogeneous data. We discuss whether datasets with different natures may benefit from different distance functions and to what extent the component of a distance function that deals with missing values influences such choice. Our experiments show that missing data has a significant impact on distance computation and the obtained results provide guidelines on how to choose appropriate distance functions depending on data characteristics (continuous, categorical or heterogeneous datasets) and the objective of the study (classification or imputation tasks).

## 1. Introduction

Real-world domains are often afflicted by Missing Data (MD), i.e., absent information in datasets for which the respective values are unknown. This severely compromises the performance of most classification models, which either (i) cannot internally handle missing information or (ii) result in the definition of misguided decision boundaries (Lin and Tsai, 2020). Over the years, several approaches have been discussed to surpass this issue, among which machine learning-based imputation stands out as the most popular (García-Laencina et al., 2010). It consists of replacing the absent values with plausible estimates taken from the complete training data portion and, contrarily to other approaches, does not require the elimination of instances with missing values, is model-agnostic (i.e., it does not require that data distributions are modelled by some procedure), and independent of the final classification task, i.e., past the imputation stage, the classification task can be addressed by any classifier.

Among machine learning-based imputation strategies, k-Nearest Neighbours Imputation (kNNI), since its proposal in the yearly 00's (Troyanskaya et al., 2001), remains one of the most popular and competitive approaches (Lin and Tsai, 2020), and is a widely-used solution across several application domains (Tabassian et al., 2016; Sun et al., 2017; Huang et al., 2017; Abnane et al., 2019; Fu et al., 2019), especially those that require a strong notion of pattern similarity, such as healthcare domains (Jerez et al., 2010; Santos et al., 2015; García-Laencina et al., 2015; Huang et al., 2016). Essentially, kNNI is based on the intuitive principle of associating the distance between two patterns to the likelihood of their values being similar. Accordingly, for a given pattern with missing information, the imputation process involves finding its most similar neighbours and use their information to produce an estimate for the missing values. Beyond its simplicity, kNNI possesses other desirable traits: it is a nonparametric method which does not require any assumptions on the data (Tutz and Ramzan, 2015), can predict both continuous and categorical features (Batista and Monard, 2003), has proven to preserve the data distribution (Santos et al., 2017), and allows for a great interpretability and explainability (Amorim et al., 2018). Also,

* Corresponding author.
*E-mail address:* miriams@dei.uc.pt (M.S. Santos).

https://doi.org/10.1016/j.engappai.2022.104791
Received 25 September 2021; Received in revised form 14 January 2022; Accepted 25 February 2022
Available online xxxx
0952-1976/

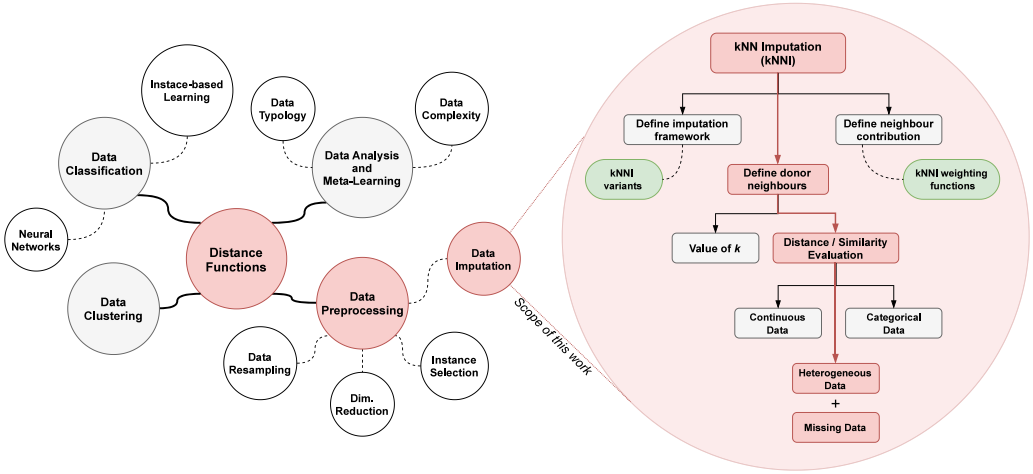



**Fig. 1.** Distance functions are embedded in several fields of machine learning, enhancing the performance of similarity-based algorithms, either in data classification, data analysis, data preprocessing, or data clustering. The scope of this work is concerned with data imputation (kNN imputation in particular), where distance functions are used to evaluate the similarity between patterns in order to find suitable donor neighbours to produce plausible estimates for missing values. The distance functions considered in this work incorporate both the computation of heterogeneous data (continuous and categorical data), as well as missing data, and can be further examined in any other domains that rely on distance metric learning.

contrarily to most machine learning-based imputation strategies, kNNI is a lazy learner, i.e., it does not require the creation of an explicit predictive model for each missing feature (Batista and Monard, 2003). Therefore, it can directly handle instances with multiple missing values and the adjustment to new training data is performed continuously, without the need to retrain predictive models. Provided with thoughtful adaptations, it even has the potential to accommodate more complex problems (e.g., concept drifts Zhang et al. (2010)).

Nevertheless, the efficiency of kNNI is largely conditioned by certain challenging factors (Fig. 1). One relies on the definition of suitable donor neighbours, which in turn implies the choice of both an appropriate number of neighbours, $k$, and a distance function, $D$. Other impactful decisions concern the definition of the imputation framework, i.e., kNNI variants (e.g., iterative, sequential, cluster-based, incomplete case-based Brás and Menezes (2007), Kim et al. (2004), Hruschka et al. (2004) and Van Hulse and Khoshgoftaar (2014)) and/or the strategy to weight the contribution of each neighbour to the final missing value estimate, i.e., kNNI adaptations or weighting schemes (e.g., mean/mode, distance-weighted, rank-weighted Dudani, 1976; Huang et al., 2017). However, note that while kNNI frameworks/variants and adaptations/weighting schemes can be thought as general modifications of the traditional kNNI formulation, the definition of both a donor set and a distance function is a mandatory requirement. Nonetheless, and although all these aspects contribute to the successful behaviour of kNNI, they have not received the same attention in related research over the past decades. Whereas tuning the optimal number of $k$ nearest neighbours, or experimenting with several possible values for improved results is nowadays a standard practice across most imputation papers (García-Laencina et al., 2010, 2015; Pan et al., 2015), and increasing research has been investigating the effect of applying different kNNI weighting adaptations and variants (García-Laencina et al., 2009; Luengo et al., 2012; Tutz and Ramzan, 2015; Jiang and Yang, 2015; Al-Helali et al., 2021), the search for a suitable distance function remains often neglected (related work is presented in Section 2).

This is true both from a imputation as well as classification perspective (kNN classification), among other related fields (Fig. 1), and is perhaps due to the existing lack of insight regarding the behaviour of different distance functions. Note that the chosen value of $k$ is directly associated to a local or global nature of kNN, as it relates to the size of the neighbourhood considered for imputation or classification. Naturally, smaller values of $k$ define stricter imputation estimates or classification rules, focusing on a local perspective of the domain. In turn, weighting functions control the impact that the patterns in the defined neighbourhood have in determining the final imputed value or class label. Ultimately, there is also some intuition on appropriate weighting functions, depending on the characteristics of data. For instance, overlapped domains or domains presenting certain structural biases should respond better to weighted imputation approaches: this is not only intuitive as it is also empirical, since the fact that distance metric learning is inherent to a broad spectrum of fields in machine learning (Suárez et al., 2021) (Fig. 1), makes it possible to exchange empirical knowledge between different areas and applications (e.g., overlapped domains should benefit from weighted imputation approaches in the same way they benefit from weighted resampling and classification approaches Nekooeimehr and Lai-Yuen (2016) and Rastin et al. (2021)).

For distance functions, however, it has been difficult to derive some underlying principles that motivate the choice of one distance function over another. For the most part, existing approaches – both within the scope of kNN imputation and classification – often rely on variations of the Minkowski distance, where the Euclidean distance is the most frequently used by default (Luengo et al., 2012; Gou et al., 2019a; Fouad et al., 2021; Al-Helali et al., 2021). However, note that distance functions are not universally suited to all types of data. Variations of the Minkowski distance, such as the Manhattan and Euclidean distances, work under the assumption of continuous data. Other distance functions are more appropriate to handle categorical data, such as the Jaccard or the Value Difference Metric (VDM) distances.

Inevitably, heterogeneous data, comprising both continuous and categorical features, requires special treatment. Essentially, there are three main solutions for heterogeneous data. A common solution is to transform features so that they are represented on the same scale (Lumijärvi et al., 2004). Accordingly, continuous features may be discretised to categorical, or categorical features may be transformed to binary, using a 1/0 encoding (one-hot encoding) for each existing category (which allows arithmetic operations over values). These solutions are however suboptimal: on the one hand, determining an adequate number of categories for the discretisation of continuous features is not trivial. Besides, if categories are considered nominal, the order information is lost. One the other hand, one-hot encoding may significantly increase data dimensionality which adds time and memory complexity to kNNI. Another possibility is to combine distance functions in order to address the continuous and categorical portions separately. This, however, often results in considering a binary encoding for certain categorical features (nominal) and the use of matching coefficients

between the transformed binary vectors (Ali et al., 2019). A more refined approach is to consider heterogeneous distance functions that directly handle different types of features, thus avoiding the problems described above (Wilson and Martinez, 1997).

Yet, there is another factor that needs to be accounted for: the incorporation of missing data in the distance computation. Traditional implementations of kNNI require that donor neighbours have observed values in all features. Other kNNI variants allow the donors to have some missing information, although they are required to have the same observed features as the pattern with missing values (plus observed information in the feature to impute) (Van Hulse and Khoshgoftaar, 2014). In other frameworks, the donors are allowed to have missing values, although the computation of distances does not use all the features, but only those for which observations are available in both instances (Tutz and Ramzan, 2015).

One of the advantages of considering heterogeneous distance functions is that they are flexible in incorporating operations on missing values as well. Additionally, it is possible to handle absent values differently, depending on whether they belong to a continuous or categorical feature. This allows that all existing information is considered for imputation, without discarding any data patterns or values. Finally, it allows that the presence of missing data itself is also considered in the distance computation, i.e., the uncertainty of the missing data can be accounted for: patterns comprising missing values in the same feature can either be thought to be closer (more similar) or farther from each other (less similar), or evaluated according to intermediate strategies. Popular heterogeneous distance functions, such as the Heterogeneous Euclidean-Overlap Metric (HEOM) or the Heterogeneous Value Difference Metric (HVDM) (Wilson and Martinez, 1997), consider that the distance between two values should be maximal if either of them is missing, while other definitions are more flexible (details are given on Section 4). Intuitively, we realise that missing data, their distribution among existing classes, percentage, and the rules that define their comparison will affect distance computation and consequently, kNN-based approaches, independently of the end goal (imputation, classification, clustering, resampling). In this work, we focus on kNN imputation to address complex scenarios comprising heterogeneous data – continuous and categorical (nominal and binary) features – and missing data, where the absent values themselves are incorporated in distance computation.

### 1.1. Objectives and contributions

This work follows from our recent research (Santos et al., 2020b), where we have shown that distance functions affect kNN imputation. Nevertheless, some topics remained unaddressed from the preliminary experiments. The study considered 61 datasets, although there was not a clear division between categorical and heterogeneous datasets, and continuous datasets comprised the great majority (37 datasets). Finally, only $k = 1$ was investigated and no analysis regarding imputation quality was performed.

Herein, we perform a more in-depth study of the impact of different heterogeneous distance functions on kNN imputation, both in terms of classification performance and imputation quality. Note that our objective is not to select an extensive set of possible distance functions and tune the performance of classifiers with respect to each dataset, i.e., test all possible distance functions and look for the solution that maximises classification or imputation results. On the contrary, we aim to provide a thoughtful selection of distance functions, with distinct approaches to continuous, categorical and missing data, and study the properties of each component in order to generate some insight on their behaviour. Accordingly, rather than searching for optimal results, i.e., test every approach and select the best, we aim to provide insights, i.e., some intuition over the imputation process that may ultimately lead to more informed decisions regarding the choice and application of distance functions. In sum, the contributions of this work are the following:

- A study of the impact of distance functions of kNNI and its effect on classification performance, by comparing classification models trained with datasets imputed with different distance functions;
- A thorough investigation of the behaviour of heterogeneous functions, namely how each component – treatment of continuous, categorical and missing values – affects the computation of distances (and consequently the classification results), extrapolating insights for heterogeneous datasets;
- A comparison between different downstream tasks (classification *versus* imputation), studying the impact of distance functions on the quality of imputation, besides classification performance. While on the previous cases the imputation task is seen as an auxiliary task that helps to model the classification task, here we also focus on the imputation task and evaluate distance functions regarding their ability to reconstruct the original, true values in data.

Our experiments show that distance functions play an important role on the optimisation of both classification and imputation tasks, missing data has a significant impact on distance computation, and the obtained results provide some insights on how to choose appropriate distance functions depending on data characteristics (continuous, categorical or heterogeneous datasets) and the objective of the study (classification or imputation). Below we summarise further contributions of this research work:

- We present a extensive experimental setup, with 150 datasets (50 continuous, 50 categorical and 50 heterogeneous datasets), where imputation is performed under several missing rates (5, 10, 20 and 30%), with 7 different heterogeneous distance functions that incorporate missing values computation — HEOM, HEOM-R, HVDM, HVDM-R, HVDM-S, MDE and SIMDIST (Section 4), and several values of $k$ (1, 3, 5 and 7). To our knowledge, no study has performed such a comprehensive data collection and analysis so far;
- We evaluate distance functions both regarding classification performance and imputation quality, whereas related work is often focused solely on one perspective, mostly the effect of kNNI on classification performance (Batista and Monard, 2003; Farhangfar et al., 2008; Luengo et al., 2012; Hu et al., 2016; Huang et al., 2016; Tsai and Chang, 2016);
- We analyse whether datasets with different characteristics may benefit from different distance functions and to what extent the component of a function definition that handles missing data influences such choice, by comparing different solutions for missing value incorporation in distance computation. Such an analysis derives important insights on the behaviour of distance functions and has not been previously touched upon in previous research;
- We extend our previous research (Santos et al., 2020b) in what concerns the number and characteristics of datasets, kNNI parametrisation, and also including the analysis of imputation quality. This improved experimental setup allows a more thorough theoretical and empirical analysis of the properties and behaviour of the considered distance functions.
- In order to foster the study of heterogeneous distance functions for both data imputation and general purposes in Engineering Applications, we put forward the MATLAB implementations of the distance functions studied in this work, publicly available on GitHub.[1]

### 1.2. Potential and engineering applications

Given the heterogeneity of data associated to most real-world domains and their susceptibility to missing data, data imputation becomes

[1] https://github.com/miriamspsantos/heterogeneous-distance-functions.

a central issue across several engineering problems and applications, where kNNI is regarded as the most representative algorithm among machine learning-based techniques (Lin and Tsai, 2020; Triguero et al., 2019; Tlamelo et al., 2021).

One of its most common applications is perhaps in the field of medical informatics and biomedical engineering (Abreu et al., 2016; García-Laencina et al., 2015; Lin and Tsai, 2020; Santos et al., 2020a), where erroneous predictions may have serious implications in people's lives, and therefore is its crucial to guarantee the quality of data. In addition, in these contexts it is also fundamental to guarantee data representativeness, particularly if data suffers from additional complicating factors (e.g., if the data is scarce or imbalanced Santos et al., 2015). In such scenarios, it is important that expert systems analyse the similarity between cases (here, patients), so that the estimate values obtained from the imputation process are not biased towards the most represented concepts. In other words, it is important that the imputation process is adjusted to each patient's characteristics, by analysing the information available from the most similar clinical cases, rather than considering the entire dataset. It comes therefore as no surprise that kNNI has become very popular in healthcare domains.

Nevertheless, healthcare problems (e.g., survival prediction, disease diagnosis and prognosis) are just one of the many application domains where similarity learning is crucial to devise optimal solutions. In fact, beyond the scope of data imputation, kNN has become a core algorithm across a wide range of fields and applications and is ultimately one of the most promising techniques to move towards Smart Data (Triguero et al., 2019). The fundamental basis of kNN is its ability to handle pattern similarity, which primarily results from an appropriate definition of distance functions. Accordingly, although this study is concerned with kNNI, the derived insights may be further extrapolated and explored across other frameworks and applications, not only in the scope of data imputation, but across a wider panorama of machine learning fields relying on distance metric learning (Fig. 1).

Fig. 1 presents a plethora of machine learning fields operating with similarity computation, where the distance functions studied in this work may be investigated. To further systematise the application potential of this study, Table 1 provides the reader with an explanation of how these distance functions may be incorporated both in the scope of data imputation as well as across the remaining areas, along with their frequent engineering applications.

Considering data imputation, distance functions can be applied to measure pattern similarity as an intermediate step to improve kNNI or other imputation approaches, namely via instance selection (Tsai and Chang, 2016; Huang et al., 2016; Pereira et al., 2020). Note that instance selection can also be used outside the scope of data imputation (e.g., cleaning approaches Smith et al., 2014), yet still recurring to distance functions (Tsai et al., 2019).

Another straightforward application with respect to data preprocessing is data resampling. Considering the field of Imbalanced Data, there is a plethora of data resampling algorithms that rely on distance computation, either undersampling or oversampling algorithms. Distance computation is fundamental to determine which patterns to clean/remove from data, or which patterns are suitable candidates for synthetic data generation, respectively. As an example, the original formulation of the well-known Synthetic Minority Oversampling Technique (SMOTE) considers the Euclidean distance (Chawla et al., 2002), although HEOM or HVDM are frequently used with heterogeneous data (Napierała et al., 2010; Santos et al., 2015; Napierala and Stefanowski, 2016; Wilk et al., 2016; Borowska and Stepaniuk, 2016). Using a distance function that is suited to the nature of data allows the construction of a training set that is more representative of the domain, consequently improving the performance of classifiers trained over this set.

Regarding data classification, suitable applications comprise the modification of algorithms operating with distances among patterns, such as instance-based learning, radial basis function networks, or self-organising maps (Weinberger and Saul, 2009; Parameswaran and Weinberger, 2010; Negri and Belanche, 2001) (which can also be used for data imputation).

Data clustering is also a standard application domain, where finding an appropriate way of computing similarity between patterns is key for the success of methods (Harikumar and Surya, 2015; Kalra et al., 2018).

In the field of Data Analysis and Meta-Learning, distance or similarity computation is also on the basis of several well-known data complexity measures (neighbourhood measures in particular) (Lorena et al., 2019). Another example is the characterisation of datasets via their data typology, i.e., the categorisation of examples into several types. Originally, data typology relies on the HVDM distance (Napierala and Stefanowski, 2016), although recent research has started investigating the effect of different distance metrics on the typology results (Mahin et al., 2018, 2019). This is yet another example where considering the solutions for heterogeneous data with missing values explored in this work could potentially improve results.

In sum, given the extent to which distance metric learning is used across several fields of machine learning and the data heterogeneity encountered in most real-world domains (comprising different types of features, missing values, and other difficulty factors), there are a plenitude of applications and extensions that can be derived from the solutions studied in this work, despite its focus on data imputation.

The paper is structured as follows. First, Section 2 discusses previous work regarding the use of distance functions with k-Nearest Neighbours as an imputation algorithm and as a classifier. Then, Sections 3 and 4 discuss some background on missing data theory and describe the studied heterogeneous distance functions, respectively. Section 5 describes the experimental setup in detail, while Sections 6 and 7 focus on the analysis and discussion of results regarding classification performance and imputation quality, respectively. Finally, Section 8 elaborates on the main findings of this work and presents some directions for future research.

## 2. Related work on k-nearest neighbours and distance functions

In this section, we discuss some related work on the use of distance functions coupled with k-Nearest Neighbours algorithm for data imputation (Section 2.1) and data classification (Section 2.2). To provide a panorama regarding the study of distance functions coupled with the kNN algorithm, we present an overview of these two major areas where kNN is deeply investigated. Nevertheless, we focus mostly on related work concerning data imputation, as is the main focus of this work. For a deeper analysis on kNN classification, the reader is referred to the recent work of Abu Alfeilat et al. (2019).

### 2.1. Related work on kNN imputation

In the field of kNN imputation, there are several different approaches found among related research.

Some related research considers only continuous or categorical features. Batista and Monard (2001) discuss kNN algorithm as an imputation method, considering a case study comprising one continuous dataset (the used distance function is not specified, although it is assumed the Euclidean default). Farhangfar et al. (2008) considers only discrete data (continuous features are left out of the analysis), and therefore a simple matching distance ($d_O$, Eq. (2)) is used. de Andrade Silva and Hruschka (2013) study the influence of different variants of kNNI on classification tasks, considering only continuous features and therefore using the Euclidean distance. Similarly, Tutz and Ramzan (2015) investigate improved weighting functions for kNNI, using variations of the Minkowski distance and considering solely continuous data. Eirola et al. (2013) specifically touch upon the issue of estimating distances with missing values, although considering only continuous data. Additionally, the framework works under the assumption of multivariate normal distributions. Beretta and Santaniello

**Table 1**
An overview of machine learning areas relying on distance metric learning. For each of the areas, it is explained how distance functions can be incorporated in the operations of each of the identified sub-areas, along with some examples of engineering applications and real-world problems where they can be studied.

| Machine learning area | Sub-area | Methodology | Engineering applications |
|---|---|---|---|
| Data classification | Neural networks | Distance functions are embedded in the operation of algorithms (e.g., radial basis functions networks, self-organising maps). | Fraud detection (West and Bhattacharya, 2016), software fault prediction (Malhotra, 2015), financial crisis prediction (Lin et al., 2011), engineering risk assessment (Hegde and Rokseth, 2020). |
| | Instance-based learning | Some are referred as nearest-neighbour techniques, memory-based reasoning methods, or case-based reasoning methods. These systems use a distance function to determine the similarity between a new pattern and the training data and use the nearest instance(s) to predict the target class. | Business failure prediction (Li et al., 2010), bankruptcy prediction (Cho et al., 2010), text mining (Gerhana et al., 2017), geoengineering (Sousa et al., 2017), cybersecurity (Elnaggar and Chakrabarty, 2018). |
| Data clustering | – | Clusters are found by identifying similar patterns. A suitable cluster solution comprises groups where its members have small distances among each other. | Financial distress (Lin et al., 2011), churn prediction (Mahajan et al., 2015), vehicle routing problems (Kiriş and Özcan, 2020), cybersecurity (Elnaggar and Chakrabarty, 2018). |
| Data preprocessing | Data resampling | Resampling approaches – undersampling and oversampling – use distance functions to analyse the neighbourhood of training examples and determine which patterns to clean or replicate. | Traffic accident's severity prediction (Zheng et al., 2019), residential energy modelling (Garbasevschi et al., 2021), identification of gang-related arson cases (Wang et al., 2021), solar flares forecasting (Ribeiro and Gradvohl, 2021), intrusion detection (Mbow et al., 2021). |
| | Instance selection | Prototype selection and instance selection methods use an instance-based classifier (commonly kNN) with a distance function, to find obtain a representative subset of the original training data. | Text categorisation (Barigou, 2018), smart data (Triguero et al., 2019), intrusion detection systems (Zhao et al., 2021). |
| | Dimensionality reduction | Distance functions are used as input for well-known dimensionality reduction algorithms, such as Multidimensional Scaling (MDS) or t-distributed Stochastic Neighbour Embedding (t-SNE). | Classification and visualisation of human genetic data (Li et al., 2017), Parkinson's disease (Oliveira et al., 2018), single-cell transcriptomics (Kobak and Berens, 2019), scientific visualisation, sports visualisation, forest fires analysis, virus disease analysis (Saeed et al., 2018). |
| | Data Imputation | Distance functions are used in kNN imputation as well as other imputation algorithms that operate with distances among patterns (e.g., NN, SOM, cluster-based imputation). They can also be as intermediate steps to improve other imputation approaches (e.g., via instance selection). Absent values of a given pattern are estimated using the available information of its closest neighbours. | Cancer survival prediction (Santos et al., 2015; García-Laencina et al., 2015), disease diagnosis and prognosis (Jerez et al., 2010), ubiquitous computing (Park et al., 2015), software applications and expert systems (Jäger et al., 2021), internet-of-things (IoT) smart systems (Okafor and Delaney, 2021), smart data (Triguero et al., 2019). |
| Data analysis and Meta-learning | Data complexity | Distance functions are in the base of several well-established complexity measures and instance hardness estimators (e.g., N1, N2, N3, T1 (Lorena et al., 2019), LSC (Leyva et al., 2014), CM (Anwar et al., 2014), R-value (Oh, 2011), kDN (Smith et al., 2014), among others). | Cancer detection (Sarbazi-Azad et al., 2020), Curriculum learning (Zhou et al., 2020; Nunes et al., 2021). |
| | Data typology | Depending on their local neighbourhoods, examples may be categorised into safe, borderline, rare or outlier examples (Napierala and Stefanowski, 2016). Using distinct distance functions may result in the different categorisation of examples (e.g., safe examples become borderline). | Anomaly detection (Kong et al., 2020), diabetes prediction (Nnamoko and Korkontzelos, 2020). |

(2016) study the impact of kNNI on the data structure and inferential and predictive statistics. Authors focus on problems comprising only continuous or binary features (where arithmetic operations over values may be performed), hence applying kNNI with variations of the Minkowski distance. Abnane et al. (2019) consider a set of variations of the Minkowski distance, dealing only with continuous features (categorical features were discarded from the analysis). Jadhav et al. (2019) also use only continuous features, and distance computation is performed using $d_N$ (Eq. (3)). Cheng et al. (2019) and Fouad et al. (2021) also consider only continuous features, applying the standard Euclidean distance.

Some works perform feature transformation in order to handle categorical features. Poulos and Valle (2018), Pereira et al. (2020), and Jäger et al. (2021) consider a one-hot encoding of categorical features before applying the Euclidean distance. Luengo et al. (2012) transform categorical (nominal) features to a list of numeric values, and then perform similarity computation using also the Euclidean distance. This approach may however be biased, since the transformation may distort the true similarity between patterns, as their numeric values not represent a real relationship or ordering between existing categories.

Some related research handles the imputation of heterogeneous data directly, either by resorting to heterogeneous distance functions or through the combination of distance functions adapted to each type of feature. The former is most often the case of application domains, where data is heterogeneous and may further incorporate missing data. To this regard, Jerez et al. (2010), Santos et al. (2015), and García-Laencina et al. (2015) couple kNNI with HEOM to handle real-world healthcare domains comprising continuous, categorical and missing values. Zhang (2011) also highlights the importance of choosing different distance measures for features of different types: some possibilities of distance functions are discussed for each type of attribute, and one is chosen for each type, without comparing other alternatives. Also,

the distance between patterns is only determined over observed data, i.e., missing values are not considered in distance computation. Bertsimas et al. (2017) consider a combination of the Euclidean distance with the $d_O$ (Eq. (2)) when handling heterogeneous data. Woźnica and Biecek (2020) couple $d_O$ (Eq. (2)) with $d_N$ (Eq. (3)) for categorical and continuous features, respectively.

Related research also resorts to Grey Relational Analysis (GRA) (Huang and Lee, 2004) as an alternative to Euclidean distance measure to continuous features (Huang et al., 2017), where some adaptations are considered for categorical features, so that the developed approaches can impute missing data in heterogeneous datasets (Zhang, 2012; Pan et al., 2015). However, these approaches are not compared with other heterogeneous distances, and also do not incorporate any strategies to consider missing data during distance computation. In a more recent work, Choudhury and Kosorok (2020) further modify GRA to handle missing values in similarity computation by assigning a minimal similarity value if either of the input values is missing. Nevertheless, a central issue with GRA, affecting all of the above works is that it requires the definition of a distinguishing coefficient $\rho \in [0, 1]$, for which no convincing method has been suggested so far (it is assigned to 0.5 by default) (Pan et al., 2015).

Finally, some related research seems to disregard the nature of data when studying kNNI on heterogeneous datasets. These either fail to characterise the used distance function (Batista and Monard, 2002, 2003), or refer only to the Euclidean function while no feature transformation techniques are discussed (Luengo et al., 2010; Huang et al., 2016; Tsai and Chang, 2016).

To summarise the contributions regarding kNN imputation over the past years, Table 2 provides an overview of related research. For each research work, the collected information comprises the objective of the study ("Behaviour", "Benchmark", "Application" or "Variant"), the details concerning the kNNI approach ($k$ value, considered distance measures and whether they internally handle the computation of missing values), the experimental design (number of datasets – continuous, categorical, and heterogeneous –, missing mechanisms – MCAR, MAR, and MNAR – and missing rates), and the considered downstream task (classification performance or imputation performance/quality). Furthermore, we highlight some important considerations regarding each related work, namely in what concerns the intrinsic characteristics of kNNI implementation or limitations of the experimental setup.

Note that, as mentioned in the Introduction, some variants/frameworks for kNNI improvement have been proposed over the years (e.g., SkNNI Kim et al. (2004), KMI Hruschka et al. (2004), IkNNI Brás and Menezes (2007), ICkNNI Van Hulse and Khoshgoftaar (2014), among others). However, these are precursor studies focused on specific modifications of adaptations to enhance kNNI, without a particular focus on distance functions, therefore applying the Euclidean distance by default. Although some more recent representative variants of kNNI are selected as related work and presented in Table 2, an in-depth discussion of kNN variants and adaptations is beyond the scope of this work (please refer to Huang et al. (2017) for a more comprehensive discussion).

From the assessment of Table 2, several observations should be highlighted:

- Euclidean distance is by far the most widely used distance function across all related research. However, in "Application" studies, where missing values often occur naturally in data, and domains are most frequently heterogeneous, the HEOM distance function is normally the go-to approach.
- Most related research focuses on performing "Benchmark" or "Variant" studies. These either involve the comparison of a set of data imputation techniques, or the comparison of a set of kNNI variants and frameworks, in order to determine the top performing approaches. Nevertheless, they often disregard the nature of data and the choice of appropriate distance functions: whereas finding an optimal value of $k$ is commonly a concern, the chosen distance function generally follows the default applied by software implementations.
- Several works require that the donor neighbours contain observed information in all features, or discard features with missing values when computing distances. Out of 29 research works (excluding our related research), only 5 (17%) are able to handle missing values internally during distance computation. However, the computation strategy is unanimous: if either of the input values is missing in a given feature $j$, the distance between patterns in that feature is 1 (maximal distance).
- The great majority of works evaluates data imputation by determining the improvement over the classification task (13/29), whereas only 9 works evaluate both tasks (imputation and classification), and 7 evaluate only the quality of imputation. MCAR is also the most frequently studied missing mechanism (considered in 18 works), followed by MAR (14) and MNAR (7).
- Some works either consider only continuous or categorical features, or perform feature transformation. The most frequent transformation is to perform one-hot encoding for categorical features. Other considered transformations are associated to a higher bias in distance computation: for instance, if nominal values are transformed to a list of numeric values and handled as continuous (Luengo et al., 2012), or if the distance between numeric data is defined by simple matching (Farhangfar et al., 2008).
- Whereas the information regarding the used value of $k$ is available in nearly all related research, the used distance function or feature transformation is often not disclosed, even when studies consider heterogeneous datasets.
- The largest benchmark of datasets is collected by Bertsimas et al. (2017) (84: 54/12/18) and Jäger et al. (2021) (69: 14/5/50). Nevertheless, datasets are not analysed individually according to their nature.

In contrast to related studies, both this work and our previous research (Santos et al., 2020b,a) introduces the following differences:

- They comprehend the most comprehensive collection and investigation of heterogeneous distance functions, namely HEOM, HEOM-R HVDM, HVDM-R, HVDM-S, MDE, and SIMDIST.
- All the distance functions used in this work are able to simultaneously handle continuous, categorical, and missing data. Accordingly, no feature transformation is required, all patterns with missing data are available to be donors (it is only required that they have observed information in the feature to impute), and the uncertainty of missing data can be accounted for.
- Beyond allowing distance computation with missing data, our studied distance functions further distinguish scenarios where only on input value is missing from situations where both are missing. The used strategies to handle each scenario may additionally depend on the type of feature at hand (continuous or categorical).

Finally, this current research comprises the largest benchmark of collected datasets among previous research. It considers 150 datasets from open source repositories, with an equal distribution of datasets by nature (50 continuous, 50 categorical, and 50 heterogeneous datasets) to allow a proper generalisation of results for each individual group of datasets. Additionally, and contrary to our previous research as well, this work focuses mostly on behaviour, rather than comparing and discussing results across distinct scenarios. It is highly motivated by the preliminary results obtained in Santos et al. (2020b) and Santos et al. (2020a), although it aims to provide thorough insights regarding the underlying operations of heterogeneous distance functions. We evaluate results both regarding classification performance and imputation quality, whereas related work is often focused solely on one perspective, mostly on the effect of kNNI on classification performance, as was done in our past research as well.

**Table 2**
Summary of existing literature on kNN imputation. For each related work are identified the objectives of the study, the parameters of the imputation approach, details regarding the experimental setup and the downstream task to be evaluated.

| Study | | kNN imputation approach | | | | Experimental data and simulation | | | Evaluation | | Considerations |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference | Objective[a] | k | Variants or[b] Frameworks | Distance measures | Considers[c] MVs | # Datasets[d] (Cont, Cat, H) | MCAR/MAR /MNAR | MRs[e] | Class.[f] Perf. | Imp.[g] Perf. | |
| Batista and Monard (2001) | Behaviour | 3 | N.A. | Unk. | • | 1 (1/0/0) | ✓/•/• | 10:10:50 | ✓ | • | Although not specifically stated, distance function is assumed Euclidean, as is the default in the traditional kNNI formulation. |
| Batista and Monard (2002) | Behaviour | 1, 3, 5, 10, 20, 30, 50, 100 | N.A. | Unk. | • | 3 (2/0/1) | ✓/•/• | 10:10:60 | ✓ | • | Although not specifically stated, distance function is assumed Euclidean, as is the default in the traditional kNNI formulation. Not clear how distance computation was formulated for nominal features. |
| Batista and Monard (2003) | Benchmark | 1, 3, 5, 10, 20, 30, 50, 100 | N.A. | Unk. | • | 4 (3/0/1) | ✓/•/• | 10:10:60 | ✓ | • | Although not specifically stated, distance function is assumed Euclidean, as is the default in the traditional kNNI formulation. Not clear how distance computation was formulated for nominal features. |
| Farhangfar et al. (2008) | Benchmark | 1 | N.A. | $d_O$ (Eq. (2)) | ✓ | 15 (0/13/2) | ✓/•/• | 5, 10:10:50 | ✓ | • | Considers only discrete data (i.e., discrete numerical and categorical data). Assumes $d_j = 0$ if both patterns have the same numerical or nominal values, otherwise $d_j = 1$. If either of the input values is missing, it also returns $d_j = 1$. |
| Luengo et al. (2010) | Benchmark | 10 | N.A. | Euclidean | • | 22 (9/3/10) | ✓/✓/• | MAR: Natural MCAR: 10% | ✓ | • | It is not clear how distance computation was formulated for heterogeneous datasets (e.g., nominal features). |
| Jerez et al. (2010) | Application | NNI: 1 kNNI: k chosen from CV | N.A. | HEOM | ✓ | 1 (0/0/1) | •/✓/• | Natural | ✓ | • | If either of the input values is missing, $d_j = 1$. |
| Zhang (2011) | Variant | Unk. | ✓ | Minkowski Simple Matching Jaccard, Matches Information-theoretic | • | 9 (6/0/3) | •/✓/• | 5, 10, 20, 40 | ✓ | ✓ | Distance function is a combination of several functions for each feature type. If either of the input values is missing in a given feature, that feature is ignored in distance computation. |
| Zhang (2012) | Variant | k set according to experiments | ✓ | Euclidean GRA[h] | • | 6 (2/2/2) | •/✓/• | 10, 20, 40 | ✓ | ✓ | If both input values have the same values for a categorical attribute, $GRA_j = 1$ (maximal similarity). Otherwise, $GRA_j = 0$ (minimal similarity). GRA implies the definition of a distinguishing coefficient, for which no convincing method has been suggested so far. |
| Luengo et al. (2012) | Benchmark | 10 | N.A. | Euclidean | • | 21 (3/7/11) | •/✓/• | Natural | ✓ | ✓ | Nominal values are considered as a list of integer values, starting from 1 to the number of different categories. |

(*continued on next page*)

**Table 2** (*continued*).

| Study | | kNN imputation approach | | | | Experimental data and simulation | | | Evaluation | | Considerations |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference | Objective[a] | k | Variants or[b] Frameworks | Distance measures | Considers[c] MVs | # Datasets[d] (Cont, Cat, H) | MCAR/MAR /MNAR | MRs[e] | Class.[f] Perf. | Imp.[g] Perf. | |
| de Andrade Silva and Hruschka (2013) | Benchmark | 10 | ✓ | Euclidean | • | 4 (4/0/0) | ✓/✓/• | 10, 30, 50, 70 | ✓ | ✓ | Only continuous data is considered in the experiments. |
| Eirola et al. (2013) | Behaviour | N.A. | N.A. | Statistical techniques are applied to find an expression for the expectation of the squared Euclidean distance between samples in a dataset with missing values. | | 9 (9/0/0) | Unk. Statistical techniques assume MCAR or MAR. | 5, 15, 30, 60 | • | ✓ | The study focuses on distance estimation for numerical data with missing values. The theoretical framework operates under the assumption of a multivariate normal distribution, although the algorithm has shown to be robust to violations of the assumptions regarding data distribution. |
| Tutz and Ramzan (2015) | Variant | k set by CV | ✓ | Euclidean Manhattan | • | 4 (2 Cont/2 Unk.) | ✓/•/• | 5 | • | ✓ | The computation of distances does not use all the components of the instances but only those for which observations in both instances are available. |
| Santos et al. (2015) | Application | 1 | N.A. | HEOM | ✓ | 1 (0/0/1) | Unk. | Natural | ✓ | • | If either of the input values is missing, $d_j = 1$. |
| García-Laencina et al. (2015) | Application | 1 to 40 | N.A. | HEOM | ✓ | 1 (0/0/1) | •/✓/• | Natural | ✓ | • | If either of the input values is missing, $d_j = 1$. |
| Pan et al. (2015) | Variant | 1 to 20 | ✓ | Euclidean GRA | • | 5 (2/2/1) | ✓/✓/✓ | 5, 10, 20 | ✓ | ✓ | If both input values have the same values for a categorical attribute, $GRA_j = 1$ (maximal similarity). Otherwise, $GRA_j = 0$ (minimal similarity). GRA implies the definition of a distinguishing coefficient, for which no convincing method has been suggested so far. |
| Beretta and Santaniello (2016) | Variant | 2, 3, 10 | ✓ | Minkowski Euclidean Manhattan | • | 1 (1/0/0) | ✓/•/• | 15 | • | ✓ | Experiments focus mostly on simulated continuous data and only with 1 real-world continuous dataset is considered. Only complete cases with no missing data are available as donors. |
| Huang et al. (2016) | Variant | Unk. | ✓ | Euclidean | • | 8 (4/1/3) | ✓/•/• | 5:5:50 | ✓ | • | Only the patterns with complete information in all attributes will serve as donors. The features that have missing values in the pattern to impute are ignored in distance computation. |
| Tsai and Chang (2016) | Variant | 10 | ✓ | Euclidean | • | 29 (11/9/9) | ✓/•/• | 10:10:50 | ✓ | • | Only the patterns with complete information in all attributes will serve as donors. The features that have missing values in the pattern to impute are ignored in distance computation. |
| Huang et al. (2017) | Variants | 1 to $\sqrt{N}$ in odd numbers | ✓ | Euclidean Manhattan GRA | • | 8 (8/0/0) | ✓/✓/✓ | 2.5, 5, 10, 20 | ✓ | ✓ | Focuses specifically on improvements for estimating continuous features. |

(*continued on next page*)

**Table 2** (*continued*).

| Study | | kNN imputation approach | | | | Experimental data and simulation | | | Evaluation | | Considerations |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference | Objective[a] | k | Variants or[b] Frameworks | Distance measures | Considers[c] MVs | # Datasets[d] (Cont, Cat, H) | MCAR/MAR /MNAR | MRs[e] | Class.[f] Perf. | Imp.[g] Perf. | |
| Bertsimas et al. (2017) | Variants | 1 to 100 | ✓ | Euclidean Euclidean + $d_O$ | • | 84 (54/12/18) | ✓/•/✓ | 10:10:50 | ✓ | ✓ | It is not clear how nominal features are handled in kNNI variants that use only the Euclidean distance. |
| Poulos and Valle (2018) | Benchmark | 3, 5 (`source code`) | N.A. | Euclidean (`source code`) | • | 2 (0/1/1) | ✓/•/• | 10:10:40 | ✓ | • | Missing values are introduced only on categorical features. Categorical features are transformed using one-hot encoding. |
| Abnane et al. (2019) | Application | 1 to 5 | ✓ | Minkowski Euclidean Manhattan Chebychev | • | 6 (6/0/0) | ✓/✓/✓ | 10:10:90 | • | ✓ | The study deals only with continuous features. Therefore, datasets with categorical features were discarded. |
| Jadhav et al. (2019) | Benchmark | 5 (`VIM package`) | N.A. | $d_N$ (Eq. (3)) (`VIM package`) | • | 5 (5/0/0) | Unk. | 10:10:50 | • | ✓ | Only continuous data is considered. kNNI is done by using the `VIM` package in R, where the distance between continuous features is calculated as $d_N$ (Eq. (3)). |
| Cheng et al. (2019) | Variant | 3, 5, 7, 9 | ✓ | Euclidean | • | 8 (8/0/0) | ✓/✓/• | 5:5:25 | ✓ | • | The used datasets consider only continuous features. |
| Pereira et al. (2020) | Benchmark | 5 | N.A. | Euclidean | • | 10 (5/0/5) | •/•/✓ | 10:10:40 | • | ✓ | Categorical features are transformed using one-hot encoding. |
| Woźnica and Biecek (2020) | Benchmark | NNI: 1 kNNI: 5 (`VIM package`) | N.A. | $d_N$ + $d_O$ (`VIM package`) | • | 13 (0/1/12) | Unk. | Natural | ✓ | • | kNNI is done by using the `VIM` package in R, where the distance between continuous features is calculated as $d_N$ (Eq. (3)) and the distance between categorical features as $d_O$ (Eq. (2)). |
| Choudhury and Kosorok (2020) | Variant | k set by CV | ✓ | Euclidean GRA | Euclidean (Unk.) GRA (✓) | 3 (1/1/1) | •/✓/• | 5, 10, 20 | ✓ | ✓ | It is not clear how nominal features are handled in kNNI variants that use only the Euclidean distance. In GRA, if either of the input values is missing, $GRA_j = 0$. |
| Jäger et al. (2021) | Benchmark | 1, 3, 5 | N.A. | Euclidean (`scikit-learn`) | • | 69 (14/5/50) | ✓/✓/✓ | 1, 10, 30, 50 | ✓ | ✓ | Considers one-hot encoding for categorical features. |
| Fouad et al. (2021) | Benchmark | 2 to $N$ | ✓ | Euclidean | • | 15 (15/0/0) | ✓/✓/✓ | 1, 5,10, 20 | • | ✓ | The proposed imputation techniques can only handle continuous features, not categorical features. |
| **Our related research:** | | | | | | | | | | | |
| Santos et al. (2020a) | Application | 1, 3, 5, 7 | N.A. | HEOM, HEOM-R HVDM, HVDM-R HVDM-S, MDE SIMDIST | ✓ | 31 (0/0/31) | ✓/•/• | 5, 10, 20, 30 | ✓ | • | All distances handle continuous and categorical features, as well as missing data. Some distinguish situations where only one value is missing or both are missing. |
| Santos et al. (2020b) | Benchmark | 1 | N.A. | HEOM, HEOM-R HVDM, HVDM-R HVDM-S, MDE SIMDIST | ✓ | 61 (37/1/23) | ✓/•/• | 5, 10, 20, 30 | ✓ | • | All distances handle continuous and categorical features, as well as missing data. Some distinguish situations where only one value is missing or both are missing. |

(*continued on next page*)

**Table 2** (*continued*).

| Study | | kNN imputation approach | | | | Experimental data and simulation | | | Evaluation | | Considerations |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reference | Objective[a] | k | Variants or[b] Frameworks | Distance measures | Considers[c] MVs | # Datasets[d] (Cont, Cat, H) | MCAR/MAR /MNAR | MRs[e] | Class.[f] Perf. | Imp.[g] Perf. | |
| **This work**: | Behaviour | 1, 3, 5, 7 | N.A. | HEOM, HEOM-R HVDM, HVDM-R HVDM-S, MDE SIMDIST | ✓ | 150 (50/50/50) | ✓/•/• | 5, 10, 20, 30 | ✓ | ✓ | All distances handle continuous and categorical features, as well as missing data. Some distinguish situations where only one value is missing or both are missing. |

[a]**Objective of the study**: Study of kNNI as an imputation model (“Behaviour”), Proposal or study of new approaches (modifications, adaptations, frameworks, optimisation techniques) to improve kNNI (“Variants”), Application of kNNI to real-world domain (“Application”), Uses kNNI in a benchmark study of data imputation approaches (“Benchmark”).
[b]**Variants or Frameworks**: The study compares some well-established kNNI variants of frameworks (e.g., adaptations of the original kNNI formulation, weighting schemes).
[c]**Considers MVs**: The used distance function incorporates the computation of missing values.
[d]**# Datasets**: Number of total datasets (continuous/categorical/heterogeneous).
[e]**MRs**: Missing rates used in the experiments. A code of “10:10:50”, means that MRs are considered from 10% to 50%, in a step of 10, i.e., {10, 20, 30, 40, 50}%. “Natural” means that missing values occur naturally in the dataset (not artificially generated).
[f]**Class. Perf.**: Imputation results are evaluated according to the benefits for classification performance (e.g., Accuracy, AUC, F1).
[g]**Imp. Perf.**: Imputation results are evaluated according to the quality of reconstructed values, i.e., imputation performance (e.g., MSE, RMSE, MAE).
[h]**GRA:** Grey Relational Analysis, which can be used to measure distance, by applying $D = 1 - GRA$.

### 2.2. Related work on kNN classification

In the field of data classification, there is a greater interest in the search of optimal distance functions, with a larger number of papers experimenting with several possible choices. This is perhaps due to the fact that in classification tasks, kNN is directly used to the endgame objective, i.e., predicting the final class labels, whereas in data imputation, it is used as an intermediate process, since the classification task may be addressed (and improved) by any other learning paradigm. Batista and Silva (2009) present a benchmark study in kNN classification considering the value of $k$, different heterogeneous distance functions (HEOM, HVDM and HMOM which uses the Manhattan distance rather than Euclidean as in HEOM), and different weighting functions. Despite some datasets comprised missing values, there were no experiments with increasing amount of missing data. No significant differences were found among the three studied distance functions, although this may be due to the uneven number of datasets with different types (16 continuous, 4 categorical and 10 heterogeneous datasets). Hu et al. (2016) discuss whether the distance function may affect kNN performance over different medical datasets. Authors use the Euclidean, Minkowski, Cosine, and Chi Square for both continuous, categorial and heterogeneous data, neglecting the nature of features. Ali et al. (2019) investigate the performance of kNN on heterogeneous data, although described as a mixture of continuous and binary features (no nominal features are considered). Different distance measures are defined and compared, based on the combination of well-known distance functions for continuous and binary data. Prasatha et al. (2019) present a comprehensive review on kNN classification attending to distinct distance functions and include a through experimental study focused on defining the best distance measures to be used with kNN classifier. However, experiments consider only continuous and binary features (no heterogeneous data or functions are discussed) and no missing values are allowed in the training data. Recent kNN classification approaches include (Gou et al., 2019a,b; Ertuğrul, 2019; Wang and Yang, 2020), although recurring to variations of the Minkowski distance, most often the Euclidean distance.

Overall, as depicted in the Introduction, related work on kNN imputation or classification is more frequently focused on exploring new variants or weighting schemes to devise optimal frameworks for kNN behaviour. Although searching for suitable distance functions is also a step towards the definition of optimal solutions, this remains an overlooked topic, especially in complex domains comprising heterogeneous data – continuous and categorical (binary and nominal) data – as well as missing values. To this regard, this study offers a new perspective on the subject since (i) several heterogeneous distance functions are compared both in terms of impact in classification performance as well as imputation performance, (ii) the analysis is segregated by dataset type (continuous, categorical or heterogeneous) so that each component of the distance functions can be compared, (iii) the chosen functions directly incorporate missing values in distance computation.

## 3. Missing data background

According to Rubin (1976), there are three underlying mechanisms under which data can be missing: Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR). The missing mechanisms describe the relation between the probability distribution of the missing values and the observed and unobserved information in data, via conditional probabilities, as we explain in what follows. Consider $\mathbf{X}$ as a variable representing a given dataset, which can be divided into $\mathbf{X}_{obs}$ and $\mathbf{X}_{miss}$, i.e., the observed and missing values in $\mathbf{X}$, respectively. Consider also a missing data indicator $\mathbf{M}$ as a 0/1 matrix determining the locations of the missing values in $\mathbf{X}$ (where “1” denotes a missing value). Rubin’s missing data theory characterises the missing mechanisms by defining to what extent the probability distribution of $\mathbf{M}$ may depend on $\mathbf{X}$:

- Missing Completely At Random (MCAR): MCAR is formulated as $p(\mathbf{M} = 1 \mid \mathbf{X}) = p(\mathbf{M} = 1)$, demonstrating that $\mathbf{M}$ is completely unrelated to the input data $\mathbf{X}$, either $\mathbf{X}_{obs}$ and $\mathbf{X}_{miss}$. Accordingly, the probability of missing values is completely random;
- Missing At Random (MAR): MAR mechanism characterises a situation where $\mathbf{M}$ depends on the observed information in the dataset, $\mathbf{X}_{obs}$, but not on $\mathbf{X}_{miss}$, i.e., $p(\mathbf{M} = 1 \mid \mathbf{X}) = p(\mathbf{M} = 1 \mid \mathbf{X}_{obs})$. Hence, the probability of missing values depends solely on available, observed information in data;
- Missing Not At Random (MNAR): In MNAR, $\mathbf{M}$ depends both on $\mathbf{X}_{obs}$ and $\mathbf{X}_{miss}$, meaning that the probability of missing values may be related to the both observed and unobserved information in data. The missing data model is therefore described in its full extension, $p(\mathbf{M} = 1 \mid \mathbf{X}) = p(\mathbf{M} = 1 \mid \mathbf{X}_{obs}, \mathbf{X}_{miss})$.

For a more detailed formulation of the missing mechanisms and illustrative examples, the reader is referred to Santos et al. (2019). In this work, we will focus on MCAR mechanism for synthetic generation of missing data (more details will be given in Section 5). MCAR is the most frequently studied missing mechanism among imputation works, especially when coupled with kNN imputation (Batista and Monard, 2003; Farhangfar et al., 2008; Tutz and Ramzan, 2015; Huang et al., 2016; Tsai and Chang, 2016; Lin and Tsai, 2020). Additionally,

we chose MCAR for consistency and control across different types of datasets (continuous, categorical and heterogeneous), namely to avoid the limitations found for multivariate MAR and MNAR missing data generation regarding categorical data, as thoroughly described in Santos et al. (2019). Finally, focusing solely on MCAR mechanism avoids the need of additional experiments to choose suitable determining features for MAR and MNAR and perform distinct runs depending on the chosen set of features. Since the evaluation of distance functions under several missing rates and stochastic runs is inherently computationally expensive, and the focus of the work relies on the evaluation of their behaviour rather than finding the best possible solution under defined conditions (e.g., missing mechanisms and rates), focusing on MCAR simplifies the experimental design without compromising the study objectives. Nevertheless, examining MAR and MNAR assumptions are possible directions for future research.

## 4. Heterogeneous distance functions for missing data

In this work, distance computation relies on the evaluation of seven distinct distance functions: HEOM and HVMD (Wilson and Martinez, 1997), their redefinitions (HEOM-R and HVDM-R (Juhola and Laurikkala, 2007), and HVDM-S (Santos et al., 2020b)), SIMDIST (Belanche Muñoz and Hernández González, 2012), and MDE (AbdAllah and Shimshoni, 2016). HEOM and HVDM are commonly used in the context of heterogeneous data, across different domains (Jerez et al., 2010; Santos et al., 2015; García-Laencina et al., 2015; Napierala and Stefanowski, 2016; Borowska and Stepaniuk, 2016). HEOM-R, HVDM-R and HVDM-S were included as alternatives to their predecessors due to their considerations regarding the treatment of missing values (Juhola and Laurikkala, 2007). SIMDIST and MDE, although not originally designed for data imputation, have provided interesting results in preliminary research (Santos et al., 2020b,a). The distance functions described in this section are implemented in a MATLAB library publicly available on GitHub. [2] Furthermore, distances were chosen based on three main criteria. First, they were required to handle different natures of data simultaneously (i.e., heterogeneous data) either in their original formulation or with minimal modifications (which is the case of MDE, previously extended to handle nominal data in Santos et al. (2020b)). Secondly, the set of chosen distance functions was required to incorporate diverse strategies to evaluate different types of features, as well as missing data. Naturally, HEOM-R, HVDM-R, and HVDM-S, as redefinitions of HEOM and HVDM, use the same respective strategy to handle continuous and categorical features, though not missing values. Otherwise, chosen distance functions follow different mechanisms for distance computation and treatment of missing values. Some further distinguish situations where on or both values are missing and/or estimate the distance differently, depending on the feature type. Finally, distance functions should be easy to compute. A well-known drawback of kNN-related approaches is the need to evaluate the similarity among all patterns in data, which may be computationally expensive and time-consuming for larger datasets (Batista and Monard, 2002). Although some strategies have been explored to surpass such limitations (Deng et al., 2016; Maillo et al., 2017), this issue falls outside of the scope of this work.

Herein, we start by briefly providing some essential notation on distance computation, whereas the mathematical formulation of each considered distance function is discussed along this section. Given a dataset $\mathbf{X}$, represented by a $n \times p$ matrix (where $n$ is the number of patterns and $p$ is the number of features), distance functions measure the distance between two patterns $\mathbf{x}_A$ and $\mathbf{x}_B$ through a sum of their individual distances in each $j$th feature ($j = 1, \ldots, p$), $d_j(x_{Aj}, x_{Bj})$, as $D(\mathbf{x}_A, \mathbf{x}_B) = \sqrt{\sum_{j=1}^{p} d_j(x_{Aj}, x_{Bj})^2}$. However, they differ on the computation of individual $d_j$ distances and treatment of missing values, as explained in what follows.

[2] https://github.com/miriamspsantos/heterogeneous-distance-functions.

### 4.1. Heterogeneous Euclidean-overlap metric

The definition of $d_j(x_{Aj}, x_{Bj})$ for Heterogeneous Euclidean-Overlap Metric (HEOM) distance (Wilson and Martinez, 1997) depends on the type of feature $j$ (Eq. (1)). For categorical/nominal features, $d_j$ is defined as an overlap metric, $d_O$ (Eq. (2)); while for continuous features, the normalised euclidean distance, $d_N$ (Eq. (3)), is used instead ($x_j$ represents all values of the $j$th feature). However, $d_O$ and $d_N$ are only computed if both input values, $x_{Aj}$ and $x_{Bj}$ are available; otherwise, if either of them is missing, $d_j(x_{Aj}, x_{Bj})$ is defined as 1. As shown in Eq. (1), the individual $d_j$ distances vary between 0 and 1, and therefore a missing value in the $j$th feature is traduced as a maximum $d_j$ distance between $\mathbf{x}_A$ and $\mathbf{x}_B$.

$$d_j(x_{Aj}, x_{Bj}) = \begin{cases} 1, & \text{if } j \text{ is missing in } x_{Aj} \text{ or } x_{Bj}, \\ d_O(x_{Aj}, x_{Bj}), & \text{if } j \text{ is a categorical feature}, \\ d_N(x_{Aj}, x_{Bj}), & \text{if } j \text{ is a continuous feature} \end{cases} \tag{1}$$

$$d_O(x_{Aj}, x_{Bj}) = \begin{cases} 0, & \text{if } x_{Aj} = x_{Bj} \\ 1, & otherwise \end{cases} \tag{2}$$

$$d_N(x_{Aj}, x_{Bj}) = \frac{\left| x_{Aj} - x_{Bj} \right|}{max(x_j) - min(x_j)} \tag{3}$$

### 4.2. Heterogeneous value difference metric

The Heterogeneous Value Difference Metric (HVDM) (Wilson and Martinez, 1997), defines the distance between $\mathbf{x}_A$ and $\mathbf{x}_B$ as described by Eq. (4). Again, if both values $x_{Aj}$ and $x_{Bj}$ are observed, the type of $j$ determines the computation of individual $d_j$ distances: $d_{vdm}$ is used for categorical/nominal features (Eq. (5)) while $d_{diff}$ is used for continuous features (Eq. (6)).

$$d_j(x_{Aj}, x_{Bj}) = \begin{cases} 1, & \text{if } j \text{ is missing in } x_{Aj} \text{ or } x_{Bj}, \\ d_{vdm}(x_{Aj}, x_{Bj}), & \text{if } j \text{ is a categorical feature}, \\ d_{diff}(x_{Aj}, x_{Bj}), & \text{if } j \text{ is a continuous feature} \end{cases} \tag{4}$$

The computation of $d_{vdm}$, as shown in Eq. (5), requires information on the class targets of each pattern $\mathbf{x}_i$ ($i = 1, \ldots, n$), herein referred to as $c_i$. Thus, $d_{vdm}$ is computed as a sum over all classes, where $C$ is the number of classes in the problem domain — as we are focusing on binary problems, $C = 2$, and therefore $c_i \in \{1, 2\}$. $N_{x_{Aj},c}$ is the number of patterns in $\mathbf{X}$ that have value $x_{Aj}$ in feature $j$ and class target $c$, while $N_{x_{Aj}}$ is the number of patterns in $\mathbf{X}$ that have value $x_{Aj}$ in feature $j$ (the same for $x_{Bj}$).

$$d_{vdm}(x_{Aj}, x_{Bj}) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{x_{Aj},c}}{N_{x_{Aj}}} - \frac{N_{x_{Bj},c}}{N_{x_{Bj}}} \right|^2} \tag{5}$$

Similarly to HEOM, the continuous features are scaled by $d_{diff}$, considering 4 standard deviations ($\sigma$) of $x_j$.

$$d_{diff}(x_{Aj}, x_{Bj}) = \frac{\left| x_{Aj} - x_{Bj} \right|}{4\sigma_{x_j}} \tag{6}$$

### 4.3. Redefinitions of HEOM and HVDM

Redefinitions of HEOM and HVDM (Juhola and Laurikkala, 2007) propose that missing values are considered "special values", and that the distance between two missing values is assumed to be 0 (missing values are considered equal values). Accordingly, HEOM-R and HVDM-R are different from their original formulations in what concerns the treatment of missing values (Eq. (7)):

$$d_j(x_{Aj}, x_{Bj}) = \begin{cases} 1, & \text{if } j \text{ is missing only in } x_{Aj} \text{ or } x_{Bj}, \\ 0, & \text{if } j \text{ is missing in both } x_{Aj} \text{ and } x_{Bj} \end{cases} \tag{7}$$

In addition, we propose another possible redefinition for HVDM: if missing values are considered an "special" nominal category, $d_{vdm}$ may

be applied in the case that only $x_{Aj}$ or only $x_{Bj}$ are missing, and $j$ is categorical/nominal, referred to as HVDM-S (Eq. (8)).

$$d_j(x_{Aj}, x_{Bj}) = \begin{cases} 0, & \text{if } x_{Aj} \text{ and } x_{Bj} \text{ are both missing,} \\ 1, & \text{if } x_{Aj} \text{ or } x_{Bj} \text{ are missing and } j \text{ is continuous,} \\ d_{vdm}(x_{Aj}, x_{Bj}), & \text{if } x_{Aj} \text{ or } x_{Bj} \text{ are missing and } j \text{ is categorical} \end{cases} \quad (8)$$

### 4.4. Similarity for heterogeneous data

SIMDIST defines a similarity measure, where $S_{ABj}$ represents the similarity between patterns $\mathbf{x}_A$ and $\mathbf{x}_B$ according to feature $j$.

$$S_{ABj} = \begin{cases} \frac{1}{2}, & \text{if either } x_{Aj} \text{ or } x_{Bj} \text{ are missing,} \\ z\left(\frac{s_{ABj}}{s_j}\right), & \text{if both } x_{Aj} \text{ and } x_{Bj} \text{ are known} \end{cases} \quad (9)$$

$s_{ABj}$ is an intermediate similarity distance between $x_{Aj}$ and $x_{Bj}$ and is determined according to the type of $j$ (either a categorical/nominal or continuous feature). In the above equation, $s_j$ represents the mean similarity among all patterns according to $j$ and $z$ is a normalisation function $z : (0, +\infty) \rightarrow (0, 1)$, described as $z(a) = \frac{a}{a+1}$ (Belanche Muñoz and Hernández González, 2012).

For categorical/nominal features, $s_{ABj}$ is defined by Eq. (10), where $P_{lj}$ in the fraction of patterns that takes value $x_{lj}$ for feature $j$. In practice, $P_{lk}$ is the fraction of examples that assume value $x_{Aj}$ or $x_{Bj}$ for $j$, since for this computation they are equal, as shown in Eq. (10).

$$s_{ABj} = \begin{cases} 0, & \text{if } x_{Aj} \neq x_{Bj}, \\ 1 - P_{lj}, & \text{if } x_{Aj} = x_{Bj} \end{cases} \quad (10)$$

For continuous features, $s_{ABj}$ is determined by Eq. (11), where $\max(x_j)$ and $\min(x_j)$ are the maximum and minimum values observed in $j$, respectively.

$$s_{ABj} = 1 - \frac{| x_{Aj} - x_{Bj} |}{max(x_j) - min(x_j)} \quad (11)$$

In Eq. (9), $S_{ABj}$ is assumed to be $\frac{1}{2}$ when $x_{Aj}$ or $x_{Bj}$ are missing which is the equivalent of replacing the missing similarity between $x_{Aj}$ or $x_{Bj}$ by the mean similarities of all patterns according to $j$. Replacing the missing similarity $s_{ABj}$ by the mean of all similarities in $j$, $s_j$, we would obtain $z(\frac{s_j}{s_j}) = \frac{1}{2}$. Naturally, this *similarity* function $S$ reveals how "alike" two values are whereas we are interested in obtaining a value of "how far apart" the values are. Therefore, it needs to be adjusted to reflect a *distance* between patterns, rather than a *similarity*. As $S_{ABj}$ is defined in the domain [0,1], the distance between $x_A$ and $x_B$ in $j$ is given by $d_j(x_{Aj}, x_{Bj}) = 1 - S_{ABj}$. Thus, the calculation of this distance, which will be referred to as SIMDIST, starts by determining the individual similarities $S_{ABj}$, which are then transformed to individual $d_j$ distances. Then, since the distance matrix among all examples is available for all features, the computation of $D(\mathbf{x}_A, \mathbf{x}_B)$ is the same as for the previous distances.

### 4.5. Mean Euclidean Distance

Mean Euclidean Distance ($MD_E$) (AbdAllah and Shimshoni, 2014, 2016) defines three possibilities for comparing two values of a given feature $j$:

1. Both values are known: When $x_{Aj}$ and $x_{Bj}$ are observed, their distance is defined as the standard euclidean distance:

$$MD_E(x_{Aj}, x_{Bj}) = (x_{Aj} - x_{Bj})^2 \quad (12)$$

2. One value is missing: When either $x_{Aj}$ or $x_{Bj}$ are missing, $MD_E$ is approximated as the mean distance of each value of $x_j$ to the observed value. Considering that $x_{Aj}$ is missing and $x_{Bj}$ is observed, $MD_E$ is defined by Eq. (13). To ease the interpretation of Eq. (13), we consider an auxiliary variable $x = x_j$. Thus, $\mu_x$ and $\sigma_x$ are equivalent to $\mu_{x_j}$ and $\sigma_{x_j}$, and refer to the mean and standard deviation of all the observed values of $x_j$.

$$\begin{aligned} MD_E(x_{Aj}, x_{Bj}) &= E\left((x - x_{Bj})^2\right) \\ &= \int p(x)(x - x_{Bj})^2 dx \\ &= (x_{Bj} - \mu_x)^2 + \sigma_x^2 \end{aligned} \quad (13)$$

3. Both values are missing: When both $x_{Aj}$ and $x_{Bj}$ are missing, the $MD_E$ is approximated as the mean distance between all values of $x_j$ (Eq. (14)). Similarly, we consider the auxiliary variables $x, y = x_j$.

$$\begin{aligned} MD_E(x_{Aj}, x_{Bj}) &= \iint p(x)p(y)(x - y)^2 dx dy \\ &= \left(E(x) - E(y)\right)^2 + \sigma_x^2 + \sigma_y^2 \\ &= 2\sigma_x^2 \end{aligned} \quad (14)$$

To allow a proper weighting of continuous features with different ranges, a min–max normalisation (Eq. (15)) is applied before the euclidean distance is computed. This normalisation scales all continuous features to the same range, avoiding that features with a larger range assume a higher weight in the distance computation.

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \quad (15)$$

However, Eqs. (12) to (14) define the $MD_E$ distance for continuous features. For heterogeneous datasets, these equations need to be extended for the categorical/nominal case. To extend $MD_E$ for categorical/nominal features, we shall consider the standard overlap distance, $d_O$ (Eq. (2)) and define a categorical version of $MD_E$, which we will refer to as $MD_O$.

1. Both values are known: In this case, $MD_O$ is the same as $d_O$.

$$MD_O(x_{Aj}, x_{Bj}) = \begin{cases} 0, & \text{if } x_{Aj} = x_{Bj} \\ 1, & otherwise \end{cases} \quad (16)$$

2. One value is missing: Supposing $x_{Aj}$ is missing and $x_{Bj}$ is observed, $MD_O$ is computed as the mean distance between all elements in $x_j$ and $x_{Bj}$. Again, we make use of $x = x_j$. Given the definition of $d_O$, the sum will only be non-zero when $x \neq x_{Bj}$, hence the simplification.

$$\begin{aligned} MD_O(x_{Aj}, x_{Bj}) &= \sum_x p(x)\ d_O(x, x_{Bj}) \\ &= \sum_{x \neq x_{Bj}} p(x) \\ &= 1 - p(x_{Bj}) \end{aligned} \quad (17)$$

3. Both values are missing: When both $x_{Aj}$ and $x_{Bj}$ are missing, $MD_O$ is determined as the mean distance between all elements in $x_j$. Similarly, we consider auxiliary variables $x, y = x_j$.

$$\begin{aligned} MD_O(x_{Aj}, x_{Bj}) &= \sum_x \sum_y p(x)p(y)\ d_O(x, y) \\ &= \sum_x \sum_{y \neq x} p(x)p(y) \\ &= 1 - \sum_x p^2(x) \end{aligned} \quad (18)$$

Finally, after the individual distances are computed, their aggregation is performed as for the remaining distances, $D(\mathbf{x}_A, \mathbf{x}_B)$, assuming $d_j(x_{Aj}, x_{Bj})$ as $MD_E(x_{Aj}, x_{Bj})$ or $MD_O(x_{Aj}, x_{Bj})$, depending on the feature type (continuous or categorical/nominal). Note that $MD_E(x_{Aj}, x_{Bj})$ already corresponds to $d_j(x_{Aj}, x_{Bj})^2$ (Eqs. (12) to (14)), therefore, only the $MD_O(x_{Aj}, x_{Bj})$ component should be squared when performing the aggregation.

## 5. Experimental setup

An overview of the considered experimental setup is presented in Fig. 2. We started by collecting several datasets from open-source repositories, Dua and Graff (2019), Kaggle (0000), Vanschoren et al. (2013) and Alcalá-Fdez et al. (2011). All datasets are originally complete (i.e., without missing data), so that both the missing mechanism and percentage are controlled parameters of our experiments. Furthermore, all datasets represent binary-classification problems, to simplify the classification stage of the experimental setup (since, as previously detailed, kNN imputation may present an added complexity in terms of memory and computational time). Thus, rather than the number of classes, we focus on the heterogeneity among datasets with respect to their sample sizes, number of features, application domains, and imbalance ratios (IR). More importantly, to fully understand to what extent each component of a function definition influences imputation and classification performance, we focus on dataset diversity in what concerns their type of features, thus collecting a total of 150 datasets where 50 are continuous, 50 categorical and 50 are heterogeneous.

For each dataset, a holdout partitioning was performed (Fig. 2) and missing data was generated in each training set. Then, to determine the impact of imputation on classification performance, both the training sets with missing values (BASELINE approach) and the imputed training sets (kNN imputation) were used to train Classification and Regression Trees (CART) models, and the classification performance was evaluated using Sensitivity, F-measure and G-mean (Santos et al., 2018). Additionally, the quality of imputation was also evaluated, by examining the differences between the original training sets (ground truth) and the imputed training sets (Fig. 2). Additional considerations regarding the proposed setup are as follows:

- **Data Partitioning:** Each dataset is partitioned following a stratified holdout method (80% of data for training and 20% for testing) (Farhangfar et al., 2008; Valdiviezo and Van Aelst, 2015), where each set respects the proportion of class labels (same IR for training and test sets). Additionally, 30 runs of holdout partition are performed for each dataset;
- **Missing Data Generation:** Missing values were generated at 4 different rates (5, 10, 20 and 30%) under a Missing Completely At Random (MCAR) mechanism. Additionally, we guarantee that the same missing rate was inserted in both classes according to the IR of the dataset, i.e., we guarantee that missing data is affecting both classes proportionally to their distribution. Finally, missing data is inserted only on training sets since the objective of this work is to analyse the effect of different distance functions on kNN as imputation method and the consequent impact on the classification model's learning ability (Batista and Monard, 2003).
- **Data Imputation:** kNN imputation considered 7 distance functions (described in Section 4), and 4 values for $k$ (1, 3, 5 and 7 nearest neighbours). Additionally, kNN uses a weighted average of the neighbours' feature values to impute continuous features, whereas categorical features are imputed with the most common value among the nearest neighbours, i.e., the mode ($Mo$). Considering an example pattern $\mathbf{x}_Z$ for which a value is missing on its feature $j$ and a set of its $k$ nearest neighbours $\mathbf{V}$, the estimated value of $x_{Zj}$, i.e., $\hat{y}_{Zj}$ is determined as:

$$\hat{y}_{Zj} = \begin{cases} \frac{\sum_{i=1}^{k} w_{Vi} x_{Vij}}{\sum_{i=1}^{k} w_{Vi}}, & \text{if } j \text{ is continuous,} \\ Mo(\mathbf{V}_j), & \text{if } j \text{ is categorical} \end{cases} \tag{19}$$

  The weights for continuous features are inversely proportional to the distance between pattern $\mathbf{x}_Z$ and its $i$th nearest neighbour, i.e., $w_{Vi} = \frac{1}{D(\mathbf{x}_Z, \mathbf{x}_{Vi})^2}$;
- **Classification:** CART models were chosen since they are relatively fast to construct and to provide classification results. Furthermore, these models are able to handle missing data directly through the use of surrogate splits (without discarding any patterns or observed values from the dataset), thus allowing to study the impact of imputation on classification performance by comparing models constructed from missing data with models constructed from imputed data (Twala, 2009; Valdiviezo and Van Aelst, 2015).
- **Evaluation:** The impact of distance functions on data imputation is discussed in terms of classification performance and imputation quality. Regarding classification performance, Sensitivity, F-measure and G-mean are presented due to robustness to the existing class imbalance of the collected datasets (Santos et al., 2018). For assessing imputation quality, Normalised Mean Absolute Error (NMAE) and the percentage of matches, Matches (%) were computed (Pereira et al., 2020).

Overall, the entire experimental setup involved the analysis of 150 datasets × 30 versions × 4 missing rates (BASELINE approach) + 2 × 50 datasets × 30 versions × 4 missing rates × 4 $k$ values × 7 distance functions (kNN imputation of categorical and heterogeneous datasets) + 50 datasets × 30 versions × 4 missing rates × 4 $k$ values × 6 distance functions (kNN imputation of continuous datasets) = **498,000 datasets**.

In the following sections, we focus on the analysis of the obtained experimental results, regarding two aspects: the impact on classification performance (Section 6) and the impact on imputation quality (Section 7).

Regarding classification performance (Section 6), we are interested in comparing the classification results obtained with CART models trained with different imputed training sets (on the same test set). As an example, consider two training sets imputed with and HEOM and MDE ($k = 1$, for instance), $\hat{\mathbf{X}}_{\mathbf{HEOM}}$ and $\hat{\mathbf{X}}_{\mathbf{MDE}}$. For each imputed training set, the same CART model (with the same initial conditions/parameters) is trained. After the training stage, there are two distinct CART models, that will be used predict new cases on the same test set. The top performing imputation approach (distance function) is the one that originates the CART model with the highest classification results. In such a way, we determine which distance function benefits the most the classification task, i.e., produces estimates for missing values that ease the classification task and improve the classification results. Within this analysis, we also consider CART models built with training sets with missing values (BASELINE approach), for which the comparison of classification results is the same as explained.

Regarding imputation quality (Section 7), we evaluate the imputation task directly by comparing the original training set values with the estimates produced by each distance function. Following the previous example, consider that $\mathbf{X_o}$ represents the original training set and $\mathbf{X_m}$ the training dataset with missing values. Then, we compare $\hat{\mathbf{X}}_{\mathbf{HEOM}}$ and $\hat{\mathbf{X}}_{\mathbf{MDE}}$ with $\mathbf{X_o}$ in the positions where $\mathbf{X_m}$ is missing and evaluate each distance function in what concerns the recovery/reconstruction of missing data. The best imputation approach (distance function) is the one that produces estimates (imputed values) that are closest to the original values.

## 6. Impact on classification performance

In this section we analyse the impact of distance functions on kNN imputation regarding classification performance. Distance functions are compared in terms of the classification performance achieved by CART models built on datasets imputed with different distances. In this case, we consider that the main objective is to solve a classification task, i.e., imputation methods are evaluated in what concerns their ability to produce more accurate and efficient classification models. The imputation task is considered an auxiliary task whose purpose is to obtain imputed values that help to model the classification task.

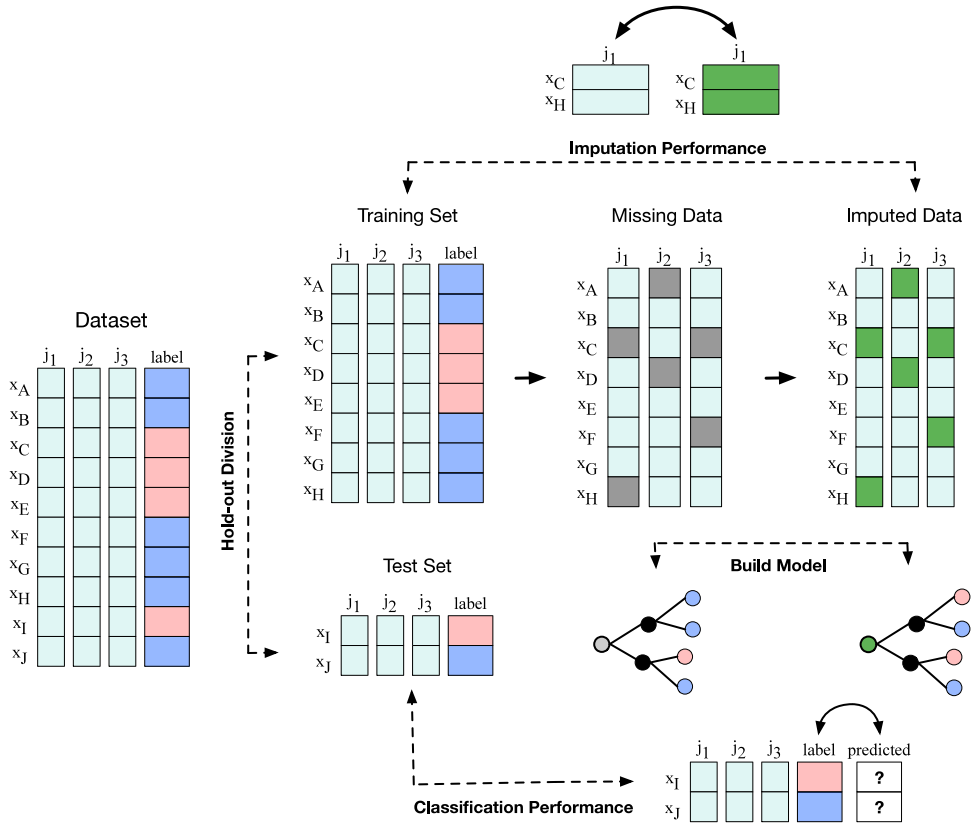



**Fig. 2.** An overview of the experimental setup. Each complete dataset is first divided into a training and test partitions, and the training set is subjected to loss in some features (missing values are synthetically introduced). Then, using kNNI with distinct distance functions, the training set containing missing values is imputed and becomes complete. The evaluation of classification performance is performed by comparing the predictions of a decision tree model built with an incomplete training set with one built using the imputed training set, over the same test data. In turn, the quality of imputation in evaluated by analysing the difference between the true values in data (original training set) with those generated by the kNNI approach (imputed training set).

### 6.1. Overall effect on kNN imputation

Tables 3, 4, 5 and 6 report on the overall performance results of CART classification for $k = 1$, 3, 5 and 7, respectively, considering 8 approaches: training sets with missing values (BASELINE) and training sets imputed with 7 different distances: HEOM, HEOM-R, HVDM, HVDM-R, HVDM-S, MDE and SIMDIST. The results consider the average Sensitivity (*Sens*), F-measure (*F1*) and *G-mean* obtained for missing rates (MRs) of 5, 10, 20 and 30% on all datasets. The top performing approach for each performance metric is marked in bold.

The first observation is that, overall, for all $k$ values, MRs, and performance metrics, classifiers constructed from imputed data obtain higher classification results than those learned from data with missing values, i.e., datasets imputed with kNN (for any distance function) outperform the BASELINE results. An exception occurs for $k = 1$, where, for a MR of 30%, CART models trained with missing values obtain higher Sensitivity and F1 results than all distance functions, except the top 2 performing distances, HVDM-S and MDE (Table 3).

Additionally, as the missing rate increases, so does the difference between the BASELINE and the top kNN imputation approach, for all $k$ values. The difference between the results obtained by the considered distance functions also becomes more noticeable with increasing amounts of missing data, especially for $k = 1$ and 3 (Tables 3 and 4). For a MR of 5%, the classification results obtained with each distance function are close, with a difference from the best to worst distance function of 0.001 ($k = 1$) and 0.002–0.003 ($k = 3$), whereas for a MR of 30%, differences increase to 0.015–0.022 ($k = 1$) and 0.009–0.013 ($k = 3$).[3] For higher values of $k$, although differences between distance functions increase with the missing rate, differences are more subtle (Tables 5 and 6).

Another important observation is that, whereas for MRs of 5 and 10% distances behave similarly, with SIMDIST, HVDM-R, HVDM-S and MDE among the top performing approaches ($k = 1$ and 3), for MRs of 20 and 30%, HVDM-S and MDE present superior performance results (for $k = 1$ and 3, HVDM-S is the top performing approach for a MR of 20%, whereas MDE seems superior for 30%). As expected, for $k = 5$ and 7, the best results become more scattered across other distance functions. Nevertheless, for these values of $k$, HVDM-S is consistently the best approach for MRs of 20% and 30% (SIMDIST also appears as a top performer for a MR of 5% in both scenarios).

These results suggest that for a dataset with given, invariable, characteristics (imbalance ratio, number of categorical and continuous features, number of samples), the choice of the best distance function is often dependent on the missing rate. Given these findings, we proceed to analyse the datasets by category (continuous, categorical and heterogeneous datasets) in order to assess the behaviour of each distance function in different contexts. To that end, a ranking strategy is used.

As previously explained, the majority of the considered datasets are imbalanced, which is a frequent problem in several domains (Das et al., 2018). Therefore, we focus on sensitivity results for the following analysis, where a particular importance is given to correct predictions of the minority class, which is considered to be the concept of interest (positive class).

Firstly, datasets were divided into three groups (*Continuous Datasets*, *Categorical Datasets* and *Heterogeneous Datasets*) and for each missing rate (5, 10, 20 and 30%) and $k$ value (1, 3, 5 and 7), all approaches are ranked for each dataset based on the obtained sensitivity results. Then, the average rank of each approach is determined and a statistical analysis is conducted.

[3] These values concern the difference between the best and worst results obtained by distance functions, considering all classification metrics.

**Table 3**
CART performance results without imputation (BASELINE) and with kNN imputation ($k$ = 1) using several distance functions. Best results are marked in bold.

| *Distance* | *MR* | *Sens* | *F1* | *G-mean* | *MR* | *Sens* | *F1* | *G-mean* |
|---|---|---|---|---|---|---|---|---|
| BASELINE | | 0.524 ± 0.326 | 0.527 ± 0.323 | 0.583 ± 0.313 | | 0.520 ± 0.328 | 0.522 ± 0.325 | 0.577 ± 0.316 |
| HEOM | | 0.530 ± 0.327 | 0.532 ± 0.324 | 0.588 ± 0.313 | | 0.524 ± 0.328 | 0.526 ± 0.325 | 0.580 ± 0.314 |
| HEOM-REDEF | | 0.530 ± 0.326 | 0.532 ± 0.324 | 0.588 ± 0.312 | | 0.522 ± 0.327 | 0.525 ± 0.324 | 0.580 ± 0.313 |
| HVDM | *5%* | 0.530 ± 0.326 | 0.532 ± 0.324 | 0.588 ± 0.312 | *10%* | 0.523 ± 0.325 | 0.526 ± 0.322 | 0.581 ± 0.310 |
| HVDM-REDEF | | 0.530 ± 0.327 | **0.533** ± 0.323 | **0.589** ± 0.312 | | 0.521 ± 0.326 | 0.524 ± 0.323 | 0.579 ± 0.312 |
| HVDM-S | | 0.530 ± 0.326 | **0.533** ± 0.322 | **0.589** ± 0.311 | | **0.527** ± 0.324 | 0.529 ± 0.320 | **0.585** ± 0.308 |
| MDE | | 0.530 ± 0.326 | 0.532 ± 0.322 | 0.588 ± 0.311 | | **0.527** ± 0.322 | **0.530** ± 0.320 | **0.585** ± 0.308 |
| SIMDIST | | **0.531** ± 0.327 | 0.532 ± 0.324 | 0.588 ± 0.313 | | 0.525 ± 0.327 | 0.528 ± 0.323 | 0.583 ± 0.311 |
| BASELINE | | 0.504 ± 0.328 | 0.505 ± 0.326 | 0.558 ± 0.320 | | 0.490 ± 0.328 | 0.491 ± 0.326 | 0.542 ± 0.322 |
| HEOM | | 0.508 ± 0.321 | 0.511 ± 0.318 | 0.568 ± 0.308 | | 0.484 ± 0.319 | 0.485 ± 0.315 | 0.544 ± 0.307 |
| HEOM-REDEF | | 0.505 ± 0.323 | 0.508 ± 0.318 | 0.565 ± 0.310 | | 0.483 ± 0.319 | 0.485 ± 0.315 | 0.542 ± 0.308 |
| HVDM | *20%* | 0.509 ± 0.321 | 0.510 ± 0.318 | 0.568 ± 0.308 | *30%* | 0.486 ± 0.319 | 0.486 ± 0.314 | 0.543 ± 0.308 |
| HVDM-REDEF | | 0.503 ± 0.321 | 0.507 ± 0.317 | 0.563 ± 0.309 | | 0.485 ± 0.319 | 0.487 ± 0.314 | 0.544 ± 0.308 |
| HVDM-S | | **0.515** ± 0.317 | **0.517** ± 0.314 | **0.576** ± 0.303 | | 0.497 ± 0.318 | 0.496 ± 0.311 | 0.556 ± 0.303 |
| MDE | | 0.513 ± 0.323 | 0.514 ± 0.318 | 0.572 ± 0.308 | | **0.505** ± 0.321 | **0.500** ± 0.316 | **0.560** ± 0.308 |
| SIMDIST | | 0.508 ± 0.323 | 0.511 ± 0.318 | 0.567 ± 0.309 | | 0.484 ± 0.318 | 0.487 ± 0.315 | 0.543 ± 0.307 |

**Table 4**
CART performance results without imputation (BASELINE) and with kNN imputation ($k$ = 3) using several distance functions.

| *Distance* | *MR* | *Sens* | *F1* | *G-mean* | *MR* | *Sens* | *F1* | *G-mean* |
|---|---|---|---|---|---|---|---|---|
| BASELINE | | 0.524 ± 0.326 | 0.527 ± 0.323 | 0.583 ± 0.313 | | 0.520 ± 0.328 | 0.522 ± 0.325 | 0.577 ± 0.316 |
| HEOM | | 0.533 ± 0.326 | 0.534 ± 0.322 | 0.590 ± 0.311 | | 0.530 ± 0.325 | 0.530 ± 0.322 | 0.586 ± 0.310 |
| HEOM-REDEF | | 0.531 ± 0.328 | 0.533 ± 0.324 | 0.589 ± 0.312 | | 0.526 ± 0.326 | 0.527 ± 0.322 | 0.583 ± 0.311 |
| HVDM | *5%* | 0.532 ± 0.326 | 0.534 ± 0.323 | 0.591 ± 0.310 | *10%* | 0.530 ± 0.326 | 0.531 ± 0.322 | 0.587 ± 0.310 |
| HVDM-REDEF | | **0.534** ± 0.327 | **0.535** ± 0.324 | **0.592** ± 0.311 | | 0.527 ± 0.327 | 0.528 ± 0.323 | 0.583 ± 0.312 |
| HVDM-S | | 0.533 ± 0.327 | 0.534 ± 0.324 | 0.590 ± 0.311 | | 0.530 ± 0.325 | 0.531 ± 0.320 | 0.587 ± 0.308 |
| MDE | | 0.532 ± 0.325 | 0.533 ± 0.321 | 0.591 ± 0.308 | | **0.532** ± 0.324 | **0.532** ± 0.321 | 0.589 ± 0.309 |
| SIMDIST | | **0.534** ± 0.326 | **0.535** ± 0.322 | **0.592** ± 0.310 | | **0.532** ± 0.326 | **0.532** ± 0.323 | **0.590** ± 0.309 |
| BASELINE | | 0.504 ± 0.328 | 0.505 ± 0.326 | 0.558 ± 0.320 | | 0.490 ± 0.328 | 0.491 ± 0.326 | 0.542 ± 0.322 |
| HEOM | | 0.516 ± 0.320 | 0.515 ± 0.316 | 0.574 ± 0.306 | | 0.504 ± 0.321 | 0.498 ± 0.315 | 0.559 ± 0.306 |
| HEOM-REDEF | | 0.513 ± 0.321 | 0.513 ± 0.317 | 0.572 ± 0.307 | | 0.499 ± 0.321 | 0.495 ± 0.314 | 0.555 ± 0.305 |
| HVDM | *20%* | 0.516 ± 0.325 | 0.515 ± 0.319 | 0.573 ± 0.309 | *30%* | 0.501 ± 0.319 | 0.496 ± 0.312 | 0.557 ± 0.303 |
| HVDM-REDEF | | 0.514 ± 0.320 | 0.513 ± 0.316 | 0.572 ± 0.305 | | 0.498 ± 0.321 | 0.493 ± 0.315 | 0.553 ± 0.307 |
| HVDM-S | | **0.522** ± 0.319 | **0.520** ± 0.314 | **0.581** ± 0.301 | | 0.510 ± 0.317 | **0.504** ± 0.310 | **0.566** ± 0.301 |
| MDE | | 0.519 ± 0.321 | 0.517 ± 0.317 | 0.577 ± 0.306 | | **0.511** ± 0.321 | **0.504** ± 0.314 | 0.565 ± 0.306 |
| SIMDIST | | 0.519 ± 0.323 | 0.519 ± 0.318 | 0.577 ± 0.307 | | 0.504 ± 0.318 | 0.499 ± 0.312 | 0.559 ± 0.303 |

**Table 5**
CART performance results without imputation (BASELINE) and with kNN imputation ($k$ = 5) using several distance functions.

| *Distance* | *MR* | *Sens* | *F1* | *G-mean* | *MR* | *Sens* | *F1* | *G-mean* |
|---|---|---|---|---|---|---|---|---|
| BASELINE | | 0.524 ± 0.326 | 0.527 ± 0.323 | 0.583 ± 0.313 | | 0.520 ± 0.328 | 0.522 ± 0.325 | 0.577 ± 0.316 |
| HEOM | | 0.533 ± 0.327 | 0.534 ± 0.324 | 0.590 ± 0.312 | | **0.532** ± 0.326 | **0.532** ± 0.322 | **0.588** ± 0.310 |
| HEOM-REDEF | | 0.532 ± 0.326 | 0.534 ± 0.323 | 0.590 ± 0.312 | | 0.530 ± 0.328 | 0.531 ± 0.323 | 0.587 ± 0.311 |
| HVDM | *5%* | 0.533 ± 0.325 | 0.535 ± 0.322 | 0.592 ± 0.309 | *10%* | 0.531 ± 0.328 | 0.531 ± 0.323 | 0.587 ± 0.312 |
| HVDM-REDEF | | 0.533 ± 0.326 | 0.535 ± 0.322 | 0.592 ± 0.309 | | 0.529 ± 0.329 | 0.529 ± 0.324 | 0.585 ± 0.313 |
| HVDM-S | | 0.532 ± 0.327 | 0.535 ± 0.323 | 0.591 ± 0.310 | | 0.529 ± 0.325 | 0.530 ± 0.321 | 0.587 ± 0.308 |
| MDE | | 0.532 ± 0.324 | 0.534 ± 0.321 | 0.591 ± 0.309 | | 0.529 ± 0.326 | 0.530 ± 0.323 | 0.586 ± 0.312 |
| SIMDIST | | **0.535** ± 0.326 | **0.536** ± 0.322 | **0.593** ± 0.310 | | 0.530 ± 0.328 | 0.530 ± 0.323 | 0.587 ± 0.311 |
| BASELINE | | 0.504 ± 0.328 | 0.505 ± 0.326 | 0.558 ± 0.320 | | 0.490 ± 0.328 | 0.491 ± 0.326 | 0.542 ± 0.322 |
| HEOM | | 0.522 ± 0.323 | **0.521** ± 0.318 | 0.579 ± 0.307 | | 0.503 ± 0.321 | 0.497 ± 0.314 | 0.557 ± 0.305 |
| HEOM-REDEF | | 0.513 ± 0.321 | 0.511 ± 0.317 | 0.571 ± 0.306 | | 0.507 ± 0.323 | 0.501 ± 0.314 | 0.562 ± 0.306 |
| HVDM | *20%* | 0.522 ± 0.323 | 0.520 ± 0.317 | 0.579 ± 0.306 | *30%* | 0.506 ± 0.323 | 0.499 ± 0.316 | 0.560 ± 0.308 |
| HVDM-REDEF | | 0.518 ± 0.325 | 0.515 ± 0.319 | 0.574 ± 0.309 | | 0.503 ± 0.324 | 0.498 ± 0.316 | 0.558 ± 0.308 |
| HVDM-S | | **0.525** ± 0.321 | **0.521** ± 0.315 | **0.582** ± 0.303 | | **0.512** ± 0.323 | **0.504** ± 0.314 | **0.566** ± 0.305 |
| MDE | | 0.521 ± 0.321 | 0.518 ± 0.317 | 0.578 ± 0.307 | | 0.506 ± 0.323 | 0.501 ± 0.316 | 0.561 ± 0.308 |
| SIMDIST | | 0.519 ± 0.322 | 0.519 ± 0.317 | 0.576 ± 0.307 | | 0.506 ± 0.320 | 0.500 ± 0.314 | 0.561 ± 0.304 |

To determine whether there is a statistically significant difference among approaches (for each group, missing rate and $k$ value), the Friedman test was run under the null-hypothesis that the performance of all approaches is equivalent (Demšar, 2006). For each group of datasets, missing percentage and $k$ value, the $F_F$ statistic is computed and compared with the established critical value for the F-distribution at a 5% significance level, $F_c$ (Table 7).

Considering all groups and $k$ values, the Friedman test did not detect any statistically significant differences between the approaches for missing rates of 5% and 10% (for these MRs, the calculated $F_F$ statistic is not superior to the established critical value $F_c$ and therefore the null hypothesis could not be rejected). This is also true for some combinations of groups, MR and $k$, as established from Table 7. Apart from these exceptions, as the missing rate increases (20 and 30%), the null hypothesis of equivalence between approaches is rejected, even for increasing values of $k$. This indicates that although $k$-parametrisation plays an important role on the optimisation of kNN imputation results, it is important not to overlook the distance function hyperparameter, as it seems to play an important role on determining the best approach, especially for higher missing rates.

Since the null-hypothesis was often rejected for higher missing rates 20 and 30%, the Nemenyi test was applied to post-hoc testing (at a 5% significance level), to compare all methods against each other. Tables 8, 9, 10 and 11 show the average sensitivity ranks of each

**Table 6**
CART performance results without imputation (BASELINE) and with kNN imputation ($k$ = 7) using several distance functions.

| *Distance* | *MR* | *Sens* | *F1* | *G-mean* | *MR* | *Sens* | *F1* | *G-mean* |
|---|---|---|---|---|---|---|---|---|
| BASELINE | ***5%*** | 0.524 ± 0.326 | 0.527 ± 0.323 | 0.583 ± 0.313 | ***10%*** | 0.520 ± 0.328 | 0.522 ± 0.325 | 0.577 ± 0.316 |
| HEOM | | 0.534 ± 0.326 | 0.535 ± 0.322 | 0.593 ± 0.310 | | 0.531 ± 0.326 | **0.532** ± 0.322 | **0.588** ± 0.310 |
| HEOM-REDEF | | 0.534 ± 0.326 | 0.535 ± 0.322 | 0.593 ± 0.309 | | 0.530 ± 0.326 | 0.530 ± 0.321 | 0.587 ± 0.310 |
| HVDM | | 0.533 ± 0.326 | 0.535 ± 0.323 | 0.592 ± 0.310 | | **0.532** ± 0.328 | **0.532** ± 0.324 | **0.588** ± 0.313 |
| HVDM-REDEF | | 0.533 ± 0.326 | 0.535 ± 0.322 | 0.592 ± 0.309 | | 0.528 ± 0.328 | 0.529 ± 0.324 | 0.585 ± 0.312 |
| HVDM-S | | 0.534 ± 0.326 | **0.536** ± 0.322 | 0.593 ± 0.309 | | 0.531 ± 0.325 | 0.531 ± 0.321 | **0.588** ± 0.308 |
| MDE | | 0.533 ± 0.327 | 0.534 ± 0.323 | 0.591 ± 0.311 | | 0.530 ± 0.326 | 0.531 ± 0.323 | 0.587 ± 0.312 |
| SIMDIST | | **0.535** ± 0.325 | **0.536** ± 0.322 | **0.594** ± 0.309 | | 0.531 ± 0.328 | 0.531 ± 0.324 | **0.588** ± 0.312 |
| BASELINE | ***20%*** | 0.504 ± 0.328 | 0.505 ± 0.326 | 0.558 ± 0.320 | ***30%*** | 0.490 ± 0.328 | 0.491 ± 0.326 | 0.542 ± 0.322 |
| HEOM | | 0.521 ± 0.321 | 0.518 ± 0.316 | 0.579 ± 0.304 | | 0.506 ± 0.319 | 0.500 ± 0.312 | 0.562 ± 0.303 |
| HEOM-REDEF | | 0.518 ± 0.324 | 0.516 ± 0.317 | 0.575 ± 0.306 | | 0.505 ± 0.322 | 0.499 ± 0.315 | 0.561 ± 0.307 |
| HVDM | | 0.523 ± 0.322 | 0.522 ± 0.317 | 0.581 ± 0.305 | | 0.508 ± 0.321 | 0.501 ± 0.313 | 0.562 ± 0.305 |
| HVDM-REDEF | | 0.518 ± 0.325 | 0.515 ± 0.318 | 0.575 ± 0.306 | | 0.503 ± 0.323 | 0.497 ± 0.315 | 0.557 ± 0.307 |
| HVDM-S | | **0.527** ± 0.322 | **0.523** ± 0.315 | **0.583** ± 0.302 | | **0.509** ± 0.320 | **0.502** ± 0.312 | **0.564** ± 0.303 |
| MDE | | 0.523 ± 0.320 | 0.520 ± 0.316 | 0.581 ± 0.304 | | **0.509** ± 0.322 | **0.502** ± 0.316 | 0.562 ± 0.306 |
| SIMDIST | | 0.520 ± 0.323 | 0.518 ± 0.317 | 0.578 ± 0.305 | | 0.508 ± 0.324 | 0.500 ± 0.317 | 0.561 ± 0.308 |

**Table 7**
$F_F$ statistic calculated for each group of datasets, divided by missing rates and $k$ values. Highlighted values (shaded in grey) indicate statistically significant differences between the approaches (Baseline and kNN imputation with different distance functions).

| | *k* | *5%* | *10%* | *20%* | *30%* |
|---|---|---|---|---|---|
| | *1* | 1.28 | 1.34 | 1.24 | 3.72 |
| ***Continuous*** | *3* | 1.31 | 1.46 | 3.16 | 0.84 |
| ***Datasets*** | *5* | 0.76 | 1.53 | 3.53 | 1.74 |
| ($F_c$ = 2.14) | *7* | 1.82 | 0.92 | 3.90 | 0.40 |
| | *1* | 0.78 | 1.51 | 2.02 | 5.17 |
| ***Categorical*** | *3* | 0.68 | 0.61 | 2.12 | 4.20 |
| ***Datasets*** | *5* | 0.78 | 0.21 | 2.42 | 2.29 |
| ($F_c$ = 2.06) | *7* | 0.63 | 1.23 | 3.96 | 2.18 |
| | *1* | 0.41 | 1.17 | 2.19 | 3.86 |
| ***Heterogeneous*** | *3* | 0.98 | 0.73 | 1.40 | 3.07 |
| ***Datasets*** | *5* | 0.58 | 0.59 | 1.86 | 4.23 |
| ($F_c$ = 2.06) | *7* | 1.04 | 0.82 | 3.97 | 3.12 |

**Table 8**
Average sensitivity ranks per missing rate, divided by groups ($k$ = 1). Critical differences for Nemenyi test ($CD_n$) are shown for each group of datasets. Lowest ranks (best results) are marked in bold. Significant differences in comparison to the best approach are shaded in grey.

| | *MR* | *B* | *HEOM* | *HEOM-R* | *HVDM* | *HVDM-R* | [a]*HVDM-S* | *MDE* | *SIMDIST* |
|---|---|---|---|---|---|---|---|---|---|
| | ***5%*** | 4.31 | 4.18 | 4.39 | 3.79 | 3.95 | – | **3.38** | 4.00 |
| ***Continuous*** | ***10%*** | 4.09 | 3.72 | 4.37 | 4.05 | 4.37 | – | **3.38** | 4.02 |
| ***Datasets*** | ***20%*** | 4.49 | 4.10 | 4.08 | 4.15 | 4.05 | – | **3.44** | 3.69 |
| ($CD_n$ = 1.27) | ***30%*** | 3.67 | 3.90 | 4.63 | 4.19 | 4.25 | – | **2.92** | 4.44 |
| | ***5%*** | 5.06 | 4.25 | 4.46 | 4.45 | 4.61 | **4.18** | 4.78 | 4.21 |
| ***Categorical*** | ***10%*** | 4.25 | 4.77 | 4.77 | 4.90 | 4.77 | **3.61** | 4.37 | 4.56 |
| ***Datasets*** | ***20%*** | 4.51 | 4.39 | 4.88 | 4.61 | 5.24 | **3.63** | 4.09 | 4.65 |
| ($CD_n$ = 1.48) | ***30%*** | 4.58 | 5.04 | 4.57 | 5.11 | 4.84 | **3.09** | 3.61 | 5.16 |
| | ***5%*** | 4.79 | 4.69 | 4.21 | 4.59 | 4.55 | 4.44 | **4.15** | 4.58 |
| ***Heterogeneous*** | ***10%*** | 4.56 | 4.40 | 4.91 | 4.29 | 5.17 | **4.01** | 4.37 | 4.29 |
| ***Datasets*** | ***20%*** | 4.27 | 4.63 | 4.86 | 4.03 | 5.34 | **3.84** | 4.17 | 4.86 |
| ($CD_n$ = 1.48) | ***30%*** | 5.12 | 5.14 | 4.95 | 4.45 | 4.83 | 3.61 | **3.42** | 4.48 |

*B: BASELINE.*

[a]For continuous datasets, HVDM-S is equivalent to HVDM-R.

approach, considering each group and missing rate, for $k$ = 1, 3, 5 and 7, respectively. The winning method (with the lowest rank) is marked in bold and statistically significant differences between the best approach and the remaining are shaded in grey.

Regarding $k$ = 1, the best method is consistent over all MRs for continuous and categorical datasets (Table 8). For continuous datasets, MDE stands out as the winning approach, whereas for categorical datasets, HVDM-S is the best performing approach. For heterogeneous datasets, MDE and HVDM-S are the top performing approaches, with HVDM-S obtaining higher performance results for intermediate MRs (10 and 20%), whereas MDE obtains the lowest ranks for MRs of 5 and 30%.

Results obtained for $k$ = 3 are similar (Table 9), where MDE and HVDM-S figure consistently among the best approaches. On contrary, the best results for $k$ = 5 and 7 (Tables 10 and 11), are more scattered across other approaches. Nevertheless, HVDM-S remains among the top approaches for categorical and heterogeneous data: for $k$ = 5, HVDM-S is considered the best approach on MRs of 20 and 30% for both categorical and heterogeneous datasets, and for $k$ = 7, it remains the best approach for categorical data (all MRs), and heterogeneous data (10 and 20%). This confirms the rational that $k$ is not the sole parameter that should generally be tuned when developing kNN imputation approaches, since the distance function has shown to affect data imputation, particularly for categorical and heterogeneous datasets.

**Table 9**
Average sensitivity ranks per missing rate, divided by groups ($k = 3$). Critical differences for Nemenyi test ($CD_n$) are shown for each group of datasets. Lowest ranks (best results) are marked in bold. Significant differences in comparison to the best approach are shaded in grey.

| | MR | B | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|---|
| | ***5%*** | 4.71 | 3.84 | 4.04 | 4.11 | **3.64** | – | 3.84 | 3.82 |
| ***Continuous*** | ***10%*** | 4.36 | 3.73 | 4.39 | 3.76 | 4.22 | – | **3.42** | 4.12 |
| ***Datasets*** | ***20%*** | 5.08 | 3.99 | 3.94 | 3.71 | 4.10 | – | **3.37** | 3.81 |
| ($CD_n = 1.27$) | ***30%*** | 4.35 | 4.00 | 4.07 | 3.84 | 4.04 | – | **3.48** | 4.22 |
| | ***5%*** | 5.05 | 4.34 | 4.64 | 4.60 | 4.39 | **4.12** | 4.59 | 4.27 |
| ***Categorical*** | ***10%*** | 4.53 | 4.56 | 4.73 | 4.59 | 4.89 | **4.00** | 4.40 | 4.30 |
| ***Datasets*** | ***20%*** | 4.73 | 4.35 | 4.85 | 4.65 | 5.24 | **3.55** | 4.31 | 4.32 |
| ($CD_n = 1.48$) | ***30%*** | 5.01 | 4.37 | 4.48 | 5.32 | 4.85 | **3.22** | 3.84 | 4.91 |
| | ***5%*** | 4.83 | 4.75 | 4.86 | **3.97** | 4.75 | 4.38 | 4.14 | 4.32 |
| ***Heterogeneous*** | ***10%*** | 4.84 | 4.34 | 4.83 | 4.46 | 4.70 | **4.03** | 4.60 | 4.20 |
| ***Datasets*** | ***20%*** | 4.45 | 4.93 | 4.90 | 4.80 | 4.51 | **3.84** | 3.97 | 4.60 |
| ($CD_n = 1.48$) | ***30%*** | 5.59 | 4.52 | 4.62 | 4.69 | 4.86 | 3.94 | **3.73** | 4.05 |

**Table 10**
Average sensitivity ranks per missing rate, divided by groups ($k = 5$). Critical differences for Nemenyi test ($CD_n$) are shown for each group of datasets. Lowest ranks (best results) are marked in bold. Significant differences in comparison to the best approach are shaded in grey.

| | MR | B | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|---|
| | ***5%*** | 4.47 | 3.96 | 3.96 | **3.70** | 4.00 | – | 4.19 | 3.72 |
| ***Continuous*** | ***10%*** | 4.67 | 4.15 | **3.50** | 3.89 | 3.69 | – | 4.13 | 3.97 |
| ***Datasets*** | ***20%*** | 5.14 | **3.53** | 4.26 | 3.56 | 3.72 | – | 3.92 | 3.87 |
| ($CD_n = 1.27$) | ***30%*** | 4.69 | 4.26 | 4.20 | **3.61** | 3.86 | – | **3.61** | 3.77 |
| | ***5%*** | 5.05 | 4.53 | 4.54 | 4.57 | 4.72 | 4.15 | **4.11** | 4.33 |
| ***Categorical*** | ***10%*** | 4.59 | 4.37 | 4.60 | 4.51 | 4.76 | 4.41 | 4.52 | **4.24** |
| ***Datasets*** | ***20%*** | 4.88 | 4.47 | 5.03 | 4.80 | 4.66 | **3.33** | 4.26 | 4.57 |
| ($CD_n = 1.48$) | ***30%*** | 4.94 | 4.62 | 4.70 | 4.76 | 4.99 | **3.43** | 4.08 | 4.48 |
| | ***5%*** | 5.10 | 4.44 | 4.60 | 4.54 | 4.35 | 4.37 | 4.32 | **4.28** |
| ***Heterogeneous*** | ***10%*** | 4.90 | **4.20** | 4.81 | 4.28 | 4.36 | 4.25 | 4.61 | 4.59 |
| ***Datasets*** | ***20%*** | 4.80 | 4.38 | 5.42 | 4.08 | 4.33 | **3.97** | 4.29 | 4.73 |
| ($CD_n = 1.48$) | ***30%*** | 5.67 | 5.29 | 4.04 | 4.48 | 4.40 | **3.47** | 4.36 | 4.29 |

**Table 11**
Average sensitivity ranks per missing rate, divided by groups ($k = 7$). Critical differences for Nemenyi test ($CD_n$) are shown for each group of datasets. Lowest ranks (best results) are marked in bold. Significant differences in comparison to the best approach are shaded in grey.

| | MR | B | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|---|
| | ***5%*** | 4.71 | **3.41** | 3.94 | 3.71 | 3.91 | – | 4.19 | 4.13 |
| ***Continuous*** | ***10%*** | 4.55 | 3.84 | 4.05 | 3.88 | 3.92 | – | **3.62** | 4.14 |
| ***Datasets*** | ***20%*** | 5.27 | 3.97 | 3.63 | 3.67 | 3.94 | – | 3.97 | **3.55** |
| ($CD_n = 1.27$) | ***30%*** | 4.40 | 3.92 | 3.92 | **3.86** | 4.10 | – | 3.94 | **3.86** |
| | ***5%*** | 4.96 | 4.61 | 4.77 | 4.34 | 4.58 | **4.18** | 4.32 | 4.24 |
| ***Categorical*** | ***10%*** | 4.49 | 4.43 | 4.66 | 4.43 | 5.20 | **3.97** | 4.73 | 4.09 |
| ***Datasets*** | ***20%*** | 5.05 | 4.92 | 5.23 | 4.66 | 4.64 | **3.14** | 4.30 | 4.06 |
| ($CD_n = 1.48$) | ***30%*** | 4.93 | 4.80 | 4.98 | 4.46 | 4.82 | **3.47** | 4.18 | 4.36 |
| | ***5%*** | 5.09 | 4.55 | **4.08** | 4.95 | 4.30 | 4.40 | 4.43 | 4.20 |
| ***Heterogeneous*** | ***10%*** | 4.80 | 4.49 | 4.83 | 4.50 | 4.50 | **3.98** | 4.78 | 4.12 |
| ***Datasets*** | ***20%*** | 5.20 | 4.44 | 4.88 | 4.27 | 4.81 | **3.41** | 3.73 | 5.26 |
| ($CD_n = 1.48$) | ***30%*** | 5.88 | 4.66 | 4.10 | **4.06** | 4.59 | 4.27 | 4.14 | 4.30 |

Considering the obtained experimental results, we establish that distance functions significantly affect kNN imputation and their performance is related to the amount of missing data. However, besides the presence of missing data, the performance of distance functions differs according to the nature of datasets, showing that it is important to isolate each component of the distance functions definition to fully characterise their behaviour.

In what follows, we analyse the behaviour of distance functions by isolating certain components of the distance computation between patterns. In particular, we start by studying continuous and categorical datasets individually and assess the impact of increasing MRs on the performance of distance functions. Then, the insights extracted from this analysis are cross-correlated with the results obtained for the heterogeneous datasets.

We focus on a more local behaviour of kNN, by analysing the results obtained with $k = 1$. As $k$ increases, the neighbourhood of a given pattern becomes larger, and it is expected that differences between distance functions become more smoothed, as previously discussed and confirmed by the overall performance results (Tables 3, 4, 5 and 6). Therefore, to allow a more thorough analysis on the behaviour of distance functions regarding the definition of each component, we consider the smallest neighbourhood: for $k = 1$, differences between distance functions will mainly rely on their definition, whereas for higher values of $k$, it becomes more difficult to distinguish the effects associated with the definition of distance functions from the increase of the $k$-neighbourhood. Despite the focus on $k = 1$, results obtained for additional values of $k$ (3, 5 and 7) are also discussed throughout this section.

### 6.2. Effect of function definition on distance computation

Throughout this section, we discuss how each component of the definition of functions affects the computation of distances, focusing

mostly on imputation results for $k = 1$ for a more local analysis. We start by cross-referencing the results presented in Tables 8 and 12.

Table 12 considers the pairwise differences between all distances: the values correspond to the difference between the ranks of the approaches in the corresponding rows and columns. Thus, positive differences indicate that the approach in the columns is better than the one in the rows, whereas negative differences indicate that the approach in the rows is better (significant differences are marked in bold). Furthermore, differences for 5 and 20% are shown in the upper part of the tables, whereas differences for 10 and 30% are presented in the lower part of the tables (and also shaded in grey).

We now tailor our analysis to the individual categories of datasets, by cross-referencing the information of Tables 8 and 12.

#### 6.2.1. Continuous datasets

For continuous datasets, MDE outperforms the remaining approaches for all missing rates, although for MRs of 5, 10 and 20% no significant differences were found (Table 8). However, for a MR of 30%, MDE achieves an average rank of 2.92, and the post-hoc concluded on its superiority over HEOM-R, HVDM, HVDM-R and SIMDIST (Tables 8 and 12). The difference for HEOM was reasonable (0.98) but not higher than the critical value (1.27).

An insightful observation is on the comparison of HEOM and HVDM with their redefinitions: HEOM-R and HVDM-R perform worse than their original formulations, suggesting that considering two missing values as being equal seems rigid and may be prejudicial for imputation (Table 12).

Regarding the remaining distances, HEOM, HVDM and SIMDIST behave somewhat similarly, except for a MR of 30%, where HEOM presents a considerably lower rank (3.90 versus 4.19/4.44). Table 12 indicates that HEOM overall performs slightly better than HVDM, which should be due to normalisation differences (Eqs. (3) and (6)). Thus, differences between HEOM, HVDM and MDE rely mostly on the treatment of missing data: we may infer that considering a distance of 1 if either $x_{Aj}$ and $x_{Bj}$ are missing also seems inadequate given the superiority of MDE over these distance functions.

Furthermore, despite some similarities in working principles of MDE and SIMDIST (considering the average distance between patterns to impute missing values), there seems to be an advantage in distinguish situations where one or both values are missing, causing MDE to be top performing approach, as no other distance distinguishes between such scenarios. An additional advantage of MDE over the remaining distances could be related to the fact that MDE takes the probability distribution of each feature into account while computing distances.

For $k = 3$, results are similar, with MDE being the top performing distance (Table 9). However, for $k$ values of 5 and 7, differences in classification performance become negligible (Tables 10 and 11). Overall, significant differences between approaches also cease to exist, due to loss of locality in kNN parametrisation.

#### 6.2.2. Categorical datasets

For categorical datasets, HVDM-S stands out as the best approach for all missing rates (Table 8).

An interesting topic for discussion is the comparison between HVDM-S, MDE and HVDM. As shown in Table 8, despite HVDM-S achieves lower ranks than MDE, the equivalence between the two distance functions is never rejected, not even for the highest missing rate. In turn, HVDM-S is significantly better than the remaining approaches for a MR of 30%. Then, a comparison of MDE with HVDM becomes insightful. Although the computation of categorical distances is different in this case (MDE uses the overlap metric while HVDM uses $d_{vdm}$ when both values are observed) the performance of both distances is not significantly different (Table 12). For 5%, HVDM is slightly better than MDE (perhaps due to the computation of $d_{vdm}$) but rapidly looses its advantage as the missing rate increases: for a MR of 30%, MDE is even significantly better than HVDM (Table 8). In turn, HVDM-S, whose definition is very close to HVDM, always surpasses MDE (Table 8). This indicates that it is the treatment of missing data (the only aspect that changes between HVDM-S and HVDM) that is responsible for the good results achieved.

Contrarily to continuous datasets, using the average distance to compute the distance between missing patterns is not the best overall approach: in this case, the ability of HVDM-S to consider the distribution of missing values in each class could be one of its greatest advantages.

Another interesting point is that, for categorical datasets, HVDM-S remains the top performing approach for larger values of $k$. For $k = 5$, MDE and SIMDIST achieve the top positions for MRs 5 and 10%, respectively (Table 10), but for $k = 3$ and 7, HVDM-S assumes the leading position for all MRs (Tables 9 and 11). Significant differences are found for some distances, where the most clear improvement is on $k = 7$ for a MR of 20%, where HVDM-S is significantly superior to all distances except MDE and SIMDIST (Table 11).

#### 6.2.3. Heterogeneous datasets

For heterogeneous datasets, MDE or HVDM-S appear as the winning approaches for all missing rates (Table 8). For a MR of 5%, MDE is the top performing approach, whereas for 10 and 20%, HVDM-S becomes superior. For a MR of 30%, both approaches behave similarly (3.61 versus 3.42 obtained by HVDM-S and MDE, respectively).

A similar trend is observed for higher values of $k$, in what concerns HVDM-S: for $k = 3$ and 7, it achieves the top results for intermediate MRs of 10 and 20% (Tables 9 and 11), whereas for $k = 5$ it is the top performer for 20 and 30% (Table 10). In turn, MDE, although presenting good results for more local neighbourhoods ($k = 3$), is not the best approach for higher $k$ values. In fact, for extreme levels of MR (5 or 30%), there is not a consensus on the best approach for higher values of $k$.

Given the results obtained for continuous and categorical datasets ($k = 1$), where MDE is the top performing approach for continuous datasets for all rates and HVDM-S is the best for categorical datasets, these results on heterogeneous datasets are somewhat expected. It would be important, however, to determine the components of each distance that affect the most the results in the case of heterogeneous data.

For a lower MR of 5%, where most values are expected to be observed, the results obtained by the two approaches do not considerably differ. When both $x_{Aj}$ and $x_{Bj}$ values are observed, differences among the two distance functions rely on the normalisation of continuous features (MDE seems to perform better according to the results obtained for continuous features) and on the treatment of categorical features (using $d_{vdm}$ or $d_O$ for HVDM-S and MDE, respectively), where in turn, HVDM-S seems superior. For higher missing rates (10, 20 and 30%), it becomes more difficult to determine which component is influencing the results the most. On one hand, MDE and HVDM-S are the top performing approaches for continuous and categorical datasets, respectively. On the other hand, they handle missing values in rather different ways.

One hypothesis is that the type of features comprised in the dataset (continuous or categorical) somewhat conditions the behaviour of distance functions. To analyse that relationship, heterogeneous datasets were divided into 3 groups: comprising mostly continuous features (CONT), comprising mostly categorical features (CAT) and comprising the same number of continuous and categorical features (EQUAL). Then, the performance of HVDM-S and MDE was compared in terms of percentage of wins and ties. Here, "wins" refer to the percentage of datasets where one distance function outperforms the other (HVDM-S outperforms MDE or vice-versa), whereas "ties" refer to situations where both distance functions achieve the same performance results. Table 13 presents the described analysis, showing the percentage of datasets for which each distance function outperforms the other and the percentage of ties, for each group.

**Table 12**
Differences between ranks for each comparison of distance functions for 5, 10, 20 and 30% and $k = 1$ (10 and 30% are shaded in grey). Significant differences are marked in bold.

*Continuous Datasets: 5 and 10%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | [a]HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | 0.13 | −0.08 | 0.52 | 0.36 | – | 0.93 | 0.31 |
| HEOM | −0.37 | – | −0.21 | 0.39 | 0.23 | – | 0.80 | 0.18 |
| HEOM-R | 0.28 | 0.65 | – | 0.60 | 0.44 | – | 1.01 | 0.39 |
| HVDM | −0.04 | 0.33 | −0.32 | – | −0.16 | – | 0.41 | −0.21 |
| HVDM-R | 0.28 | 0.65 | 0.00 | 0.32 | – | – | 0.57 | −0.05 |
| HVDM-S | – | – | – | – | – | – | – | – |
| MDE | −0.71 | −0.34 | −0.99 | −0.67 | −0.99 | – | – | −0.62 |
| SIMDIST | −0.07 | 0.30 | −0.35 | −0.03 | −0.35 | – | 0.64 | – |

*Continuous Datasets: 20 and 30%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | [a]HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | 0.39 | 0.41 | 0.34 | 0.44 | – | 1.05 | 0.80 |
| HEOM | 0.23 | – | 0.02 | −0.05 | 0.05 | – | 0.66 | 0.41 |
| HEOM-R | 0.96 | 0.73 | – | −0.07 | 0.03 | – | 0.64 | 0.39 |
| HVDM | 0.52 | 0.29 | −0.44 | – | 0.10 | – | 0.71 | 0.46 |
| HVDM-R | 0.58 | 0.35 | −0.38 | 0.06 | – | – | 0.61 | 0.36 |
| HVDM-S | – | – | – | – | – | – | – | – |
| MDE | −0.75 | −0.98 | **−1.71** | **−1.27** | **−1.33** | – | – | −0.25 |
| SIMDIST | 0.77 | 0.54 | −0.19 | 0.25 | 0.19 | – | **1.52** | – |

*Categorical Datasets: 5 and 10%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | 0.81 | 0.60 | 0.61 | 0.45 | 0.88 | 0.28 | 0.85 |
| HEOM | 0.52 | – | −0.21 | −0.20 | −0.36 | 0.07 | −0.53 | 0.04 |
| HEOM-R | 0.52 | 0.00 | – | 0.01 | −0.15 | 0.28 | −0.32 | 0.25 |
| HVDM | 0.65 | 0.13 | 0.13 | – | −0.16 | 0.27 | −0.33 | 0.24 |
| HVDM-R | 0.52 | 0.00 | 0.00 | −0.13 | – | 0.43 | −0.17 | 0.40 |
| HVDM-S | −0.64 | −1.16 | −1.16 | −1.29 | −1.16 | – | −0.60 | −0.03 |
| MDE | 0.12 | −0.40 | −0.40 | −0.53 | −0.40 | 0.76 | – | 0.57 |
| SIMDIST | 0.31 | −0.21 | −0.21 | −0.34 | −0.21 | 0.95 | 0.19 | – |

*Categorical Datasets: 20 and 30%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | 0.12 | −0.37 | −0.10 | −0.73 | 0.88 | 0.42 | −0.14 |
| HEOM | 0.46 | – | −0.49 | −0.22 | −0.85 | 0.76 | 0.30 | −0.26 |
| HEOM-R | −0.01 | −0.47 | – | 0.27 | −0.36 | 1.25 | 0.79 | 0.23 |
| HVDM | 0.53 | 0.07 | 0.54 | – | −0.63 | 0.98 | 0.52 | −0.04 |
| HVDM-R | 0.26 | −0.20 | 0.27 | −0.27 | – | **1.61** | 1.15 | 0.59 |
| HVDM-S | **−1.49** | **−1.95** | **−1.48** | **−2.02** | **−1.75** | – | −0.46 | −1.02 |
| MDE | −0.97 | −1.43 | −0.96 | **−1.50** | −1.23 | 0.52 | – | −0.56 |
| SIMDIST | 0.58 | 0.12 | 0.59 | 0.05 | 0.32 | **2.07** | **1.55** | – |

*Heterogeneous Datasets: 5 and 10%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | 0.10 | 0.58 | 0.20 | 0.24 | 0.35 | 0.64 | 0.21 |
| HEOM | −0.16 | – | 0.48 | 0.10 | 0.14 | 0.25 | 0.54 | 0.11 |
| HEOM-R | 0.35 | 0.51 | – | −0.38 | −0.34 | −0.23 | 0.06 | −0.37 |
| HVDM | −0.27 | −0.11 | −0.62 | – | 0.04 | 0.15 | 0.44 | 0.01 |
| HVDM-R | 0.61 | 0.77 | 0.26 | 0.88 | – | 0.11 | 0.40 | −0.03 |
| HVDM-S | −0.55 | −0.39 | −0.90 | −0.28 | −1.16 | – | 0.29 | −0.14 |
| MDE | −0.19 | −0.03 | −0.54 | 0.08 | −0.80 | 0.36 | – | −0.43 |
| SIMDIST | −0.27 | −0.11 | −0.62 | 0.00 | −0.88 | 0.28 | −0.08 | – |

*Heterogeneous Datasets: 20 and 30%*

| | BASELINE | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| BASELINE | – | −0.36 | −0.59 | 0.24 | −1.07 | 0.43 | 0.10 | −0.59 |
| HEOM | 0.02 | – | −0.23 | 0.60 | −0.71 | 0.79 | 0.46 | −0.23 |
| HEOM-R | −0.17 | −0.19 | – | 0.83 | −0.48 | 1.02 | 0.69 | 0.00 |
| HVDM | −0.67 | −0.69 | −0.50 | – | −1.31 | 0.19 | −0.14 | −0.83 |
| HVDM-R | −0.29 | −0.31 | −0.12 | 0.38 | – | **1.50** | 1.17 | 0.48 |
| HVDM-S | **−1.51** | **−1.53** | −1.34 | −0.84 | −1.22 | – | −0.33 | −1.02 |
| MDE | **−1.70** | **−1.72** | **−1.53** | −1.03 | −1.41 | −0.19 | – | −0.69 |
| SIMDIST | −0.64 | −0.66 | −0.47 | 0.03 | −0.35 | 0.87 | 1.06 | – |

[a]For continuous datasets, HVDM-S is equivalent to HVDM-R.

From the analysis of Table 13, several observations stand out. First, the percentage of ties when datasets are mostly continuous (CONT) is over the double than when datasets are mostly categorical (CAT), for MRs of 5 and 10%, indicating that one important difference between the two distance functions relies on the treatment of categorical values. For higher missing rates (20 and 30%), the difference between ties becomes less noticeable, suggesting that other factors may be at play, such as the treatment of missing data.

For intermediate missing rates (10 and 20%), the results obtained by HVDM-S and MDE follow the overall results shown in Table 8, with HVDM-S and MDE being superior for CAT and CONT datasets respectively. For 30%, CAT groups suffers an inversion of results (MDE becomes the best approach), whereas for CONT group results remain the same. This suggests that the major advantage of HVDM-S is on treatment of missing values in categorical features, when one value

**Table 13**
Performance comparison between HVDM-S and MDE, regarding the percentage of wins and ties ($k$ = 1), for each scenario (CAT, CONT and EQUAL).

| MR | CONT | | | CAT | | | EQUAL | | |
|---|---|---|---|---|---|---|---|---|---|
| | HVDM-S | MDE | TIE | HVDM-S | MDE | TIE | HVDM-S | MDE | TIE |
| ***5%*** | **46.7** | 33.3 | 20 | 42.9 | **47.6** | 9.5 | 7.1 | **71.4** | 21.4 |
| ***10%*** | 33.3 | **53.3** | 13.3 | **57.1** | 38.1 | 4.8 | **64.3** | 35.7 | 0 |
| ***20%*** | 33.3 | **53.3** | 13.3 | **47.6** | 42.9 | 9.5 | **50** | **50** | 0 |
| ***30%*** | 33.3 | **53.3** | 13.3 | 42.9 | **47.6** | 9.5 | 35.7 | **64.3** | 0 |

might be missing. When the MR is high, and it is more likely that both $x_{Aj}$ and $x_{Bj}$ values are missing, MDE seems to be superior.

The behaviour observed for the EQUAL group is consistent with this observation. For a MR of 5%, MDE performs exceptionally well, being superior to HVDM-S for 71.4% of datasets, but both distances perform equally well for 21.4% of datasets. As the MR increases, there are no more ties between methods. For a MR of 10%, there is a 64.3/35.7 difference between HVDM-S and MDE which may be due to the superiority of HVDM-S over categorical features. Nevertheless, for a MR of 20, differences decrease to 50/50 and lastly, MDE becomes the best approach for 30%.

Overall, HVDM-S shows a good behaviour for intermediate MRs (10 and 20%), whereas MDE performs well on extremes, especially for 30%. Aligned with the hypotheses that HVDM-S might not adequately address situations where both values are missing is the degradation in performance observed for heterogeneous datasets when comparing the results obtained by HVDM and HVDM-R (Table 8). For MRs greater than 5%, HVDM-R presents a degradation in performance when compared to HVDM. Note that the only difference between these approaches is that for HVDM-R, two missing values are considered equal, i.e., $d_j(x_{Aj}, x_{Bj}) = 0$. This effect was not so strongly observed for continuous or categorical data individually, but it seems to considerably affect the results on heterogeneous data. Such assignment seems to be impairing the classification performance and, given that HVDM-S follows the same procedure, this indicates that HVDM-S could be improved regarding this aspect.

Finally, another interesting observation regarding heterogeneous data is that HEOM, a popular solution for several heterogeneous domains, has not stood-out as the best approach for any missing rate. When compared to all the remaining distance functions, HEOM was only superior to HEOM-R and HVDM-R (10 and 20%) and SIMDIST (20%), lagging behind in all remaining scenarios (Table 12), which suggests that, although simple, it may not be the go-to approach, as suggested in several application papers (please refer to Table 2).

To ease the interpretation of results, Table 18 summarises the main conclusions derived for each group of datasets in what concerns the discussion on classification performance.

## 7. Impact on imputation quality

In this section, we analyse the imputation task directly and discuss the impact of the considered distance functions on the quality of imputation, focusing on their Predictive Accuracy (PAC), i.e., on their ability to reconstruct the original values in data (García-Laencina et al., 2010; Santos et al., 2017). PAC was assessed through the computation of the Normalised Mean Absolute Error (NMAE) and the percentage of matches, Matches (%), for continuous and categorical features, respectively.

Traditionally, the Mean Absolute Error (MAE) is computed as shown in Eq. (20), where $y_i$ and $\hat{y}_i$ represent the original value (ground truth) and imputed value and $n$ is the number of values that were missing in feature $x_j$. The MAE of a feature $x_j$ therefore represents an average of the deviation between the original and the imputed values. Naturally, the MAE is measured on the same scale as $x_j$, and since that dataset features may consider different scales, a normalisation (NMAE) is required to produce a final MAE measure for the entire dataset. In this work we considered a normalisation over $x_j$ values, i.e., $NMAE = \frac{MAE}{max(x_j) - min(x_j)}$. Accordingly, the final NMAE of a dataset is the average NMAE of all its features, where values closer to 0 indicate more accurate imputations.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid \tag{20}$$

The percentage of matches is given by Eq. (21), and indicates the proportion of categorical values that were exactly recreated (i.e., the imputed categorical value matches the original). In this case, accurate imputations should return a value closer to 100%.

$$Matches\ (\%) = \frac{100 \times \sum_{y_i = \hat{y}_i} 1}{n} \tag{21}$$

Tables 14, 15, 16 and 17 show the NMAE and Matches (%) results obtained with all distance functions, for $k$ values of 1, 3, 5 and 7, respectively. For all values of $k$, both NMAE and Matches (%) results are similar: for continuous datasets, SIMDIST is the top performing approach for all $k$, whereas for categorical and heterogeneous datasets, MDE is overall the best approach, with little exceptions where HVDM or SIMDIST outperform the remaining.

For continuous datasets, the NMAE is generally low, with a minimum value of 0.09 ($k$ = 5 and 7) and maximum of 0.156 ($k$ = 1), and there are no substantial differences between distance functions, even among different $k$ values.

For categorical datasets, however, MDE stands out when compared to the remaining approaches, achieving a percentage of exact matches around 60%, versus the 50%–56% obtained by the remaining ($k$ = 1). As the $k$ value increases, this difference becomes less noticeable, although MDE remains the top approach. An important note, however, is the lower imputation quality of HVDM-S on categorical data, when compared to the remaining distance functions: for all $k$ values, it obtains the lowest percentage of exact matches on categorical features. This observation confirms that classification and imputation are different tasks and therefore their evaluation should be carefully performed.

Nevertheless, the imputation quality results obtained by HVDM-S agree with its definition as described in Section 4 and discussed throughout the paper. On the one hand, since $d_{vdm}$ considers class targets when computing distances, HVDM-S (and generally all HVDM-like functions) considers some information regarding the classification task while computing distances, which grant it a major advantage for classification purposes. On the other hand, two values $x_Aj$ and $x_Bj$ are considered similar if their class distribution is similar which, while for classification purposes it may be beneficial, it may have undesirable consequences in terms of imputation quality. As an example, consider a dataset where $j$ represents a categorical feature, "Chest Pain", with possible values of "low", "moderate", "high" and "very high". If "high" and "very high" are often both associated with class "heart attack", imputing a missing value (whose original category is "high") as "high" or "very high" will not have consequences in terms of classification, but will not be translated into an exact match. This affects all HVDM-like functions (HVDM, HVDM-R, HVDM-S), which perform worse than the remaining approaches. For the particular case of HVDM-S, results are especially worse since missing values, considered as an extra category, are simply additional confounding factors in terms of imputation quality.

**Table 14**
NMAE and Matches (%) divided by groups and missing rates for $k = 1$ (best results are marked in bold).

| | MR | HEOM | HEOM-R | HVDM | HVDM-R | [a]HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| | **5%** | 0.107 ± 0.061 | 0.119 ± 0.058 | 0.107 ± 0.061 | 0.117 ± 0.058 | – | 0.110 ± 0.047 | **0.102** ± 0.061 |
| ***Continuous*** | **10%** | 0.115 ± 0.060 | 0.131 ± 0.056 | 0.114 ± 0.060 | 0.129 ± 0.058 | – | 0.112 ± 0.047 | **0.106** ± 0.061 |
| ***Datasets*** | **20%** | 0.128 ± 0.057 | 0.145 ± 0.054 | 0.127 ± 0.058 | 0.143 ± 0.056 | – | 0.118 ± 0.046 | **0.113** ± 0.060 |
| *NMAE* | **30%** | 0.139 ± 0.056 | 0.156 ± 0.054 | 0.138 ± 0.057 | 0.154 ± 0.056 | – | 0.123 ± 0.045 | **0.120** ± 0.059 |
| | **5%** | 55.4 ± 17.4 | 55.5 ± 17.4 | 54.6 ± 17.1 | 54.0 ± 16.7 | 50.0 ± 14.7 | **59.8** ± 16.0 | 55.7 ± 17.6 |
| ***Categorical*** | **10%** | 55.3 ± 17.0 | 55.3 ± 16.9 | 54.3 ± 16.7 | 53.6 ± 16.4 | 50.9 ± 15.5 | **59.8** ± 15.5 | 55.6 ± 17.2 |
| ***Datasets*** | **20%** | 54.5 ± 16.5 | 54.4 ± 16.3 | 53.4 ± 16.2 | 52.8 ± 16.0 | 51.5 ± 15.6 | **59.6** ± 15.2 | 54.7 ± 16.5 |
| *Matches (%)* | **30%** | 53.6 ± 15.9 | 53.4 ± 15.5 | 52.4 ± 15.7 | 51.9 ± 15.3 | 51.5 ± 15.8 | **59.2** ± 14.8 | 53.9 ± 16.0 |
| | **5%** | 0.202 ± 0.064 | 0.206 ± 0.065 | 0.190 ± 0.059 | 0.200 ± 0.059 | 0.194 ± 0.060 | **0.187** ± 0.060 | 0.201 ± 0.065 |
| | | 56.1 ± 16.4 | 55.9 ± 16.4 | 55.9 ± 14.9 | 55.0 ± 14.6 | 54.4 ± 14.8 | **58.7** ± 15.9 | 56.3 ± 16.4 |
| ***Heterogeneous*** | **10%** | 0.205 ± 0.062 | 0.211 ± 0.062 | 0.198 ± 0.060 | 0.210 ± 0.059 | 0.201 ± 0.059 | **0.190** ± 0.060 | 0.204 ± 0.063 |
| ***Datasets*** | | 55.9 ± 16.0 | 55.2 ± 15.7 | 55.5 ± 14.7 | 53.9 ± 14.5 | 54.5 ± 15.0 | **58.9** ± 15.2 | 56.3 ± 16.0 |
| *NMAE* | **20%** | 0.209 ± 0.060 | 0.218 ± 0.061 | 0.208 ± 0.060 | 0.222 ± 0.061 | 0.210 ± 0.059 | **0.193** ± 0.059 | 0.208 ± 0.062 |
| *Matches (%)* | | 55.3 ± 15.0 | 54.5 ± 14.8 | 54.1 ± 14.1 | 52.7 ± 14.2 | 53.8 ± 15.0 | **58.7** ± 14.8 | 56.1 ± 14.8 |
| | **30%** | 0.215 ± 0.059 | 0.224 ± 0.062 | 0.216 ± 0.060 | 0.228 ± 0.061 | 0.217 ± 0.060 | **0.196** ± 0.059 | 0.212 ± 0.061 |
| | | 54.7 ± 14.4 | 53.6 ± 14.3 | 53.2 ± 14.0 | 52.0 ± 14.1 | 52.9 ± 14.9 | **58.5** ± 14.5 | 55.4 ± 14.3 |

[a]For continuous datasets, HVDM-S is equivalent to HVDM-R.

**Table 15**
NMAE and Matches (%) divided by groups and missing rates ($k = 3$).

| | MR | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| | **5%** | 0.096 ± 0.050 | 0.105 ± 0.047 | 0.095 ± 0.050 | 0.103 ± 0.049 | – | 0.104 ± 0.046 | **0.091** ± 0.051 |
| ***Continuous*** | **10%** | 0.102 ± 0.050 | 0.114 ± 0.047 | 0.102 ± 0.050 | 0.112 ± 0.048 | – | 0.106 ± 0.045 | **0.094** ± 0.051 |
| ***Datasets*** | **20%** | 0.113 ± 0.048 | 0.127 ± 0.045 | 0.112 ± 0.049 | 0.125 ± 0.048 | – | 0.111 ± 0.045 | **0.100** ± 0.050 |
| *NMAE* | **30%** | 0.123 ± 0.047 | 0.136 ± 0.045 | 0.121 ± 0.049 | 0.134 ± 0.048 | – | 0.116 ± 0.044 | **0.106** ± 0.049 |
| | **5%** | 57.1 ± 17.9 | 57.2 ± 17.8 | 56.2 ± 17.5 | 55.6 ± 17.1 | 51.2 ± 15.3 | **60.4** ± 16.0 | 57.5 ± 18.0 |
| ***Categorical*** | **10%** | 57.2 ± 17.2 | 57.2 ± 17.1 | 56.2 ± 17.0 | 55.4 ± 16.5 | 52.1 ± 16.0 | **60.3** ± 15.6 | 57.5 ± 17.4 |
| ***Datasets*** | **20%** | 56.4 ± 16.8 | 56.3 ± 16.4 | 55.2 ± 16.5 | 54.6 ± 15.9 | 52.6 ± 16.0 | **60.0** ± 15.3 | 56.8 ± 16.8 |
| *Matches (%)* | **30%** | 55.7 ± 16.1 | 55.7 ± 15.7 | 54.3 ± 15.9 | 54.0 ± 15.3 | 52.6 ± 16.2 | **59.7** ± 14.9 | 55.9 ± 16.3 |
| | **5%** | 0.177 ± 0.056 | 0.179 ± 0.056 | **0.172** ± 0.055 | 0.178 ± 0.055 | 0.173 ± 0.054 | 0.175 ± 0.058 | 0.177 ± 0.057 |
| | | 58.3 ± 16.0 | 58.0 ± 15.9 | 57.4 ± 15.0 | 56.6 ± 14.9 | 55.8 ± 15.0 | **59.3** ± 15.4 | 58.4 ± 15.8 |
| ***Heterogeneous*** | **10%** | 0.179 ± 0.055 | 0.183 ± 0.054 | **0.176** ± 0.055 | 0.184 ± 0.054 | 0.178 ± 0.054 | 0.177 ± 0.058 | 0.179 ± 0.055 |
| ***Datasets*** | | 57.8 ± 15.7 | 57.2 ± 15.6 | 56.9 ± 14.7 | 55.6 ± 14.7 | 55.9 ± 14.9 | **59.6** ± 14.8 | 58.4 ± 15.7 |
| *NMAE* | **20%** | 0.184 ± 0.053 | 0.189 ± 0.054 | 0.184 ± 0.054 | 0.192 ± 0.054 | 0.183 ± 0.053 | **0.180** ± 0.057 | 0.182 ± 0.055 |
| *Matches (%)* | | 57.0 ± 15.2 | 56.4 ± 15.0 | 55.8 ± 14.5 | 54.8 ± 14.5 | 55.1 ± 15.1 | **59.2** ± 14.6 | 57.8 ± 14.9 |
| | **30%** | 0.188 ± 0.054 | 0.194 ± 0.054 | 0.190 ± 0.055 | 0.198 ± 0.054 | 0.189 ± 0.054 | **0.182** ± 0.057 | 0.186 ± 0.056 |
| | | 56.4 ± 14.8 | 55.5 ± 14.6 | 55.1 ± 14.4 | 54.0 ± 14.3 | 54.5 ± 14.9 | **59.0** ± 14.4 | 57.2 ± 14.7 |

**Table 16**
NMAE and Matches (%) divided by groups and missing rates ($k = 5$).

| | MR | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| | **5%** | 0.094 ± 0.048 | 0.102 ± 0.045 | 0.094 ± 0.048 | 0.100 ± 0.046 | – | 0.103 ± 0.045 | **0.090** ± 0.048 |
| ***Continuous*** | **10%** | 0.101 ± 0.047 | 0.111 ± 0.044 | 0.100 ± 0.048 | 0.109 ± 0.046 | – | 0.105 ± 0.045 | **0.092** ± 0.048 |
| ***Datasets*** | **20%** | 0.111 ± 0.046 | 0.123 ± 0.043 | 0.110 ± 0.047 | 0.122 ± 0.046 | – | 0.110 ± 0.044 | **0.098** ± 0.048 |
| *NMAE* | **30%** | 0.121 ± 0.045 | 0.132 ± 0.043 | 0.119 ± 0.047 | 0.131 ± 0.046 | – | 0.115 ± 0.043 | **0.104** ± 0.047 |
| | **5%** | 58.9 ± 17.5 | 58.9 ± 17.3 | 57.6 ± 17.2 | 57.1 ± 16.6 | 52.0 ± 15.8 | **60.7** ± 16.5 | 59.1 ± 17.7 |
| ***Categorical*** | **10%** | 58.5 ± 17.1 | 58.5 ± 17.0 | 57.4 ± 16.9 | 56.6 ± 16.2 | 52.8 ± 16.3 | **60.7** ± 15.9 | 58.8 ± 17.2 |
| ***Datasets*** | **20%** | 57.8 ± 16.5 | 57.5 ± 16.3 | 56.4 ± 16.2 | 55.6 ± 15.6 | 53.3 ± 16.2 | **60.5** ± 15.3 | 58.1 ± 16.7 |
| *Matches (%)* | **30%** | 57.0 ± 15.7 | 56.9 ± 15.3 | 55.6 ± 15.4 | 55.1 ± 14.8 | 53.2 ± 16.2 | **59.9** ± 15.0 | 57.3 ± 15.8 |
| | **5%** | 0.171 ± 0.054 | 0.173 ± 0.054 | **0.167** ± 0.054 | 0.172 ± 0.053 | 0.168 ± 0.053 | 0.171 ± 0.057 | 0.172 ± 0.055 |
| | | 59.1 ± 16.1 | 59.1 ± 16.0 | 58.4 ± 15.0 | 57.7 ± 14.9 | 56.8 ± 15.0 | 59.5 ± 15.5 | **59.8** ± 15.9 |
| ***Heterogeneous*** | **10%** | 0.174 ± 0.052 | 0.176 ± 0.052 | **0.171** ± 0.054 | 0.178 ± 0.052 | 0.172 ± 0.052 | 0.174 ± 0.056 | 0.174 ± 0.054 |
| ***Datasets*** | | 58.7 ± 15.6 | 58.3 ± 15.5 | 57.7 ± 14.6 | 56.7 ± 14.5 | 56.8 ± 14.7 | **59.8** ± 14.9 | 59.1 ± 15.5 |
| *NMAE* | **20%** | 0.179 ± 0.052 | 0.182 ± 0.051 | 0.179 ± 0.053 | 0.186 ± 0.052 | 0.178 ± 0.052 | **0.177** ± 0.056 | **0.177** ± 0.053 |
| *Matches (%)* | | 58.0 ± 15.2 | 57.6 ± 14.9 | 56.6 ± 14.5 | 55.9 ± 14.5 | 56.1 ± 14.9 | **59.5** ± 14.7 | 58.7 ± 15.0 |
| | **30%** | 0.182 ± 0.052 | 0.187 ± 0.052 | 0.184 ± 0.053 | 0.191 ± 0.052 | 0.183 ± 0.052 | **0.179** ± 0.056 | 0.180 ± 0.054 |
| | | 57.4 ± 14.7 | 56.6 ± 14.6 | 56.1 ± 14.4 | 55.2 ± 14.4 | 55.4 ± 14.7 | **59.2** ± 14.4 | 58.0 ± 14.8 |

For heterogeneous datasets, MDE remains the overall performing approach for all $k$, being the top performer for MRs of 20 and 30%, whereas for lower missing rates, HVDM and SIMDIST perform slightly better. However, NMAE values obtained with MDE are higher than the ones obtained for exclusively continuous data, whereas the percentage of matches remains consistently around 60%, as for categorical datasets. On the other hand, HVDM-S, although with slightly lower values of Matches (%) than the remaining distances, performs similarly to the remaining (especially as $k$ increases) contrary to what was observed for exclusively categorical datasets. Regarding NMAE values, HVDM-S also performs similarly to the remaining distance functions, often with slightly better results and improving as $k$ increases.

Overall, the experimental results suggest that, in terms of imputation quality, and considering all $k$ values, SIMDIST is the top performing approach for continuous data whereas MDE is the best approach for categorical and heterogeneous data. Nevertheless, it should be stated that, as previously discussed, imputation and classification are different tasks and both perspectives may be considered while evaluating imputation approaches. The disagreement on HVDM-S (for categorical datasets, HVDM-S performs the best in terms of classification results

**Table 17**
NMAE and Matches (%) divided by groups and missing rates ($k = 7$).

| | MR | HEOM | HEOM-R | HVDM | HVDM-R | HVDM-S | MDE | SIMDIST |
|---|---|---|---|---|---|---|---|---|
| | ***5%*** | 0.095 ± 0.047 | 0.102 ± 0.044 | 0.094 ± 0.047 | 0.100 ± 0.045 | – | 0.103 ± 0.044 | **0.090** ± 0.047 |
| ***Continuous*** | ***10%*** | 0.101 ± 0.046 | 0.111 ± 0.043 | 0.100 ± 0.047 | 0.109 ± 0.045 | – | 0.105 ± 0.044 | **0.093** ± 0.047 |
| ***Datasets*** | ***20%*** | 0.112 ± 0.045 | 0.122 ± 0.043 | 0.111 ± 0.046 | 0.121 ± 0.045 | – | 0.110 ± 0.043 | **0.098** ± 0.047 |
| *NMAE* | ***30%*** | 0.120 ± 0.044 | 0.131 ± 0.042 | 0.119 ± 0.046 | 0.130 ± 0.045 | – | 0.115 ± 0.043 | **0.103** ± 0.046 |
| | ***5%*** | 59.5 ± 17.0 | 59.5 ± 16.8 | 58.4 ± 16.7 | 58.1 ± 16.1 | 52.5 ± 15.9 | **61.0** ± 16.4 | 59.8 ± 17.2 |
| ***Categorical*** | ***10%*** | 59.4 ± 16.8 | 59.2 ± 16.7 | 58.2 ± 16.5 | 57.3 ± 15.9 | 53.3 ± 16.5 | **60.9** ± 15.9 | 59.6 ± 16.9 |
| ***Datasets*** | ***20%*** | 58.7 ± 16.1 | 58.5 ± 15.9 | 57.2 ± 15.8 | 56.5 ± 15.1 | 53.7 ± 16.4 | **60.6** ± 15.3 | 59.0 ± 16.3 |
| *Matches (%)* | ***30%*** | 57.7 ± 15.5 | 57.6 ± 15.1 | 56.2 ± 15.1 | 55.7 ± 14.5 | 53.7 ± 16.2 | **60.1** ± 14.9 | 58.1 ± 15.7 |
| | ***5%*** | 0.169 ± 0.052 | 0.170 ± 0.052 | **0.164** ± 0.053 | 0.169 ± 0.052 | 0.165 ± 0.052 | 0.170 ± 0.056 | 0.170 ± 0.053 |
| | | 59.4 ± 15.9 | 59.4 ± 15.9 | 58.7 ± 14.7 | 58.3 ± 14.7 | 57.3 ± 14.8 | 59.9 ± 15.4 | **60.0** ± 15.7 |
| ***Heterogeneous*** | ***10%*** | 0.172 ± 0.051 | 0.174 ± 0.051 | **0.169** ± 0.053 | 0.175 ± 0.051 | 0.170 ± 0.051 | 0.172 ± 0.056 | 0.172 ± 0.053 |
| ***Datasets*** | | 59.2 ± 15.3 | 59.0 ± 15.2 | 58.1 ± 14.3 | 57.4 ± 14.4 | 57.2 ± 14.6 | **60.1** ± 14.8 | 59.9 ± 15.2 |
| *NMAE* | ***20%*** | 0.176 ± 0.051 | 0.179 ± 0.050 | 0.177 ± 0.052 | 0.183 ± 0.052 | 0.176 ± 0.051 | **0.175** ± 0.055 | **0.175** ± 0.053 |
| *Matches (%)* | | 58.4 ± 15.1 | 58.1 ± 14.8 | 57.3 ± 14.3 | 56.7 ± 14.2 | 56.7 ± 14.6 | **59.6** ± 14.7 | 59.2 ± 15.0 |
| | ***30%*** | 0.180 ± 0.051 | 0.184 ± 0.051 | 0.182 ± 0.052 | 0.188 ± 0.052 | 0.181 ± 0.051 | **0.178** ± 0.055 | **0.178** ± 0.053 |
| | | 58.1 ± 14.5 | 57.2 ± 14.6 | 56.7 ± 14.3 | 55.8 ± 14.3 | 56.0 ± 14.4 | **59.3** ± 14.4 | 58.6 ± 14.7 |

**Table 18**
Summary of conclusions on continuous, categorical and heterogeneous datasets regarding both classification and imputation quality.

| | Classification performance | Imputation quality |
|---|---|---|
| ***Continuous Datasets*** | • Overall, MDE outperforms the remaining distance functions for all MRs ($k = 1$ and 3).<br>• For higher values of $k$ differences become negligible.<br>• Considering two values as being equal or a maximal distance if one value is missing seems prejudicial.<br>• Distinguishing situations where only one or both values are missing seems beneficial. | • Considering all $k$ values and MRs, SIMDIST is the top performing approach. |
| ***Categorical Datasets*** | • For all $k$, HVDM-S is the overall top performing approach across all MRs.<br>• Considering the distribution of missing data in each class seems beneficial. | • Considering all $k$ values and MRs, MDE is the top performing approach.<br>• For all $k$ values and MRs, HVDM-S performs worse than the remaining distance functions. |
| ***Heterogeneous Datasets*** | • For $k = 1$, MDE and HVDM-S are the top performing approaches.<br>• HVDM-S handles missing data in categorical features better when one value is missing. For higher MRs, MDE is superior.<br>• For higher $k$ values, HVDM-S remains the top performer, for intermediate MRs. For 5% or 30% MRs, there is no consensus. | • For $k = 1$, MDE is the best approach.<br>• For higher values of $k$, MDE is the best approach for MRs of 20 and 30%, although HVDM and SIMDIST perform slightly better in some scenarios.<br>• Regarding Matches (%), HVDM-S performs similarly to the remaining distance functions, especially as $k$ increases. |

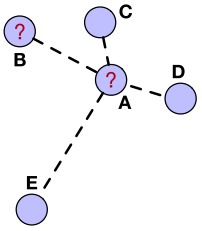



**Fig. 3.** kNN imputation schema for a $k = 3$ neighbourhood: patterns with missing values in the feature of interest (such as $\mathbf{x}_B$) will be disregarded for imputation.

while being the worst approach in terms of imputation quality) suggests that different metrics assess different aspects (in this case, the performance on different tasks) and that evaluation should be conducted on the most relevant aspects for the domain, while considering appropriate measures. The top imputation approach in terms of classification performance is not necessarily the top approach in terms of imputation quality, and it is important to determine which is more critical for the problem at hand.

Finally, the NMAE and Matches (%) results obtained for different values of $k$, allow us to draw some conclusions regarding the weighting strategy used for data imputation. As explained in Section 5, the imputation is weighted according to the distance of each neighbour on continuous features whereas for categorical the mode is used instead. In terms of Matches (%), it seems that an increase of $k$ slightly improves the results (the mode is computed considering a higher number of neighbours). In terms of NMAE, although results do not considerably change for continuous datasets, they increasingly improve for heterogeneous datasets, as $k$ increases, meaning that although the neighbourhood is increasing, which may typically lead to a distortion on the imputed values as more neighbours are being considered, the weighting strategy presented in Eq. (19) is able to take advantage of a broader concept surrounding the missing pattern, while also minimising such distortion, by given a higher weight to closer neighbours. This is especially relevant for missing data imputation, as the neighbours that can act as "donors" for imputation are dependent on the availability of values on a given feature. To illustrate this idea, please refer to Fig. 3. In a multivariate MCAR scenario, all values from all features (and patterns) are equally likely to be missing. Thus, consider pattern $\mathbf{x}_A$, whose value for a given feature $j = 1$ ($f_1$) for instance, is missing (denoted by "?"). If we considered a $k = 3$ neighbourhood, then patterns $\mathbf{x}_B$, $\mathbf{x}_C$ and $\mathbf{x}_D$ should be considered for imputation. However, it happens that pattern $\mathbf{x}_B$ is also missing a value on $f_1$. Considering distances that handle missing data allows to consider $\mathbf{x}_B$, $\mathbf{x}_C$ and $\mathbf{x}_D$ as donors even if they have some missing values, i.e, they could serve as donors for $\mathbf{x}_A$ for $f_2$, for instance. However, donors must have observed values on the feature considered for imputation. In this case, as $\mathbf{x}_B$ is also missing a value in $f_1$, the next closest neighbour needs to be considered, $\mathbf{x}_E$, although it may be farther than the remaining neighbours. This may not have a great impact in terms of classification performance (ultimately, all points could belong to the same class), but it may be provoke a distortion in terms of imputation quality (especially NMAE). However, weighting donors based on their distance to $\mathbf{x}_A$ would make the contribution of $\mathbf{x}_E$ mainly negligible.

Taken together, these differences found between both tasks (classification and imputation) also suggest something important: that for

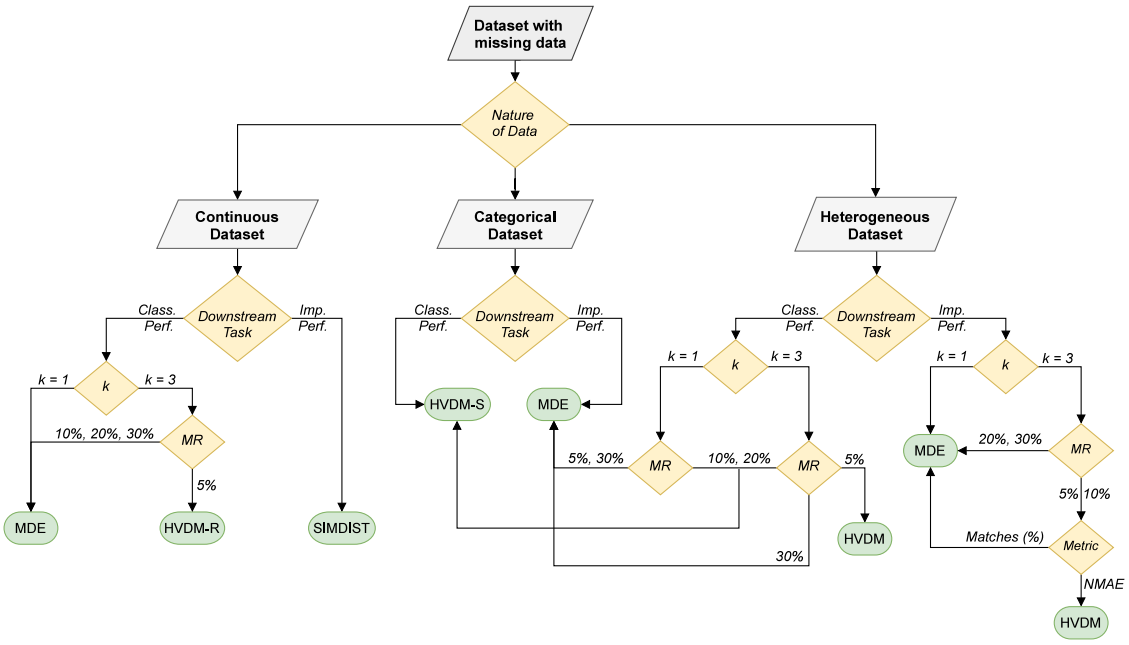



**Fig. 4.** Summary of best practices for researchers regarding kNNI imputation ($k = 3, 5$), considering datasets with different characteristics (nature of features and missing rates), as well as distinct downstream tasks (classification and imputation).

classification purposes, the chosen distance function may significantly impact the obtained results, whereas regarding imputation purposes, although the distance function plays an important role, the $k$ parametrisation and weighting scheme used for imputation are also potentially impactful for superior results.

The main conclusions on imputation quality are also depicted on Table 18 considering each group of datasets individually.

## 8. Conclusions and future work

In this work, we performed a comparison of several heterogeneous distance functions that handle missing values across a benchmark of 150 datasets with different characteristics (continuous, categorical and heterogeneous datasets). Whereas Sections 6 and 7 provide a detailed analysis on classification performance and imputation quality, respectively, herein we focus on summarising the main conclusion of the work, while also elaborating on possible future research directions. To that end, Table 18 presents a summary of the main conclusions obtained for both classification performance and imputation quality, while particularly focusing on the reflections discussed throughout the paper regarding continuous, categorical, and heterogeneous datasets.

In turn, Fig. 4 summarises the main recommendations for researchers approaching domain affected by missing data, where kNNI is a sensible solution of choice. Recommendations regarding the most suitable distance functions for kNNI attend to the desired downstream task (classification or imputation) and to the characteristics of the dataset (nature of features and missing rate). Values of $k = 3, 5$ are chosen as the most representative of a local approximation of imputation. Lower values maintain the variability of data in the domain and are common in real-world application domains (please refer to Table 2).

We conclude the paper by describing the main lessons learned from the experimental data, and presenting promising lines for future research:

- For all $k$ values and missing rates, learning classifiers from imputed data is preferred to classification with missing data, as kNN imputation generally outperforms the BASELINE results. For some scenarios ($k = 1$ and MR of 30%) building CART models with missing data might be preferred to imputing with some distance functions, though not preferred over MDE or HVDM-S;
- As the missing rates increases, differences in classification performance between distance functions become more significant, especially for $k = 1$ and 3, showing that missing data has a considerable impact on classification performance. For higher values of $k$, differences are more subtle;
- In terms of classification performance, MDE and HVDM-S are the top two performing approaches: MDE stands out as the best approach for continuous datasets ($k = 1$ and 3), while for categorical datasets, HVDM-S frequently outperforms all others (for all $k$). For heterogeneous datasets, both MDE and HVDM-S figure consistently among the best approaches, for all $k$;
- For continuous datasets, the major difference between distance functions consists in the treatment of missing data. Rather than defining similarities according to the availability of $x_{Aj}$ or $x_{Bj}$ directly, the best approach considers the average similarities among observed values in data. Also, distinguishing situations where one value is missing or two values are missing seems a suitable approach;
- For categorical datasets, the ability of HVDM-S to use information on the distribution of missing values by class seems to be the key to the good performance results achieved;
- For heterogeneous datasets, an improved distance function could combine the properties of MDE and HVDM-S. MDE provides a better treatment for continuous features, whereas HVDM-S is superior for categorical features. Regarding categorical features, when one value is missing, the computation used by HVDM-S on categorical features seems to be the most suitable, whereas when both values are missing, MDE seems to perform better (although HVDM-S could be improved by readjusting this comparison);
- Also regarding classification performance, we argue that HEOM, although widely used across several heterogeneous domains may not be the go-to approach, as others have shown to be more beneficial;

- Regarding imputation quality and considering all $k$ values, SIMDIST is the top performing approach for continuous data, whereas MDE seems better for categorical and heterogeneous data;
- Of note are also the results obtained by HVDM-S for categorical data. While it obtains the highest classification results, it performs poorly in terms of imputation quality. This suggested that considering the class of patterns while performing imputation helps to model the classification task, although it does not benefit the imputation task per se;
- Differences found among the analysis of classification versus imputation quality suggest that, for classification performance, the choice of distance function is the most determining aspect to obtain superior results (especially for categorical and heterogeneous datasets). For imputation quality, the $k$-parametrisation and weighting scheme seem also important to obtain improved results;
- Classification and imputation are different tasks and their evaluation should be performed accordingly, using adequate metrics. It is not guaranteed that the top approach in terms of classification performs the best in terms of imputation quality. A suitable imputation approach should consider both (to this regard, MDE obtains robust results); however, both the objective and conditions of the study (missing rate, characteristics of data) should be taken into account to perform an informed decision on the best imputation approach.

Future work is focused on further studying heterogeneous datasets under extended experimental conditions, e.g., generating missing values only on categorical or continuous features. Preliminary results have shown that HVDM-S remains a suitable approach for most scenarios and shows a particularly good behaviour for more complex datasets (Santos et al., 2020a). A more in-depth analysis of categorical features (and ratio of categorical to continuous features) is also an interesting topic for further research: we expect to find different results depending on the number of multi-valued nominal attributes and number of categorical/continuous features. Other promising directions would be the development of a novel distance function based on the advantages of the distances studied in this work and finally, the investigation of other missing data mechanisms (e.g., MAR), missing rates (>30%) and strategies of weighting features differently (e.g., based on their mutual information or discriminative power).

## CRediT authorship contribution statement

**Miriam Seoane Santos:** Conceptualization, Methodology, Literature search, Investigation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Pedro Henriques Abreu:** Conceptualization, Validation, Writing – review & editing, Supervision. **Alberto Fernández:** Validation, Writing – review & editing. **Julián Luengo:** Validation, Writing – review & editing. **João Santos:** Writing – review & editing.

## Declaration of competing interest

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.

## Acknowledgements

This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., Portugal, within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020. It is also partially supported by Andalusian frontier regional project A-TIC-434-UGR20 and by the Spanish Ministry of Science and Technology under project PID2020-119478GB-I00 including European Regional Development Funds. The work is further supported by the FCT, Portugal Research Grant SFRH/BD/138749/2018.

## References

AbdAllah, L., Shimshoni, I., 2014. Mean shift clustering algorithm for data with missing values. In: International Conference on Data Warehousing and Knowledge Discovery. Springer, pp. 426–438.

AbdAllah, L., Shimshoni, I., 2016. K-means over incomplete datasets using mean Euclidean distance. In: Machine Learning and Data Mining in Pattern Recognition. Springer, pp. 113–127.

Abnane, I., Hosni, M., Idri, A., Abran, A., 2019. Analogy software effort estimation using ensemble KNN imputation. In: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, pp. 228–235.

Abreu, P.H., Santos, M.S., Abreu, M.H., Andrade, B., Silva, D.C., 2016. Predicting breast cancer recurrence using machine learning techniques: a systematic review. ACM Comput. Surv. 49 (3), 1–40.

Abu Alfeilat, H.A., Hassanat, A.B., Lasassmeh, O., Tarawneh, A.S., Alhasanat, M.B., Eyal Salman, H.S., Prasath, V.S., 2019. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. Big Data 7 (4), 221–248.

Al-Helali, B., Chen, Q., Xue, B., Zhang, M., 2021. A new imputation method based on genetic programming and weighted KNN for symbolic regression with incomplete data. Soft Comput. 25 (8), 5993–6012.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2011. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework.. J. Mult.-Valued Logic Soft Comput. 17.

Ali, N., Neagu, D., Trundle, P., 2019. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. SN Appl. Sci. 1 (12), 1–15.

Amorim, J.P., Domingues, I., Abreu, P.H., Santos, J., 2018. Interpreting deep learning models for ordinal problems.. In: ESANN. pp. 373–378.

de Andrade Silva, J., Hruschka, E.R., 2013. An experimental study on the use of nearest neighbor-based imputation algorithms for classification tasks. Data Knowl. Eng. 84, 47–58.

Anwar, N., Jones, G., Ganesh, S., 2014. Measurement of data complexity for classification problems with unbalanced data. Statist. Anal. Data Mining ASA Data Sci. J. 7 (3), 194–211.

Barigou, F., 2018. Impact of instance selection on kNN-based text categorization. J. Inform. Process. Syst. 14 (2), 418–434.

Batista, G., Monard, M.C., 2001. A study of K-nearest neighbour as a model-based method to treat missing data. In: Proceedings of the Argentine Symposium on Artificial Intelligence, Vol. 30, pp. 1–9.

Batista, G., Monard, M.C., 2002. A study of K-nearest neighbour as an imputation method. HIS 87 (251–260), 48.

Batista, G., Monard, M.C., 2003. An analysis of four missing data treatment methods for supervised learning. Appl. Artif. Intell. 17 (5–6), 519–533.

Batista, G., Silva, D.F., 2009. How k-nearest neighbor parameters affect its performance. In: Argentine Symposium on Artificial Intelligence. pp. 1–12.

Belanche Muñoz, L.A., Hernández González, J., 2012. Similarity networks for heterogeneous data. In: ESANN 2012. pp. 215–220.

Beretta, L., Santaniello, A., 2016. Nearest neighbor imputation algorithms: a critical evaluation. BMC Med. Inform. Decision Making 16 (3), 74.

Bertsimas, D., Pawlowski, C., Zhuo, Y.D., 2017. From predictive methods to missing data imputation: an optimization approach. J. Mach. Learn. Res. 18 (1), 7133–7171.

Borowska, K., Stepaniuk, J., 2016. Imbalanced data classification: A novel re-sampling approach combining versatile improved SMOTE and rough sets. In: IFIP International Conference on Computer Information Systems and Industrial Management. Springer, pp. 31–42.

Brás, L.P., Menezes, J.C., 2007. Improving cluster-based missing value estimation of DNA microarray data. Biomolecular Engineering 24 (2), 273–282.

Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16, 321–357.

Cheng, C.-H., Chan, C.-P., Sheu, Y.-J., 2019. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. Eng. Appl. Artif. Intell. 81, 283–299.

Cho, S., Hong, H., Ha, B.-C., 2010. A hybrid approach based on the combination of variable selection using decision trees and case-based reasoning using the Mahalanobis distance: For bankruptcy prediction. Expert Syst. Appl. 37 (4), 3482–3488.

Choudhury, A., Kosorok, M.R., 2020. Missing data imputation for classification problems. arXiv preprint arXiv:2002.10709.

Das, S., Datta, S., Chaudhuri, B.B., 2018. Handling data irregularities in classification: Foundations, trends, and future challenges. Pattern Recognit. 81, 674–693.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7 (Jan), 1–30.

Deng, Z., Zhu, X., Cheng, D., Zong, M., Zhang, S., 2016. Efficient kNN classification algorithm for big data. Neurocomputing 195, 143–148.

Dua, D., Graff, C., 2019. UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml.
Dudani, S.A., 1976. The distance-weighted k-nearest-neighbor rule. IEEE Trans. Syst. Man Cybern. SMC-6 (4), 325–327.
Eirola, E., Doquire, G., Verleysen, M., Lendasse, A., 2013. Distance estimation in numerical data sets with missing values. Inform. Sci. 240, 115–128.
Elnaggar, R., Chakrabarty, K., 2018. Machine learning for hardware security: opportunities and risks. J. Electron. Test. 34 (2), 183–201.
Ertuğrul, O.F., 2019. A novel distance metric based on differential evolution. Arab. J. Sci. Eng. 44 (11), 9641–9651.
Farhangfar, A., Kurgan, L., Dy, J., 2008. Impact of imputation of missing values on classification error for discrete data. Pattern Recognit. 41 (12), 3692–3705.
Fouad, K.M., Ismail, M.M., Azar, A.T., Arafa, M.M., 2021. Advanced methods for missing values imputation based on similarity learning. PeerJ Comput. Sci. 7, e619.
Fu, Y., He, H.S., Hawbaker, T.J., Henne, P.D., Zhu, Z., Larsen, D.R., 2019. Evaluating k-nearest neighbor (kNN) imputation models for species-level aboveground forest biomass mapping in northeast China. Remote Sens. 11 (17), 2005.
Garbasevschi, O.M., Schmiedt, J.E., Verma, T., Lefter, I., Altes, W.K.K., Droin, A., Schiricke, B., Wurm, M., 2021. Spatial factors influencing building age prediction and implications for urban residential energy modelling. Comput. Environ. Urban Syst. 88, 101637.
García-Laencina, P., Abreu, P.H., Abreu, M.H., Afonoso, N., 2015. Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. Comput. Biol. Med. 59, 125–133.
García-Laencina, P., Sancho-Gómez, J.-L., Figueiras-Vidal, A., 2010. Pattern classification with missing data: a review. Neural Comput. Appl. 19 (2), 263–282.
García-Laencina, P.J., Sancho-Gómez, J.-L., Figueiras-Vidal, A.R., Verleysen, M., 2009. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. Neurocomputing 72 (7–9), 1483–1493.
Gerhana, Y.A., Atmadja, A.R., Zulfikar, W.B., Ashanti, N., 2017. The implementation of K-nearest neighbor algorithm in case-based reasoning model for forming automatic answer identity and searching answer similarity of algorithm case. In: 2017 5th International Conference on Cyber and IT Service Management (CITSM). IEEE, pp. 1–5.
Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., Yang, H., 2019a. A generalized mean distance-based k-nearest neighbor classifier. Expert Syst. Appl. 115, 356–372.
Gou, J., Qiu, W., Yi, Z., Xu, Y., Mao, Q., Zhan, Y., 2019b. A local mean representation-based K-nearest neighbor classifier. ACM Trans. Intell. Syst. Technol. (TIST) 10 (3), 1–25.
Harikumar, S., Surya, P., 2015. K-medoid clustering for heterogeneous datasets. Procedia Comput. Sci. 70, 226–237.
Hegde, J., Rokseth, B., 2020. Applications of machine learning methods for engineering risk assessment–A review. Saf. Sci. 122, 104492.
Hruschka, E.R., Hruschka, E.R., Ebecken, N.F., 2004. Towards efficient imputation by nearest-neighbors: A clustering-based approach. In: Australasian Joint Conference on Artificial Intelligence. Springer, pp. 513–525.
Hu, L.-Y., Huang, M.-W., Ke, S.-W., Tsai, C.-F., 2016. The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus 5 (1), 1304.
Huang, J., Keung, J.W., Sarro, F., Li, Y.-F., Yu, Y.-T., Chan, W., Sun, H., 2017. Cross-validation based k nearest neighbor imputation for software quality datasets: an empirical study. J. Syst. Softw. 132, 226–252.
Huang, C.-C., Lee, H.-M., 2004. A grey-based nearest neighbor approach for missing attribute value prediction. Appl. Intell. 20 (3), 239–252.
Huang, M.-W., Lin, W.-C., Chen, C.-W., Ke, S.-W., Tsai, C.-F., Eberle, W., 2016. Data preprocessing issues for incomplete medical datasets. Expert Syst. 33 (5), 432–438.
Jadhav, A., Pramod, D., Ramanathan, K., 2019. Comparison of performance of data imputation methods for numeric dataset. Appl. Artif. Intell. 33 (10), 913–933.
Jäger, S., Allhorn, A., Bießmann, F., 2021. A benchmark for data imputation methods. Front. Big Data 48.
Jerez, J., Molina, I., García-Laencina, P., Alba, E., Ribelles, N., Martín, M., Franco, L., 2010. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artif. Intell. Med. 50 (2), 105–115.
Jiang, C., Yang, Z., 2015. CKNNI: an improved knn-based missing value handling technique. In: International Conference on Intelligent Computing. Springer, pp. 441–452.
Juhola, M., Laurikkala, J., 2007. On metricity of two heterogeneous measures in the presence of missing values. Artif. Intell. Rev. 28 (2), 163–178.
Kaggle, https://www.kaggle.com, Accessed: 2021-09-25.
Kalra, M., Lal, N., Qamar, S., 2018. K-mean clustering algorithm approach for data mining of heterogeneous data. In: Information and Communication Technology for Sustainable Development. Springer, pp. 61–70.
Kim, K.-Y., Kim, B.-J., Yi, G.-S., 2004. Reuse of imputed data in microarray analysis increases imputation efficiency. BMC Bioinformatics 5 (1), 1–9.
Kiriş, S.B., Özcan, T., 2020. Metaheuristics approaches to solve the employee bus routing problem with clustering-based bus stop selection. In: Artificial Intelligence and Machine Learning Applications in Civil, Mechanical, and Industrial Engineering. IGI Global, pp. 217–239.
Kobak, D., Berens, P., 2019. The art of using t-SNE for single-cell transcriptomics. Nature Commun. 10 (1), 1–14.
Kong, J., Kowalczyk, W., Menzel, S., Bäck, T., 2020. Improving imbalanced classification by anomaly detection. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 512–523.
Leyva, E., González, A., Perez, R., 2014. A set of complexity measures designed for applying meta-learning to instance selection. IEEE Trans. Knowl. Data Eng. 27 (2), 354–367.
Li, W., Cerise, J.E., Yang, Y., Han, H., 2017. Application of t-SNE to human genetic data. J. Bioinform. Comput. Biol. 15 (04), 1750017.
Li, H., Huang, H.-B., Sun, J., Lin, C., 2010. On sensitivity of case-based reasoning to optimal feature subsets in business failure prediction. Expert Syst. Appl. 37 (7), 4811–4821.
Lin, W.-Y., Hu, Y.-H., Tsai, C.-F., 2011. Machine learning in financial crisis prediction: a survey. IEEE Trans. Syst. Man Cybern. Part C 42 (4), 421–436.
Lin, W.-C., Tsai, C.-F., 2020. Missing value imputation: a review and analysis of the literature (2006–2017). Artif. Intell. Rev. 53 (2), 1487–1509.
Lorena, A.C., Garcia, L.P., Lehmann, J., Souto, M.C., Ho, T.K., 2019. How complex is your classification problem? A survey on measuring classification complexity. ACM Comput. Surv. 52 (5), 1–34.
Luengo, J., García, S., Herrera, F., 2010. A study on the use of imputation methods for experimentation with radial basis function network classifiers handling missing attribute values: The good synergy between rbfns and eventcovering method. Neural Netw. 23 (3), 406–418.
Luengo, J., García, S., Herrera, F., 2012. On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl. Inf. Syst. 32 (1), 77–108.
Lumijärvi, J., Laurikkala, J., Juhola, M., 2004. A comparison of different heterogeneous proximity functions and Euclidean distance. Stud. Health Technol. Inform. 107 (Pt 2), 1362–1366.
Mahajan, V., Misra, R., Mahajan, R., 2015. Review of data mining techniques for churn prediction in telecom. J. Inform. Organizational Sci. 39 (2), 183–197.
Mahin, M., Islam, M.J., Debnath, B.C., Khatun, A., 2019. Tuning distance metrics and k to find sub-categories of minority class from imbalance data using k nearest neighbours. In: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, pp. 1–6.
Mahin, M., Islam, M.J., Khatun, A., Debnath, B.C., 2018. A comparative study of distance metric learning to find sub-categories of minority class from imbalance data. In: 2018 International Conference on Innovation in Engineering and Technology (ICIET). IEEE, pp. 1–6.
Maillo, J., Ramírez, S., Triguero, I., Herrera, F., 2017. KNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowl.-Based Syst. 117, 3–15.
Malhotra, R., 2015. A systematic review of machine learning techniques for software fault prediction. Appl. Soft Comput. 27, 504–518.
Mbow, M., Koide, H., Sakurai, K., 2021. An intrusion detection system for imbalanced dataset based on deep learning. In: 2021 Ninth International Symposium on Computing and Networking (CANDAR). IEEE, pp. 38–47.
Napierala, K., Stefanowski, J., 2016. Types of minority class examples and their influence on learning classifiers from imbalanced data. J. Intell. Inf. Syst. 46 (3), 563–597.
Napierała, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples. In: International Conference on Rough Sets and Current Trends in Computing. Springer, pp. 158–167.
Negri, S., Belanche, L., 2001. Heterogeneous kohonen networks. In: International Work-Conference on Artificial Neural Networks. Springer, pp. 243–252.
Nekooeimehr, I., Lai-Yuen, S.K., 2016. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. Expert Syst. Appl. 46, 405–416.
Nnamoko, N., Korkontzelos, I., 2020. Efficient treatment of outliers and class imbalance for diabetes prediction. Artif. Intell. Med. 104, 101815.
Nunes, G.H., Martins, G.O., Forster, C.H., Lorena, A.C., 2021. Using instance hardness measures in curriculum learning. In: Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional. SBC, pp. 177–188.
Oh, S., 2011. A new dataset evaluation method based on category overlap. Comput. Biol. Med. 41 (2), 115–122.
Okafor, N.U., Delaney, D.T., 2021. Missing data imputation on IoT sensor networks: Implications for on-site sensor calibration. IEEE Sens. J. 21 (20), 22833–22845.
Oliveira, F.H.M., Machado, A.R., Andrade, A.O., 2018. On the use of t-distributed stochastic neighbor embedding for data visualization and classification of individuals with Parkinson's disease. Comput. Math. Methods Med. 2018.
Pan, R., Yang, T., Cao, J., Lu, K., Zhang, Z., 2015. Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. Appl. Intell. 43 (3), 614–632.
Parameswaran, S., Weinberger, K.Q., 2010. Large margin multi-task metric learning. In: Advances in Neural Information Processing Systems. pp. 1867–1875.
Park, J.H., Shen, H., Cao, J.-n., Xhafa, F., Jeong, Y.-S., 2015. Advanced modeling and services based mathematics for ubiquitous computing.
Pereira, R.C., Abreu, P.H., Rodrigues, P.P., 2020. Vae-bridge: Variational autoencoder filter for bayesian ridge imputation of missing data. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–7.
Poulos, J., Valle, R., 2018. Missing data imputation for supervised learning. Appl. Artif. Intell. 32 (2), 186–196.

Prasatha, V., Alfeilate, H.A.A., Hassanate, A., Lasassmehe, O., Tarawnehf, A.S., Alhasanatg, M.B., Salmane, H.S.E., 2019. Effects of distance measure choice on knn classifier performance - a review. p. 39, arXiv Preprint arXiv:1708.04321v3.

Rastin, N., Jahromi, M.Z., Taheri, M., 2021. A generalized weighted distance k-nearest neighbor for multi-label problems. Pattern Recognit. 114, 107526.

Ribeiro, F., Gradvohl, A., 2021. Machine learning techniques applied to solar flares forecasting. Astron. Comput. 35, 100468.

Rubin, D.B., 1976. Inference and missing data. Biometrika 63 (3), 581–592.

Saeed, N., Nam, H., Haq, M.I.U., Muhammad Saqib, D.B., 2018. A survey on multidimensional scaling. ACM Comput. Surv. 51 (3), 1–25.

Santos, M.S., Abreu, P.H., García-Laencina, P., Simão, A., Carvalho, A., 2015. A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. J. Biomed. Inform. 58, 49–59.

Santos, M.S., Abreu, P.H., Wilk, S., Santos, J., 2020a. Assessing the impact of distance functions on K-nearest neighbours imputation of biomedical datasets. In: International Conference on Artificial Intelligence in Medicine. Springer, pp. 486–496.

Santos, M.S., Abreu, P.H., Wilk, S., Santos, J., 2020b. How distance metrics influence missing data imputation with k-nearest neighbours. Pattern Recognit. Lett..

Santos, M.S., Pereira, R.C., Costa, A.F., Soares, J.P., Santos, J., Abreu, P.H., 2019. Generating synthetic missing data: A review by missing mechanism. IEEE Access 7, 11651–11667.

Santos, M.S., Soares, J.P., Abreu, P.H., Araújo, H., Santos, J., 2017. Influence of data distribution in missing data imputation. In: International Conference on Artificial Intelligence in Medicine in Europe. Springer, pp. 285–294.

Santos, M.S., Soares, J.P., Abreu, P.H., Araújo, H., Santos, J., 2018. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. IEEE Comput. Intell. Mag. 13 (4), 59–76.

Sarbazi-Azad, S., Abadeh, M.S., Mowlaei, M.E., 2020. Using data complexity measures and an evolutionary cultural algorithm for gene selection in microarray data. Soft Comput. Lett. 100007.

Smith, M.R., Martinez, T., Giraud-Carrier, C., 2014. An instance level analysis of data complexity. Mach. Learn. 95 (2), 225–256.

Sousa, L.R., Miranda, T., Sousa, R.L., Tinoco, J., 2017. The use of data mining techniques in rockburst risk assessment. Engineering 3 (4), 552–558.

Suárez, J.L., García, S., Herrera, F., 2021. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. Neurocomputing 425, 300–322.

Sun, B., Ma, L., Cheng, W., Wen, W., Goswami, P., Bai, G., 2017. An improved k-nearest neighbours method for traffic time series imputation. In: 2017 Chinese Automation Congress (CAC). IEEE, pp. 7346–7351.

Tabassian, M., Alessandrini, M., Jasaityte, R., De Marchi, L., Masetti, G., D'hooge, J., 2016. Handling missing strain (rate) curves using K-nearest neighbor imputation. In: 2016 IEEE International Ultrasonics Symposium (IUS). IEEE, pp. 1–4.

Tlamelo, E., Thabiso, M., Dimane, M., Thabo, S., Banyatsang, M., Oteng, T., 2021. A survey on missing data in machine learning. J. Big Data 8 (1).

Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S., Herrera, F., 2019. Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. Wiley Interdiscip. Rev. Data Mining Knowl. Discov. 9 (2), e1289.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B., 2001. Missing value estimation methods for DNA microarrays. Bioinformatics 17 (6), 520–525.

Tsai, C.-F., Chang, F.-Y., 2016. Combining instance selection for better missing value imputation. J. Syst. Softw. 122, 63–71.

Tsai, C.-F., Lin, W.-C., Hu, Y.-H., Yao, G.-T., 2019. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Inform. Sci. 477, 47–54.

Tutz, G., Ramzan, S., 2015. Improved methods for the imputation of missing data by nearest neighbor methods. Comput. Statist. Data Anal. 90, 84–99.

Twala, B., 2009. An empirical comparison of techniques for handling incomplete data using decision trees. Appl. Artif. Intell. 23 (5), 373–405.

Valdiviezo, H.C., Van Aelst, S., 2015. Tree-based prediction on incomplete data using imputation or surrogate decisions. Inform. Sci. 311, 163–181.

Van Hulse, J., Khoshgoftaar, T.M., 2014. Incomplete-case nearest neighbor imputation in software measurement data. Inform. Sci. 259, 596–610.

Vanschoren, J., van Rijn, J., Bischl, B., Torgo, L., 2013. OpenML: Networked science in machine learning. SIGKDD Explor. 15 (2), 49–60.

Wang, C., Yang, Y., 2020. Nearest neighbor with double neighborhoods algorithm for imbalanced classification. Int. J. Appl. Math. 50 (1).

Wang, N., Zhao, S., Cui, S., Fan, W., 2021. A hybrid ensemble learning method for the identification of gang-related arson cases. Knowl.-Based Syst. 218, 106875.

Weinberger, K., Saul, L., 2009. Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. 10 (Feb), 207–244.

West, J., Bhattacharya, M., 2016. Intelligent financial fraud detection: a comprehensive review. Comput. Secur. 57, 47–66.

Wilk, S., Stefanowski, J., Wojciechowski, S., Farion, K., Michalowski, W., 2016. Application of preprocessing methods to imbalanced clinical data: An experimental study. In: Conference of Information Technologies in Biomedicine. Springer, pp. 503–515.

Wilson, R., Martinez, T., 1997. Improved heterogeneous distance functions. J. Artificial Intelligence Res. 6, 1–34.

Woźnica, K., Biecek, P., 2020. Does imputation matter? Benchmark for predictive models. arXiv preprint arXiv:2007.02837.

Zhang, S., 2011. Shell-neighbor method and its application in missing data imputation. Appl. Intell. 35 (1), 123–133.

Zhang, S., 2012. Nearest neighbor selection for iteratively kNN imputation. J. Syst. Softw. 85 (11), 2541–2552.

Zhang, P., Zhu, X., Tan, J., Guo, L., 2010. SKIF: a data imputation framework for concept drifting data streams. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1869–1872.

Zhao, F., Xin, Y., Zhang, K., Niu, X., 2021. Representativeness-based instance selection for intrusion detection. Secur. Commun. Netw. 2021.

Zheng, M., Li, T., Zhu, R., Chen, J., Ma, Z., Tang, M., Cui, Z., Wang, Z., 2019. Traffic accident's severity prediction: A deep-learning approach-based CNN network. IEEE Access 7, 39897–39910.

Zhou, T., Wang, S., Bilmes, J.A., 2020. Curriculum learning by dynamic instance hardness. Adv. Neural Inf. Process. Syst. 33.