UNIVERSIDADE D
COIMBRA

Ricardo Daniel Cardoso Pereira

Artificial Intelligence Strategies for Missing Data Imputation using Healthcare Data

PhD Thesis in Informatics Engineering, Intelligent Systems, supervised by Professor Pedro Manuel Henriques da Cunha Abreu and Professor Pedro Pereira Rodrigues, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra

June 2023

This page is intentionally left blank.

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

PhD Thesis in Informatics Engineering
Intelligent Systems

Artificial Intelligence Strategies for Missing Data Imputation using Healthcare Data

Author:
Ricardo Daniel Cardoso Pereira
rdpereira@dei.uc.pt

Advisors:
Professor **Pedro Henriques Abreu**, PhD
Professor **Pedro Pereira Rodrigues**, PhD

Coimbra, 2023

This page is intentionally left blank.

Acknowledgements

Pursuing a PhD has been the hardest challenge in my life. Throughout these past years, several people were crucial in supporting me and my work.

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Pedro Abreu, for all the support given during this PhD. He was always available to help and encourage me to move forward and achieve my goals. Without him, I would not have succeeded in many of them. I would also like to thank my co-supervisor, Prof. Pedro Rodrigues, for all the interesting discussions that helped refine my ideas.

I would also like to extend my gratitude to all the institutions that supported me during this PhD, especially the *Fundação para a Ciência e a Tecnologia*, for the financial support during these years, and the Cognitive and Media Systems group of the Centre for Informatics and Systems of the University of Coimbra, for the financial aid and for providing me with the necessary means to conduct my research.

Thank you to all my friends that have been part of this journey, particularly my research buddies Miriam and José, with whom I shared many achievements and frustrations.

Finally, I'm forever indebted to my family for all their unconditional support. To my parents, who have been helping and supporting me my entire life, and always want the best for me. To my sister, brother-in-law, and niece, who are always part of my life, even being one ocean apart. To Ana, for always being by my side, for helping me in everything, and for being a true partner in life. This thesis and my work are dedicated to all of you.

This page is intentionally left blank.

Abstract

Nowadays, most organizations rely on data to extract valuable insights, which can be obtained through simple statistical analysis or more complex machine learning models. The produced outcomes of such tasks depend on the quality of the data. An issue that highly impacts this quality is missing data, which is described as the absence of values in the features of a dataset. Missing values usually harm any procedure performed with the data since the distributions of the features can potentially be shifted and misrepresented. This is especially true when the missing data fall under the Missing Not At Random (MNAR) mechanism, which states that the missing values are related to themselves or other unobserved data.

Missing data is often handled with imputation strategies, which produce estimates to replace the missing values. Such strategies work well for all missing mechanisms, except for MNAR since they only rely on the observed data and, therefore, provide biased results. Due to that, new strategies are needed to improve the estimates for MNAR values, especially when considering that this mechanism has the highest prevalence in critical domains such as healthcare.

This thesis is focused on exploring and proposing new imputation strategies for the MNAR mechanism that can improve upon the current state-of-the-art results. To achieve this goal, two main research lines were followed: leverage and extend deep learning models for imputation purposes, particularly autoencoders since they are state-of-the-art for this task; and explore data processing strategies based on multiple data sources and post-processing adjustments of the imputed values.

Regarding the use of deep learning models, three new imputation strategies were proposed: the use of variational autoencoders for filtering data prior to the estimation, the extension of variational autoencoders to include a partial multiple imputation procedure, and a new siamese autoencoder-based approach. Moreover, a comprehensive survey about technical trends and applications of autoencoders for imputation was also introduced.

For the data processing research line, two strategies were proposed: the combination of multiple data sources to mitigate the issues caused by the

relation with unobserved data, and a method to automatically estimate the delta-adjustment factor used to reduce bias in the estimates. Furthermore, a set of artificial generation strategies for MNAR values was also introduced and benchmarked.

All the developed work was validated with data from the healthcare domain since this is a sensitive context highly impacted by MNAR. The used experimental setup assessed the imputation quality by comparing the estimated values with the original ones. In some works, the impact of the imputation in classification tasks was also measured. In general terms, the proposed strategies were able to achieve better imputation results when compared to the current state-of-the-art methods. In the future, we will extend these new strategies for non-tabular data (particularly images), and we will try to leverage other deep learning models for MNAR data (e.g., generative adversarial networks).

Keywords

Missing Data, Missing Not At Random, Data Imputation, Healthcare Data

Resumo

Atualmente, a maioria das organizações depende de dados para extrair informação útil, que pode ser obtida por meio de análises estatísticas simples ou modelos de aprendizagem computacional complexos. No entanto, os resultados produzidos por estas estratégias dependem da qualidade dos dados. Um problema que tem um grande impacto nessa qualidade é a falta de dados, que é descrita como a ausência de valores nas variáveis de um *dataset*. Os valores ausentes tendem a prejudicar qualquer procedimento realizado com os dados, já que as distribuições das variáveis podem estar alteradas e deturpadas. Estas alterações são particularmente visíveis quando os dados ausentes se enquadram no mecanismo *Missing Not At Random* (MNAR), que define que os valores ausentes estão relacionados com eles próprios ou com outros dados não observados.

Os dados ausentes são tipicamente tratados com estratégias de imputação, que produzem estimativas para substituir os valores em falta. Estas estratégias funcionam bem para todos os mecanismos exceto para o MNAR, porque dependem apenas dos dados observados e, por isso, produzem resultados enviesados. Nesse sentido, são necessárias novas estratégias para melhorar as estimativas dos valores em falta do tipo MNAR. Esta necessidade é particularmente premente já que este mecanismo tem maior prevalência em domínios críticos como o da saúde.

O foco desta tese é explorar e propor novas estratégias de imputação para o mecanismo MNAR que possam melhorar os resultados obtidos pelos métodos estado-da-arte. Para atingir este objetivo, duas linhas principais de investigação foram seguidas: potenciar e estender modelos de aprendizagem profunda para fins de imputação, particularmente *autoencoders*, uma vez que são considerados estado da arte para esta tarefa; e explorar estratégias de processamento de dados com base em várias fontes e em ajustes de pós-processamento dos valores imputados.

Em relação ao uso de modelos de aprendizagem profunda, três novas estratégias de imputação foram propostas: o uso de *variational autoencoders* para filtrar dados antes da geração das estimativas, a junção de *variational autoencoders* com um procedimento de imputação múltipla parcial, e uma

nova abordagem baseada em *siamese autoencoders*. Adicionalmente, foi também desenvolvia uma pesquisa abrangente sobre tendências técnicas e aplicações de *autoencoders* para fins de imputação.

No que toca à linha de investigação de processamento de dados, duas estratégias foram propostas: a combinação de múltiplas fontes de dados para mitigar os problemas causados pelo facto do MNAR estar relacionado com dados não observados, e um método para estimar automaticamente o fator de ajuste *delta* usado para reduzir o viés nas estimativas. Adicionalmente, foi também proposto um conjunto de estratégias de geração artificial de valores ausentes do tipo MNAR, tendo estas sido comparadas num estudo de *benchmark*.

Todo o trabalho desenvolvido foi validado com dados do domínio da saúde por se tratar de um contexto sensível e altamente impactado pelo MNAR. A configuração experimental utilizada foi focada na avaliação da qualidade da imputação, comparando os valores estimados com os originais. Em alguns trabalhos, também foi medido o impacto da imputação em tarefas de classificação. Em termos gerais, as estratégias propostas foram capazes de alcançar melhores resultados de imputação quando comparadas aos métodos atualmente considerados estado da arte. No futuro, iremos aplicar estas novas estratégias a dados não tabulares (particularmente imagens) e tentaremos potenciar outros modelos de aprendizagem profunda para lidar com dados em falta do tipo MNAR (por exemplo, *generative adversarial networks*).

Palavras-Chave

Dados em Falta, Mecanismo de Ausência Não Aleatória, Imputação de Dados, Dados Médicos

Contents

This page is intentionally left blank.

Acronyms

AAE Adversarial Autoencoder.

ACA Available Case Analysis.

ADAM Automatic Delta-Adjustment Method.

AE Autoencoder.

ANN Artificial Neural Network.

ANOVA Analysis of Variance.

ARIMA Autoregressive Integrated Moving-Average.

AUC-ROC Area Under the Receiver Operating Characteristic Curve.

CCA Complete Case Analysis.

DAE Denoising Autoencoder.

DMP Dynamic Movement Primitive.

EM Expectation-Maximization.

GA Genetic Algorithm.

GAIN Generative Adversarial Imputation Nets.

GFK Geodesic Flow Kernel.

HSD Honestly Significant Difference.

IDIM Information Decomposition Imputation.

IPW Inverse Probability Weighting.

KNN k-Nearest Neighbors.

LR Logistic Regression.

LSTM Long Short-Term Memory.

LTSL Low-rank Transfer Subspace Learning.

MAE Mean Absolute Error.

MAR Missing At Random.

MBIR Missingness Based on Intra-Relation.

MBOUV Missingness Based on Own and Unobserved Values.

MBOV Missingness Based on Own Values.

MBUV Missingness Based on Unobserved Values.

MCAR Missing Completely At Random.

MI Multiple Imputation.

MICE Multiple Imputation by Chained Equations.

MIMCMC Multiple Imputation Monte Carlo Markov Chain.

MIST Multiple Imputation by Sequential Regression Trees.

MIWAE Missing Data Importance-Weighted Autoencoder.

MkNNI Mutual k-Nearest Neighbours Imputation.

MLP Multi-Layer Perceptron.

MNAR Missing Not At Random.

MSE Mean Squared Error.

NRMSE Normalized Root Mean Square Error.

OLS Ordinary Least Squares.

PCA Principal Component Analysis.

PDF Probability Density Function.

PMIVAE Partial Multiple Imputation with Variational Autoencoders.

PSNR Peak Signal-to-Noise Ratio.

QRILC Quantile Regression Imputation of Left-Censored Data.

RAE Residual Autoencoder.

RDALR Robust Domain Adaptation with Low-rank Reconstruction.

ReLU Rectified Linear Unit.

RF Random Forest.

RMSE Root Mean Square Error.

SAEI Siamese Autoencoder-based Approach for Imputation.

SI SoftImpute.

SOR Sum of Ranks.

SVD Singular Value Decomposition.

SVM Support Vector Machine.

SVR Support Vector Regression.

TRAE Tracking-Removed Autoencoder.

TSL Transfer Subspace Learning.

VAE Variational Autoencoder.

VAE-BRIDGE Variational Autoencoder Filter for Bayesian Ridge Imputation.

VAE-WL Variational Autoencoder with Weighted Loss.

This page is intentionally left blank.

List of Figures

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

From simple statistical analyses to complex machine learning solutions, any task performed with real-world data is highly influenced by its quality. One of the most common issues that can be found in datasets is missing data. This issue can be described as the absence of values in specific features and instances of the datasets, and it can impact negatively most tasks performed with the data. For example, most machine learning methods cannot cope with missing values, and the ones that can tend to suffer significant performance decreases. Furthermore, a feature containing missing values may have its distribution changed, which leads to incorrect assumptions about the original complete data. However, missing values may present different characteristics according to their nature, and they can be categorized into three mechanisms: Missing Completely At Random (MCAR), which describes missing values that are the result of a purely random event; Missing At Random (MAR), which accounts for missing values that are directly related to other data available in the dataset; and Missing Not At Random (MNAR), which happens when the missing values are related with themselves or with other unobserved data. Although there are several strategies available to deal with the MCAR and MAR mechanisms, MNAR still lacks solutions that are capable of properly tackling its nature. This is particularly problematic since MNAR is the mechanism more commonly found in several critical contexts such as healthcare. Therefore, it is imperative to propose and provide new strategies that are suitable to deal with the MNAR assumptions.

1.1 Motivation and Research Questions

The missing data issue is often addressed in the preprocessing stage of the data mining process. Most often the instances containing missing values are deleted, which is a very limiting approach that may lead to a significant loss of information. A more suitable strategy is imputation, which is the process of replacing the missing values with estimates of the original data. A simple imputation strategy is to replace the missing values with

the mean/mode of each feature. Nevertheless, this strategy is only valid when the missing values are under the MCAR mechanism. There are other more resilient imputation strategies, such as the Multiple Imputation by Chained Equations (MICE) or more recent deep learning-based methods, which can provide unbiased estimates for both the MCAR and MAR mechanisms. The trade-off for these methods is that they usually have a higher time complexity or require specific assumptions (e.g., several of them require complete data to be trained). However, most of these approaches provide estimates based on the observed data, which makes them unsuitable for the MNAR mechanism. Among the deep learning methods, autoencoders (AE) have been highlighted in recent literature as models that can provide great imputation results by learning highly complex patterns in the data [8, 35, 70, 89, 93]. They also show some resilience to the MNAR characteristics, which makes them ideal candidates to be further explored with this mechanism [8, 35, 36].

Considering that missing values under MNAR assumptions are related to unobserved data, no imputation method can provide completely unbiased estimates. Most often the estimates are generated assuming the MAR mechanism and they are post-processed to reduce the uncertainty towards the MNAR characteristics. Common strategies used for this purpose include sensitivity analysis [16], multiple imputation procedures [41], or the delta-adjustment method [108]. However, most of these strategies require manual inputs from domain experts, which makes them unfeasible when no prior knowledge of the data and the domain exists. Furthermore, the dependency on human inputs drastically reduces the applicability of such strategies. This is particularly problematic when we consider that the MNAR mechanism is frequently found in highly relevant domains such as healthcare. For example, in studies based on medical data that is routinely collected there is a high chance that the patient is going to miss some of the visits and measurements due to personal reasons that are not related to the study data, and neither are completely random (e.g., the patient may decide to stop the treatment). These missing values, which meet the MNAR characteristics, will highly influence any task performed with the study data, so it is important to provide solutions for such scenarios.

Tackling the MNAR mechanism is still an understudied topic, with very limited and human-based solutions. In this thesis, we aim to improve such solutions by leveraging artificial intelligence-based strategies to overcome the existing limitations. To achieve this goal, we focused our research on two specific directions: extend autoencoder models to perform the imputation of missing data under MNAR assumptions and explore automatic ways of improving and adjusting the MNAR imputed values. Both directions can sometimes overlap and complement each other. Furthermore, considering the impact of the MNAR values in healthcare, we focused the validation of the proposed solutions in this domain. Based on these goals, we aimed to answer the following two research questions:

- **RQ-1: Can we extend and improve autoencoder-based models to impute MNAR values?**

- **RQ-2: Can we automate human-based strategies used to improve the estimation of MNAR values?**

We developed several works to answer each of these research questions. We present and describe them in Parts I and II of this thesis, which respectively answer **RQ-1** and **RQ-2**. These works originated several publications, which are listed in Section 1.2. We also provide a joint discussion of the answers obtained for these research questions in the thesis conclusions.

1.2 Research Contributions

As part of the work developed in this thesis, the following articles were published in the mentioned conferences and journals[1]:

- Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Siamese Autoencoder-based Approach for Missing Data Imputation. In *International Conference on Computational Science 2023 (ICCS 2023)*, 2023 [**CORE Rank A**]
- Ricardo Cardoso Pereira, Pedro Pereira Rodrigues, Mário A. T. Figueiredo, and Pedro Henriques Abreu. Automatic Delta-Adjustment Method applied to Missing Not At Random Imputation. In *International Conference on Computational Science 2023 (ICCS 2023)*, 2023 [**CORE Rank A**]
- Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Partial Multiple Imputation with Variational Autoencoders: Tackling Not at Randomness in Healthcare Data. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4218–4227, 2022 [**Computer Science (Q1), Information Systems (Q1), Medical Informatics (Q1), Interdisciplinary Applications (Q1)**]
- Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. Reviewing Autoencoders for Missing Data Imputation: Technical Trends, Applications and Outcomes. *Journal of Artificial Intelligence Research*, 69:1255–1285, 2020 [**Computer Science (Q2), Artificial Intelligence (Q2)**]
  - This paper was also accepted and presented in the Journal Track of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021) [**CORE Rank A***].
- Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. VAE-BRIDGE: Variational Autoencoder Filter for Bayesian Ridge Imputation of

[1]The conference ranks are from the Computing Research and Education Association of Australasia (CORE) portal, and the journal quartiles are from Clarivate (previously Thomson Reuters). All ranks and quartiles refer to the date of publication.

Missing Data. In *2020 International Joint Conference on Neural Networks (IJCNN 2020)*, pages 1–7, 2020 [**CORE Rank A**]

- Ricardo Cardoso Pereira, Joana Cristo Santos, José Pereira Amorim, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. Missing Image Data Imputation using Variational Autoencoders with Weighted Loss. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2020)*, pages 475–480, 2020 [**CORE Rank B**]

- Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. MNAR Imputation with Distributed Healthcare Data. In *19th EPIA Conference on Artificial Intelligence (EPIA 2019)*, pages 184–195, 2019 [**CORE Rank Regional**]

- Miriam Seoane Santos, Ricardo Cardoso Pereira, Adriana Fonseca Costa, Jastin Pompeu Soares, João Santos, and Pedro Henriques Abreu. Generating Synthetic Missing Data: A Review by Missing Mechanism. *IEEE Access*, 7:11651–11667, 2019 [**Computer Science (Q1), Information Systems (Q1)**]

Moreover, the following article has been submitted and is currently under revision:

- Ricardo Cardoso Pereira, Pedro Henriques Abreu, Pedro Pereira Rodrigues, and Mário A. T. Figueiredo. Imputation of Data Missing Not At Random: Artificial Generation and Benchmark Analysis. *Expert Systems With Applications* (*submitted on July 17th, 2022*)

Based on these publications, this thesis was divided in Parts I and II. Each chapter presents a work with the necessary contextualization.

In conclusion, the work developed during the course of this thesis culminated in **2 papers published in Q1 journals** (plus an additional paper that is currently under revision), **1 paper published in a Q2 journal** (which was also **presented in a A*-ranked conference**), **3 papers published in A-ranked conferences**, **1 paper published in a B-ranked conference**, and **1 paper published in a Portuguese conference**.

1.3 Document Structure

The remainder of the document is organized in the following chapters:

- Chapter 2 introduces the missing data related concepts and a comprehensive literature review of works that address the Missing Not At Random (MNAR) mechanism;

- Chapter 3 presents a review of autoencoders used for missing data imputation;

- Chapter 4 describes the Variational Autoencoder Filter for Bayesian Ridge Imputation approach;
- Chapter 5 introduces the Partial Multiple Imputation with Variational Autoencoders method;
- Chapter 6 proposes the Siamese Autoencoder-based Approach for Imputation;
- Chapter 7 introduces an imputation strategy based on distributed data;
- Chapter 8 describes the Automatic Delta-Adjustment Method;
- Chapter 9 introduces novel artificial generation strategies for MNAR values;
- Chapter 10 draws the final conclusions of this thesis.

This page is intentionally left blank.

Chapter 2

State of the Art

This chapter presents the background knowledge and state of the art for this thesis. In particular, the missing data concepts are introduced (Section 2.1), followed by a description of common approaches to address this issue (Section 2.2). Finally, a literature review focused on the Missing Not At Random mechanism is also presented (Section 2.3).

2.1 Missing Data Concepts

Little and Rubin [54] defined the missing data concept as the unobserved values that would be meaningful for analysis if existed (i.e., the missing values would add meaningful information to the data). Considering a common tabular dataset, where observations are usually represented by rows and features by columns, one may think the only difference between the missing values is the type of data from each feature. However, an important distinction between them is their likelihood of being missing. Rubin [87] introduced the concept of missing mechanism to classify the missing values according to these probabilities through three different types: Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR). Assuming R as the binary matrix where the missing values of Y are represented, Y is composed by the observed values Y_{obs} and the missing values Y_{mis}, and ψ as the parameters of the missing data model, these missing mechanisms can be defined in the following way [115]:

- Missing Completely At Random (MCAR) - the missing values are said to be under this mechanism when their probability of being missing is always equal, which allows the assumption that their is not relation between the missingness causes and the observed data. For this reason, MCAR is the easiest mechanism to deal with, where solutions such as ignoring the missing values are valid (although not recommend for reasons related with loss of information). From a formal perspective, the missing data is said to be MCAR if Equation 2.1 holds, since the missing probability only

depends on the ψ parameters and not on the data.

$$Pr(R = 0|Y_{obs}, Y_{mis}, \psi) = Pr(R = 0|\psi) \tag{2.1}$$

- Missing At Random (MAR) - when the probability of missingness is only equal for certain observed groups (e.g., specific observed values of a feature), the missing values are said to be under this mechanism. In other words, the missing values are directly related to some part of the observed values. This is a more realistic mechanism than MCAR and, although it is not as easy to work with, there is a substantial amount of methods that can be used to handle it by modeling the data correlations. From a mathematical perspective, the probability behind this mechanism is stated in Equation 2.2, since it is now related with ψ parameters and also with the observed data.

$$Pr(R = 0|Y_{obs}, Y_{mis}, \psi) = Pr(R = 0|Y_{obs}, \psi) \tag{2.2}$$

- Missing Not At Random (MNAR) - in a scenario where the reasons behind the missingness probability are unknown and, therefore, are related to unobserved data, the missing values fall in this mechanism. In other words, both the MCAR and MAR assumptions do not hold. Moreover, this mechanism is usually identified in two different settings: the missing values are related to themselves or to other features that are not available in the dataset. MNAR is the hardest mechanism to address since it cannot be ignored or modeled by the available data. Therefore, the solutions to handle it usually rely on studying the reasons behind the missingness or sensitivity analysis, where multiple scenarios are considered to evaluate how resilient are the results. Formally, the probability for MNAR is described by all components, as Equation 2.3 shows.

$$Pr(R = 0|Y_{obs}, Y_{mis}, \psi) \tag{2.3}$$

To exemplify the three mechanisms, lets consider the data from Table 2.1. This small dataset has 15 observations for two features: the number of accidents that a person had and the insurance price for that individual. The third feature (age of the person) is not available on the dataset (i.e., it is an unobserved feature). Moreover, the missing values from the insurance price feature are represented by the "?" symbol.

Analyzing the missing values from the Table 2.1 dataset, their classification to each mechanism can be described by the following motives:

- The MCAR values are not related with the number of accidents and neither with the age of the person, making them *completely random*. This could mean that people simply forgot to insert the value or that there was an error collecting that information;
- The MAR values only occur for observations with a number of accidents greater

Table 2.1: Example of the missing data mechanisms. The *Age* feature, signed with *, is not part of the dataset (i.e., it is an unobserved feature). The symbol "?" represents the missing values.

Nº of Accidents	**Insurance Price**					**Age ***
	Complete	**MCAR**	**MAR**	**MNAR 1**	**MNAR 2**	
5	720	?	720	720	?	66
2	360	360	360	360	360	45
10	1320	?	?	?	1320	28
5	720	?	720	720	720	22
7	960	960	?	?	960	42
3	480	?	480	480	480	40
2	360	?	360	360	?	75
8	1080	1080	?	?	1080	54
4	600	?	600	600	600	32
7	960	?	?	?	?	72
1	240	240	240	240	240	53
0	120	120	120	120	120	20
6	840	840	?	840	840	37
9	1200	?	?	?	1200	28
6	840	?	?	840	?	69

than or equal to five, which means they are related to this feature. This could mean that people with an higher rate of accidents tend to not disclosed their insurance price;

- The MNAR "1" values occur when the insurance price is greater than or equal to 900, meaning that they are related with themselves. This probably means that people paying higher insurance prices do not want to report them;

- The MNAR "2" values occur for observations of people with an age greater than or equal to 60 (although this feature is not part of the dataset, being for that reason unobserved). This could mean that older people, for cultural and trust reasons, may not feel comfortable disclosing their insurance prices.

2.2 Handling Missing Data

The missing data problem cannot be neglected, otherwise it will have a major impact in tasks that use the data. For example, if the dataset is used for a classification task, most classifiers cannot be trained with missing values and the ones that can cope with this issue tend to have a significant decrease in their accuracy. García-Laencina et al. [30] proposed a taxonomy for the different approaches available to deal with missing data, and it is used

and extended in this thesis. A representation of this taxonomy is presented in Figure 2.1, and its methods are described in the following subsections.

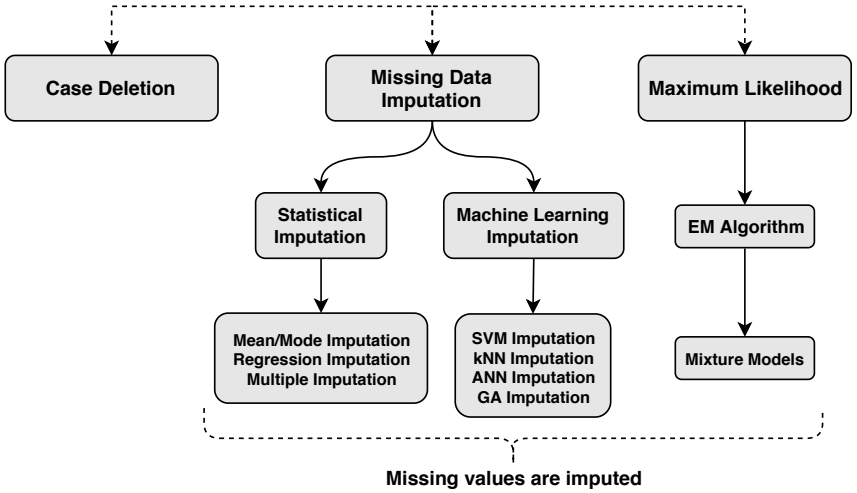

Figure 2.1: Taxonomy of methods to handle missing data (adapted from [30]).

2.2.1 Case Deletion

A method often used to deal with missing data is simply removing it. This is called case deletion, and the key idea is to delete the observations that contain missing values. However, two different deletion approaches can be applied [54]:

- Listwise deletion, also called casewise deletion or Complete Case Analysis (CCA), where all observations containing at least one missing value are eliminated, with no regard for the target use of the dataset. It is a very simple procedure, but can lead to the lost of a considerable amount of information, being for that reason only recommended in big data scenarios or when the observations with missing values are less than 5% of the data;
- Pairwise deletion, also known as Available Case Analysis (ACA), where only the observations containing missing values for the features of interest are deleted. As an example, a dataset may contain 40 features but for a specific analysis only 30 are used. Observations that are complete for the 30 features of interest but have missing values in the other 10 features should not be deleted, because they are considered complete for this specific analysis. Although this approach suffers from the same problems of listwise deletion, it reduces the loss of information.

To apply any of the deletion strategies, the missing mechanism must also be considered. Deleting data may remove relations between observations and features, and for that reason this approach should only be applied with missing values under MCAR [54].

2.2.2 Statistical Imputation

When deleting observations is not a suitable option, imputation tends to be the preferred strategy. The key idea is to generate plausible new values to replace the missing ones. Among the existent methods, statistical ones are frequently used [30]. A common approach is to use the mean or mode of the feature containing missing values for numerical and categorical variables, respectively [54]. In both cases all the available values for that feature are considered, but since the relations with other variables are neglected this approach should only be applied for MCAR values. Another strategy is to use a regression to model one or more dependent variables (the ones containing missing values) using as independent variables the remaining features of the dataset [54]. The fitting process must only consider the observations that are complete for the independent features, and the type of regression (e.g., linear or non-linear) used must be chosen taking in consideration the nature of data. The main advantage of this approach is that it considers the relation between the features with missing values and the ones that are complete, being therefore suitable for MAR scenarios [98].

Both approaches are said to be single imputation strategies, since only one value is used for the imputation of each missing value. For this reason, the imputation results may be biased because the uncertainty of the generated values is not accounted for. To address this issue multiple imputation strategies can be applied. The key idea is to perform the imputation M times, generating different but complete results each time (different imputation methods may be used). The M complete datasets are then analyzed and combined, being the result of this combination the final dataset [88]. The state-of-the-art method that uses this approach is the Multiple Imputation by Chained Equations (MICE) [15]. It creates a series of regressions where each one is modeled for each variable with missing values, meaning that each feature is modeled conditionally upon the other features. The method can be described by the following sequential steps [5]:

1. All missing values are pre-imputed with a single imputation strategy, usually the mean/mode method;

2. Each feature that previously contained missing values is set back to its state before the pre-imputation, in a sequential and alternated way;

3. Each feature from the previous step is modeled using a type of regression (linear or not), using all the remaining features as the independent variables and considering the complete observations for the fitting process;

4. The result is a complete dataset imputed by the multiple regressions. However, simple decisions like the order of the features imputed may create bias in this dataset. To remove it the process is repeated throughout several iterations until the parameters of the regressions converge to stable values.

The characteristics of the MICE method make it suitable for missing data under MAR, since it uses regressions and preserves the relationship between the features. However, considering that it follows a multiple imputation strategy that deals with uncertainty, some authors also use this and similar approaches for MNAR values [41]. The main reason is that one can look at multiple imputation as a way to perform sensitivity analysis[1], which is the most common test applied to MNAR scenarios.

Other approaches based on statistical and algebra concepts are also used, although less frequently. One is the Inverse Probability Weighting (IPW), which relies on a model that states how probable are the observed values of a feature with missing data of being observed, based on other feature that is correlated. This model is usually a logistic regression. The imputation is then performed through another model, fitted with the same data, but using a weighted least squares regression where the weights are the inverse of the outcomes from the logistic regression for each value [117]. Matrix completion methods, such as the Singular Value Decomposition (SVD) or the SoftImpute, are also found in some works. The goal is to perform the imputation by finding a low-rank matrix that is an approximation of the original one. The process usually relies on matrix factorization, where a matrix with (m, n) dimensions is decomposed into two matrices with dimensions (m, k) and (k, n), where k is the rank and must be smaller than m and n. The idea is to find the latent features that better describe the available values. The low-rank approximation matrix can finally be obtained by simply multiplying the resulting matrices from the decomposition process [113]. These methods are only valid for MAR and MCAR scenarios, since they rely on the available data from the original matrix.

2.2.3 Machine Learning Imputation

Following the same imputation concepts from the statistical methods, this task can also be achieved by using machine learning models. In theory, any algorithm that can be trained to predict new values is suitable for imputation. Generating new plausible values for numerical features may be seen as a regression problem and as a classification problem for categorical features. The more common algorithms used for this purpose are Artificial Neural Networks[2] and Support Vector Machines[3]. However, the typical pipeline for

[1] Sensitivity analysis is a test to evaluate how resilient is a model or system when confronted with different but plausible input scenarios. The consistency is obtained when the results are similar for the tested variations.

[2] Artificial Neural Network is a machine learning model based on the biological structure and functionality of the human brain. The network is composed by a set of nodes inspired by the human neurons that are interconnected. This type of algorithm is considered to be a general approximator, since it is able to map any function through a training process that uses available observations [40].

[3] Support Vector Machine is a machine learning model capable of performing binary classification by defining the hyperplane that maximizes the distance between the labels. This algorithm performs linear separation and it is able to find an optimal solution for data that meets this criteria. However, it also works with data non-linearly separable by using a kernel function that maps it into a different space where the labels can be separated through a linear hyperplane [4].

supervised algorithms must be adjusted for this imputation task, as Figure 2.2 shows [30]:

1. To prepare the dataset for the training process only the complete observations can be considered;

2. The feature having missing values must be used as the target. When more than one feature meets this criteria the process is usually repeated for each one;

3. The model is trained with the pre-processed dataset. As stated before, any supervised machine learning model may be used for this task. The decision of tackling this as a regression or classification problem is based on the target feature;

4. The trained model is used to predict the missing values of the observations that were removed from the dataset in the first step. Error metrics such as the mean absolute error or root mean squared error may be used to assess the imputation results, assuming that the ground truth is available.

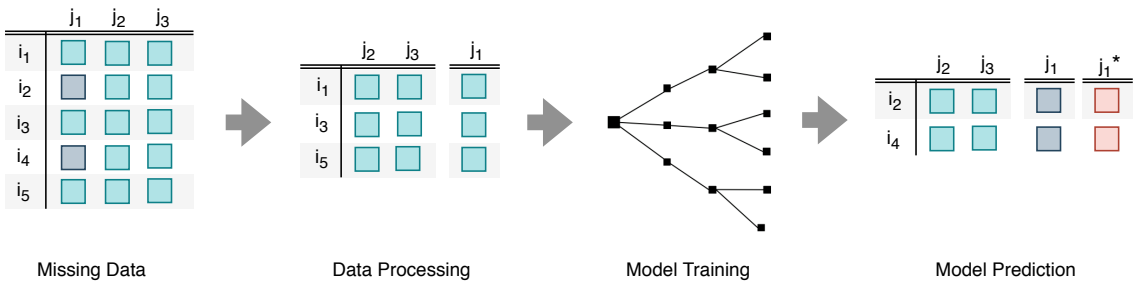

Figure 2.2: Machine learning pipeline for missing data imputation.

A specific algorithm that is widely used for imputation is the k-Nearest Neighbors (KNN) [6, 31]. It tries to find the k most similar observations to the one that contains missing values using only the features that are complete. To calculate this similarity a distance function is used, and it must be chosen taking in consideration the data type: the euclidean distance is suitable for numeric data but not for categorical. For this last case the data can be converted (one-hot encoding transforms a categorical feature into several binary ones, allowing the use of the euclidean distance) or a different distance function must be used (e.g., hamming distance works with categorical data) [7]. When $k > 1$ the values of the neighbors observations must be combined to produce the new value. For numerical data a common approach used is the simple or weighted mean, and for categorical values a vote of majority may be applied. The KNN method has a high computational cost but produces very plausible and explainable results. Nevertheless, the k parameter is quite sensitive, and different values may produced very different results.

Within deep learning, Denoising Autoencoders (DAE) [118] have been recently used for imputation. They are a type of neural network that learns a representation of the data

from the input layer and tries to reproduce it at the output layer. The Denoising variant is the one used for missing data imputation because it is designed to recover noisy data, which can exist due to data corruption via some additive mechanism or by the introduction of missing values [17]. However, the Variational variant with generative capabilities can also be used. While the basic Autoencoder simply learns a compressed representation of the input data in a unsupervised way, the Variational Autoencoder ((VAE)) learns the parameters of a probability distribution representing that data, namely the mean and variance of a Gaussian curve. By sampling from these learned parameters, the model is able to generate new data with the same characteristics [44]. For both variants, the only necessary step to take before training is to pre-impute the missing values. Different strategies can be used, but usually a simple replacement by zero or the mean/mode is done. Moreover, for the Denoising variant, in some scenarios noise may be added to force the generation of missing values. One of the possible corruption techniques consists in setting to 0 a fixed amount of features for a set of observations, which can be seen as a dropout[4] on the input layer. There are other possible corruption processes, such as adding Gaussian or salt-and-pepper noise to the input data [120]. Apart from autoencoders, the well-known Generative Adversarial Nets have also been used to perform imputation, particularly the Generative Adversarial Imputation Nets (GAIN) [126].

Finally, in rare occasions genetic algorithms have also been used to address the imputation task. These are heuristic algorithms inspired on the Darwin's theory of evolution that use operators such as reproduction, mutation, crossover and selection to generate new solutions within a population [67]. They are used to solve optimization problems through minimizing or maximizing a pre-established fitness function (i.e., objective function). The imputation of missing values may be seen as an optimization problem since the goal is to minimize the error between new solutions and a known ground truth, which requires the fitness function to be a error metric [56].

2.2.4 Maximum Likelihood

The maximum likelihood method is a mathematical approach to estimate the parameters of the distribution that better describes the observed data. This distribution will be the one that maximizes the probability of the data being in fact observed. The likelihood can be seen as the join probability of the variables under analysis, and in most scenarios, when all these variables are available, it can be calculated explicitly. However, when at least one of the variables is not fully observed (and therefore contains missing values), the likelihood must be estimated because it can no longer be calculated. The algorithm most used for this estimation task is the Expectation-Maximization (EM) [24]. It starts by assigning random values to the parameters and by creating the respective probability distribution

[4]Dropout is a regularization technique that randomly drops units from an neural network to avoid overfitting [106].

(i.e., the expectation step). After that, new observed data is fed into the model and the distribution is adjusted to account for its impact (i.e., the maximization step). The process is repeated until the parameters converge (i.e., the probability distribution does not change from the expectation to the maximization step). Although it is widely used, this method offers a high computational cost, which is a consequence of a considerable amount of iterations performing heavy calculations. Moreover, another limitation of this algorithm is its dependency of the initial random values. In another words, the iteration process may stop on a local optimal solution. Regarding the missing mechanism, EM assumes the missing values are under MAR [66].

The complexity of the data may also have an impact on the EM performance. For example, lets consider the data series plotted on Figure 2.3. When applying the algorithm to this data, it would not be able to create a probability distribution that fitted properly the entire series. However, it can be much better described by two Gaussian distributions with the parameters ($\mu = 3, \sigma = 1.5$) and ($\mu = 17, \sigma = 2.5$). This attempt to describe a data series with multiple sub-populations instead of an overall one is called a mixture model, and the EM algorithm can be applied using this concept with k different probability distributions [33]. A particularly type of mixture widely used to model numeric values is the one that follows a normal distribution (Gaussian Mixture Model). On the other hand, categorical values are usually modeled as a mixture of Bernoulli densities.

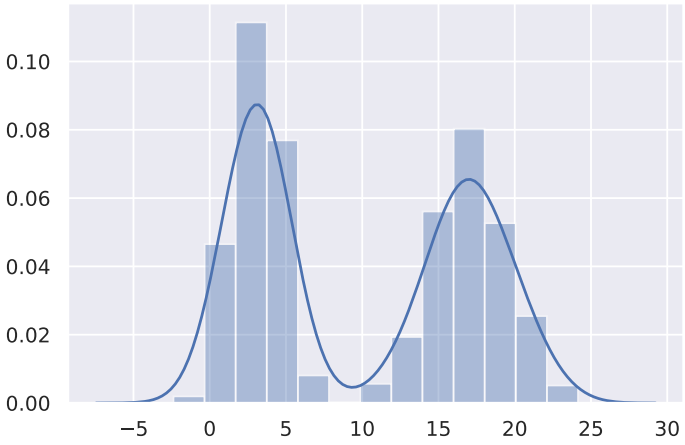

Figure 2.3: The distribution plot of a data series.

2.3 Literature Review

This section presents the most recently published works about the MNAR mechanism and how it is handled. Its contents are organized in the following way: Section 2.3.1 presents

a summarized description of the key aspects from each work; and Section 2.3.2 draws final conclusions and presents open challenges.

2.3.1 Missing Not At Random

Carreras et al. [16] conducted a sensitivity analysis study to understand the impact of assuming and treating missing data as MAR or MNAR in end-of-life care studies. The MICE method was used for both mechanisms, but for MNAR it was integrated with the delta-adjustment method. Four different adjustment values were considered, which were defined as the equispaced values between zero and half of the interquartile range of the features. The experiments were conducted with data from the ACTION study, a randomized controlled trial testing advance care planning in patients with advanced lung or colorectal cancer. The authors concluded that the imputation assuming MAR reflected that the missing values were related to poorer health conditions. These correlations changed when the MNAR mechanism was assumed, which shows that the obtained conclusions are sensible to the violation of the MAR assumptions.

Tan et al. [108] proposed a review study about the use of controlled multiple imputation in randomized controlled trials where missing data exists. The analysis considered the trials in phases II, III and IV published in The Lancet and New England Journal of Medicine between January 2014 and December 2019, covering primary and sensitivity analysis studies. The findings show that 56% of the controlled multiple imputation was performed with the delta-adjustment method. Nevertheless, most of the works report the used delta values but do not provide justifications to why the experts decided towards those values.

Boquet et al. [12, 13] proposed a method that uses a VAE for imputation and connects its output directly to a standard neural network for regression, aiming to solve the missing data issue before performing traffic forecasting. The VAE is compared with a standard Autoencoder ((AE)) and the Principal Component Analysis (PCA) method, but the imputation is not assessed directly. Instead, only the forecast error is evaluated through the Root Mean Square Error (RMSE) metric and the Mean Absolute Percentage Error (MAPE). The experiments used real traffic data from the freeway Performance Measurement System of the California Department of Transportation, and missing values under the MNAR and MCAR mechanisms were injected (the last one with rates of 10%, 20% and 40%). The forecasting results when the VAE was used outperformed the remaining methods, with improvements of at least 40% for MNAR and 17% for MCAR.

Qiu et al. [85] proposed the use of VAEs to perform the imputation of missing values under MNAR assumptions in genomic data. The authors proposed an extension to the vanilla VAE where a shift correction is applied to the decoder output in order to account for the probable distribution bias caused by the MNAR mechanism. The experiment

was conducted with two datasets, namely the pan-cancer RNA sequencing data from the Cancer Genome Atlas and DNA methylation data, and different levels of missingness were considered: 5%, 10%, 30% and 50% missing rates. The VAE and its extension were compared with the kNN and SVD methods, with the results being reported through the RMSE metric. While the VAE and the SVD presented similar results, the VAE with shift correction outperformed the remaining imputation methods for all missingness rates.

Pham et al. [84] proposed a strategy to reduce the bias of multiple imputation approaches under MNAR by obtaining the population marginal distribution of features with missing values from external datasets and using that information to properly calibrate the imputation model trough an offset adjustment. The key idea is to extract the distribution from a feature in an external source and, during the multiple imputation process, the offset will ensure the feature with missing values is adjusted to that distribution. The approach is compared with the CCA method for a simulated dataset, and it is also compared with a single imputation strategy and a standard multiple imputation for a real-world dataset with electronic health records of an unspecified number of patients from London. The results are first presented for the simulated dataset, with the error being very similar between the experimented strategies. Nevertheless, from the bias perspective, the results are considerably better for the new approach when comparing to the CCA. The same conclusions are obtained from the experiments with the real data, although the standard multiple imputation surpassed in general the CCA.

Beaulieu-Jones et al. [9] presented a study where electronic health records data is injected with missing values under the three mechanisms and also through real data modeling. Different imputation approaches are used to fill in the missing data: random sample, mean, median, KNN, SVD, SoftImpute and several MICE variants. The used dataset contained records from 602366 patients with 146 variables (age, sex, body mass index and 143 laboratory measures) from the Geisinger Health System in Pennsylvania, but only the 100000 complete observations more representative of the overall population were considered in the experiment. The results show that the best Root Mean Square Error (RMSE) values were obtained with the MICE variant that uses predictive mean matching with five nearest neighbors. However, these results are consistently worse for the MNAR mechanism when comparing with MAR and MCAR – in some cases the RMSE values are the double for MNAR. The authors also suggest that with this mechanism sensitivity analysis is highly recommended to understand the bias of the imputation in the final results.

White et al. [123] proposed a new imputation approach based on the mean score method, which handles the missing data by considering the expectation over the distribution of the missing values given the observed data. The proposed approach uses pattern-mixture models for the values estimation, and performs sensitivity analysis when the missing mechanism is not MAR. This analysis is conducted for all features, with the sensitiv-

ity parameters being varied over a range of plausible numerical values (i.e., values that are admissible and probable within the features domain). The approach is compared with missing data deletion, CCA, multiple imputation and selection model with IPW. The experiment considered four synthetic datasets generated by the authors to cover different levels of departure from the MAR mechanism (bigger departures lead to stronger MNAR assumptions). The results are discussed from a bias perspective, with the CCA and data deletion methods presenting always biased results. The bias of the remaining methods is minimal, but the proposed approach has the lowest Monte Carlo error.

Liu et al. [55] proposed a new imputation method specific for MNAR, called Information Decomposition Imputation (IDIM), based on a fuzzy membership function and the KNN algorithm. The approach starts by finding the 1-dimensional linear-information distribution of the dataset through a trapezoidal fuzzy membership function, being this information calculated to each feature of the data. In the case of all returned values being zero, the KNN algorithm is used to find and replace the missing values, otherwise the mean value of the column is used (this also happens when the KNN cannot be applied). The process is repeated iteratively until the Normalized Root Mean Square Error (NRMSE) is below a given value. The use of the membership function is the key to tackle the MNAR nature, assuming that the missing values do not exceed the maximal values of the attributes. The method is compared with the regularized expectation-maximization, mixture of kernel imputation method, KNN imputation, local least squares implementation, Bayesian principal component analysis imputation and the mean impute method. The experiment was conducted with five public datasets from the UCI Machine Learning Repository and Standard and Poor 500 index historical stock data, with 7 to 14 attributes and 500 to 4177 observations. The results show that the proposed approach surpassed the remaining ones, presenting NRMSE values generally below 0.1.

Mason et al. [62] presented a new Bayesian framework for the cost-effectiveness analysis of data from a health economics study with missing data. Firstly, the authors try to minimize the missing values by adding information from previous studies and domain experts (e.g., patients may have records from previous visits to the hospital or some missing values may be estimated by domain experts based on the available information). The statistical model is then formulated, using pattern-mixture models to address the MNAR mechanism by performing sensitivity analysis. The sensitive parameters and their ranges are defined with the help of domain experts. The approach was tested with data from a study called IMPROVE, which compared an emergency endovascular strategy with open repair for patients with ruptured abdominal aortic aneurysm. The dataset contained information from 301 patients, but only 138 have complete records. These observations encompassed features such as the age and the Hardman Index, since both are considered strong predictors of an individual emotional and physical well-being. The results are compared with the ones from the IMPROVE study by calculating the incremental net benefits of both. The purpose was to see if the previous study results continued valid after

the sensitivity analysis from the proposed approach. The probability of the incremental net benefits being positive is still high for most sensitivity analysis scenarios under MNAR. However, the probabilities vary between 76% and 97%, which shows a decrease in the confidence of the study when comparing to the 98% result obtained from the CCA.

Lin and Chu [52] proposed a new approach to jointly model multiple factors from different studies, called multivariate meta-analysis of multiple factors. Like any meta-analysis approach, the key idea is to combine data from multiple studies to generate a final dataset of results. By performing a multivariate approach over multiple factors, the final dataset is more generalizable since the features follow a more representative population instead of different sub-populations. To deal with scenarios where information is missing a Bayesian method is used for imputation. The approach was compared with other multivariate meta-analysis variants which ignore the missing values between studies. The results are evaluated through the RMSE metric and the bias. A simulated scenario with 30 datasets and 5 features was first considered, followed by a real scenario with a data from 30 studies and 8 factors related to the most common eye conditions. The results were similar in both scenarios, with the RMSE values presenting small differences between the tested models. However, the bias is considerably smaller for the proposed approach when the missing values are under MNAR. Hubbard et al. [39] proposed a similar approach using a Bayesian latent class model for electronic health records phenotyping that is able to cope with missing values under MNAR. However, the model also considers rules that are defined by domain experts. The approach is compared with a standard rule based strategy and the results show improvements with the new method, both in sensitivity (95.9% vs 91.9%) and specificity (99.7% vs 90.8%).

Sun et al. [107] proposed the use of instrumental variables (i.e., a variable that is directly correlated with the independent variable(s) and indirectly with the dependent one(s)) to assist on the identification of the distribution from a feature with missing values under MNAR, improving the imputation results as a consequence. Three methods were used and compared for the imputation task: IPW, a regression estimator and a doubly robust estimator based on the mean outcome. The experiment was conducted with a dataset containing 4997 records (81% of them complete) from a household survey of Botswana about HIV seroprevalence in adults. The results are discussed through the bias of the methods and their estimation outcomes, considering a 95% confidence level. The use of instrumental variables presents clear benefits in the results, with the confidence intervals of the estimators becoming wider and, therefore, more representative of the uncertainty inherent to MNAR. Moreover, all three methods had a similar performance, with the regression presenting slightly worse results.

Leurent et al. [50] presented a framework and a set of guidelines on how to properly perform sensitivity analysis to assess how robust is the imputation for missing values under MNAR assumptions. The authors identify the use of pattern-mixture models as the best

approach since it provides more interpretable parameters (e.g., relies on metrics such as the mean difference between missing and observed data), is the most used strategy in clinical trials, and can be easily implemented through common missing data imputation methods. The work also states that one of the strategies to perform this type of analysis is to apply multiple imputation assuming the MAR mechanism and perform several changes in the imputed data that could stand as plausible MNAR scenarios (e.g., adding or multiplying the values by a scalar). The resulting datasets can then be analyzed individually or combined through the Rubin's rules. This approach is experimented in a cost-effectiveness scenario with data from the 10TT trial, which was a controlled trial of weight-loss for obese adults in the United Kingdom. The dataset contained records from 537 participants, but only 31% were complete. Several MNAR scenarios were considered during the experiment by multiplying the imputed values by a scalar. The results, compared through the probability of the trial being cost-effective, showed inconsistency among six MNAR scenarios and one MAR (used as baseline for comparison). The probabilities varied between 19% and 75% over them, which shows that the imputed values are probably biased when assuming the MNAR mechanism.

Rezvan et al. [86] conducted a sensitivity analysis study where the missing values imputed under MAR with multiple imputation were shift to MNAR with the delta-adjustment method. The data used in the experiments was from the Longitudinal Study of Australian Children, and the goal was to estimate the association between exposure to maternal emotional distress at the age of four/five years and total difficulties at the age of eight/nine years. The adjustment values were defined with the help of domain experts through an elicitation process that allows for the formulation of the expert's feedback into a probability distribution. The authors concluded that there are significant increases in the magnitude of the association between maternal distress and total difficulties when the MNAR assumptions are assumed with a large departure from MAR.

Wei et al. [121] presented a study where different imputation methods are experimented with metabolomic data based on mass spectrometry. This type of data frequently has missing values because specific compounds often cannot be identified or quantified in some samples, mainly for biological and technical reasons. Moreover, these missing values can be from the three missing mechanisms, and for that reason all of them are considered in the study. For MAR and MCAR five imputation methods were used: Random Forest (RF), KNN, SVD, mean and median imputation. For MNAR six imputation approaches were considered: Quantile Regression Imputation of Left-Censored Data (QRILC) (i.e., missing values are imputed through a truncated distribution estimated by a quantile regression), half-minimum (i.e., the missing values are replaced with half of the minimum of the respective feature), zero imputation, RF, KNN and SVD. The experiment was executed with two real-world datasets, one with 997 subjects and 75 features, and the other one with 198 records and 130 features. For both datasets the missing values where generated in proportions from 2.5% to 50%, with a step of 2.5%. The evaluation was performed

through the NRMSE metric for the ignorable mechanisms, and NRMSE-based Sum of Ranks (SOR) for the non-ignorable one (mainly to avoid biased results). For MAR and MCAR the RF outperformed the remaining algorithms, achieving NRMSE values between 0.6 and 0.8, with the second-best method (SVD) achieving values between 0.75 and 0.9. For MNAR the QRILC method outperformed the remaining, achieving SOR values below 100, immediately followed by the half-minimum. Nevertheless, in both scenarios the error increases considerably when the missing rate also increases.

Malla et al. [60] presented a review study about the use of the propensity score method in medical studies, taking in particular consideration how the missing data is addressed in those studies. This approach describes the probability of a patient being assigned to a treatment group when considering the observed covariates. From a collection of 167 studies only 69 report how the issue is handled, where 53 used CCA and 16 multiple imputation approaches. Nevertheless, several of the studies had missing values under MNAR but all neglected to report it. Moreover, the authors claim sensitivity analysis should have been performed in those works, although it was also neglected.

Galán et al. [29] proposed a new approach based on genetic algorithms to perform imputation of missing values in questionnaires. The proposed method uses the standard crossover and mutation operators and also applies elitism, where some of the best genomes are kept from the current generation to the next one. The fitness function considers both the Akaike and Bayesian Information Criterion, two metrics that try to assess the likelihood of a mathematical model being correct, where the first one is more focused on the model error and the second one on its complexity and interpretability. The algorithm starts with a random population and stops when the fitness function shows no improvements in 100 consecutive iterations. The experiment used a public dataset from the Law School Admission Test, considering responses from 1000 students to five questions, and another dataset from the internal medical resident exam performed in Spain, with answers from 500 medical students to five questions. For both cases the amount of missing values generated varied over 10%, 15% and 20% of the observations. The approach was only compared to the MICE algorithm, with the results showing averages of wrong imputations between 31% and 35% for MCAR and between 32% and 53% for MNAR with this method. With the proposed genetic algorithm, the results show averages between 20% and 25% for MCAR and between 21% and 41% for MNAR, surpassing the other method.

Beaulieu-Jones and Moore [8] presented a study where electronic health records data with missing values is imputed with Denoising Autoencoder ((DAE)), being the results compared with the following methods: Iterative SVD, KNN imputation, SoftImpute and mean/median imputation. The experiment was conducted with the public dataset ALS Pooled Resource Open-Access Clinical Trials (PRO-ACT), containing 10723 observations (although only 1824 are used) and 23 features. The results, evaluated through the RMSE metric over missing rates between 10% and 50%, show that the DAE obtains the best

imputation error under the MCAR mechanism, with a difference from the SoftImpute between 0.005 and 0.1. For the MNAR mechanism, the work proved that the DAE also achieves the best results, although very similar to the ones obtained by the remaining methods: the DAE differs from KNN imputation by a minimum of 0.0045 and from SoftImpute by a maximum of 0.0125.

Leacy et al. [47] performed a sensitivity analysis study to understand how the departure from MAR to MNAR influenced the tasks of estimating the prevalence of a partially observed outcome and performing parametric causal mediation analyses with a partially observed mediator. The study used data from a tuberculosis (TB) and human immunodeficiency virus (HIV) prevalence survey that was conducted as part of the Zambia–South Africa TB and AIDS Reduction Study, between 2006 and 2010. The shift from MAR to MNAR was once again performed with the delta-adjustment method integrated in a multiple imputation procedure. Three adjustment values were manually chosen based on the experts opinion and on data from 3 consecutive annual rounds of HIV counseling and testing in the Karonga District of Malawi (2007 to 2010). Each of these values represented different magnitudes of departure from the MAR assumptions. The authors concluded that the estimation of the overall HIV prevalence was considerably different when assuming the MAR or MNAR mechanisms, particularly for strong departures between them.

Gondara and Wang [35] compared the use of DAE for missing data imputation with the MICE, using accuracy to evaluate binary imputation and RMSE to evaluate continuous time imputation. The experiment considered ten simulated and four real-world datasets, with these last ones being: a sample of 1000 acute coronary syndrome patients from the Global Registry of Acute Coronary Events (GRACE); 2323 observations from the European Organization for Research and Treatment of Cancer (EORTC) cancer trial database; a dataset with 861 observations of rehospitalization after surgery in patients with colorectal cancer; and finally a database containing statistics of 52000 unique hard drives. The number of features for these datasets varied between 25 to 50 for the simulated ones and 4 to 8 for the real-world ones. The results show that the DAE consistently outperforms MICE in all datasets, for both metrics, sometimes achieving an improvement over 20%. For the real-world datasets, the difference between both algorithms is between 2.4% and 64.9%. The work only considers the MCAR and MNAR mechanisms, and the imputation error is consistently better for MNAR in the simulated datasets. However, in the real-word ones, the results are mixed for both mechanisms. The authors extended the experiment in a latter work [36], exploring the use of multiple imputation with DAEs and considering two different scenarios of missing data: all the features are subjected to have missing values (uniform synthetic generation) and only half of the features are set to be missing (random synthetic generation). The experiment considered 15 real-world datasets from different contexts, with a number of observations between 101 to 58000 (the majority are small-sized) and 5 to 180 features. The results show that the DAE approach outperforms MICE for all the uniform scenarios and in 7 cases for the random scenario (this can be

seen for 2 datasets under MCAR and for 5 datasets under MNAR). Once again, the results are mixed for the MCAR and MNAR mechanisms.

Garciarena and Santana [32] introduced a statistical significance study focused on the interactions between the missing mechanism, the imputation method and the classification algorithms used with the imputed data. The following imputation methods were considered in the experiments: mean, median, most frequent value, last value carried forward, interpolation, hot deck, expectation-maximization and MICE. The study considers 10 datasets from different contexts, including medical ones (e.g. a diabetic dataset), varying from 326 to 2310 observations and 7 to 41 features. The results regarding the imputation methods are presented by showing the number of times they were significantly better than the others for each missing mechanism, and MNAR displayed the worst and more biased results when compared to MCAR and MAR. Moreover, the expectation-maximization and MICE methods presented the best general results.

Li and Zhou [51] proposed a new imputation strategy that uses the well-known maximum likelihood method for the MCAR and MAR mechanisms, and an estimating equation-based extension of the IPW method for missing data under MNAR. Moreover, for this latter mechanism, sensitivity analysis is also conducted a-priori to evaluate the imputation robustness. To properly validate the new approach a simulation study is presented with 1000 datasets containing a number of 200, 1000 or 3000 observations, and it is compared with a multiple imputation method and CCA. The results are evaluated through the bias and the Monte Carlo standard error. For the MAR mechanism, the error is very similar to all imputation approaches, but the CCA is more biased comparing to the remaining methods. For the MNAR mechanism, the proposed approach presents a higher error comparing to the other methods, but a considerably lower bias. An experiment is also conducted with real-world data, namely the CATIE-AD trial dataset, which contains records from a placebo-controlled trial with 421 Alzheimer's patients that either received atypical antipsychotic drugs or a placebo. The results were similar to the ones from the previous simulation study, and the MNAR imputed values were corroborated by sensitivity analysis, which showed they are robust with a confidence level of 95%.

Jakobsen et al. [41] discussed the use of multiple imputation approaches to handle missing values in randomized clinical trials, and a set of guidelines is presented for this purpose. The authors first present four scenarios where multiple imputation should not be used: when it is possible to ignore the missing values (i.e., perform complete case analysis), when the missing rate is too high, when the analytical model is too complex and when the missing mechanism is MNAR. Nevertheless, when analyzing this last reason, since no proper solutions exist for MNAR, the authors claim multiple imputation can be used if sensitivity analysis is performed. This type of analysis is used to understand the range of uncertainty that can be accepted without creating biased results, and the authors suggest the best-worst and worst-best approach, where at least two cases with assumptions for the

best and worst scenarios of each feature are considered. Moreover, in the case of multiple imputation approaches, regression analysis should also be conducted. The authors also claim that the results under MNAR may always be questionable, but no other appropriate solutions exist to handle it.

Tang and Ishwaran [109] presented a study were 13 variants of the RF algorithm were applied and compared for missing data imputation. This algorithm is suitable for the imputation task because it is capable of handling different types of missing data, scales well to high-dimensions while avoiding overfitting, and takes into consideration the importance of each feature in the data. The experiment considered 60 datasets from different contexts, containing up to 5000 observations and 8500 features. Moreover, the KNN algorithm was also considered in the study because of its speed and its close relationship to RFs. All the three missing mechanisms were evaluated through the relative imputation error, and the results stratified in three levels: low, medium and high correlation between the datasets. In almost all scenarios the missForest variant was the one that presented the best results, which performs the imputation by regressing each feature using the remaining ones as independent variables and generating the missing values for the dependent variable using the trained forest. The error differences between this variant and the remaining ones are quite inconsistent, particularly for the datasets with high correlation, varying from 1% to 25%. Nevertheless, the results are considerable worse for MNAR mechanism, presenting the increasing up to 70% when comparing to MCAR and MAR.

Belger et al. [10] compared the performance of two methods that deal with missing values, namely the CCA and the Multiple Imputation Monte Carlo Markov Chain (MIMCMC), when applied to the GERAS dataset. This database contains records from an 18-month observational study of costs associated with the Alzheimer's disease, and 1488 observations were considered in the experiment. All missing mechanisms are simulated with a missing rate between 10 and 40%, and the results are presented through the bias in mean costs. For the MCAR mechanism the CCA was the best approach, with a maximum bias of 5%. However, for the MAR mechanism, the results were inversed, with the multiple imputation approach outperforming the CCA, with a maximum bias of 5% for missing rates below 40%. Both imputation approaches presented very poor results for MNAR data, with bias values as high as 60%.

Van Kuijk et al. [116] presented an experimental study where a linear regression model is fitted to data containing missing values. The work compares the MICE method with the CCA approach for the imputation task, using a Monte Carlo simulation to determine the impact of all regression parameters in the results. With this strategy, 315 different datasets were considered (most of them having 1000 observations), and all of them were injected with missing values under all the three mechanisms: the MCAR values were chosen randomly; the MAR values where created by correlating two features (x_1 and x_2) and by removing the values of x_1 that are more correlated to the lower and higher values

of x_2; the MNAR values where created by deleting the lower values of a chosen variable. The study considered scenarios with 25%, 50% and 75% of missing ratio. The results are evaluated through the bias of the mean of the regression coefficient estimates and their corresponding standard errors, and they show that the multiple imputation method is the best under MAR and MCAR, but with MNAR the case deletion approach outperforms it. Nevertheless, the authors claim the conclusions from MNAR may not hold for all scenarios, which compromises the generalization of the study.

Valdiviezo and Van Aelst [114] presented a study where six variants of decision tree algorithms are used to perform classification tasks, using incomplete datasets. The authors compare the default behavior that most decision trees use to deal with missing values (surrogate splits) with five imputation approaches: median/mode imputation, proximity matrix (i.e., the imputed values are update iteratively using the proximity matrix calculated by a random forest), KNN, MICE and Multiple Imputation by Sequential Regression Trees (MIST) (i.e., an approach very similar to MICE that uses a decision tree instead of a linear regression to perform the internal estimations of the missing values). The experiment was conducted with five datasets of different medical sub-contexts (breast cancer survival, heart disease, etc.) with simulated missing values under all mechanisms. The imputation evaluation was conducted through a custom metric based on the mean squared prediction error. The results show that in general the multiple imputation approaches (MICE and MIST) are the best, although the worst results are always under the MNAR mechanism, especially when the missing rate is higher than 20%, presenting in some scenarios more than the double of the error.

2.3.2 Discussion

From the analyzed works there are some general conclusions that can be presented about the state-of-the-art approaches for imputation under MNAR:

- MNAR is still the most neglected mechanism. A considerable amount of works related to missing data had to be discarded from this state-of-the-art analysis because they only tackled MCAR and/or MAR. This is particularly worrying when considering that most real-world scenarios tend to have missing values under MNAR;
- From 29 works only 13 report how the missing values were generated. This should be a standard guideline of the missing data works (for all mechanisms) because it is essential to allow the reproducibility of the work and also to understand the validity and context of the missing values (i.e., some authors use very limited approaches for the generation of missing data, which compromises the generalization of the results);
- There are some clear patterns regarding the best methods to use for imputation under MNAR assumptions, namely pattern-mixture models, specific multiple imputation

approaches (e.g., MICE) or Bayesian methods. Nevertheless, the majority of these methods are not valid under the MNAR assumptions, since this missing mechanism can not be modeled from the observed data. Up to now, most authors simply neglect this aspect, which leads to poor and biased results for this mechanism. However, some authors perform sensitivity analysis to study the consistency of the results across a set of different MNAR models. If the results are consistent, one can assume that the imputation model is resilient (the estimated values are less or not biased under MNAR). Nevertheless, this approach does not directly address the problem since it neglects the MNAR nature, and more tailored solutions are clearly required;

- Most works use statistical approaches for the imputation task. With a few isolated exceptions that use KNN, RF or genetic algorithms, the only machine learning method that is more often found in the literature is the DAE, and it presents encouraging results;

- A new trend that was identified in four works with positive results was an attempt to use external information to improve the results under MNAR. Since this mechanism depends of non-observed data, this approach tries to add valuable information that could be correlated to the missing values, and therefore reducing the departure from the MAR mechanism. These works are still preliminary attempts, but the results look quite promising;

- From 29 analyzed works, 16 were from healthcare data contexts. This conclusion comes as no surprise since MNAR is present in multiple medical scenarios (e.g., clinical trials where the participants may be quitting the study for reasons related to the outcome that is being measured [99]) and it has a considerable negative impact in healthcare studies. Nevertheless, it is important to clarify that no context restrictions were applied during the search process of the works under analysis.

Part I

Autoencoder-based Approaches for Missing Data Imputation

This page is intentionally left blank.

Chapter 3

Reviewing Autoencoders: Technical Trends, Applications and Outcomes

Deep learning has received much attention and has been applied to several artificial intelligence problems over the past couple of years. However, its use for imputation purposes remains an understudied topic. Among the deep learning approaches used for imputation, the Autoencoder (AE) and its variants (e.g. Denoising and Variational) recently caught the eye of the research community due to their properties in what concerns the ability of learning from corrupted data, which is a natural extension to the field of missing data [23, 36, 71, 93]. This type of neural network learns a representation of the data from the input layer and tries to reproduce it at the output layer. By doing this, the model is able to learn from incomplete data and generate new plausible values for imputation. Considering the community interest in this method, a comprehensive review that covers the application of AEs to the missing data field addresses a gap in the literature and covers a hot topic for missing data researchers nowadays: the performance of deep learning techniques for data imputation.

This work addresses the use of AEs for missing data imputation by answering the following questions:

- How good is the imputation performed by AEs when compared to other algorithms?
- How do authors tune the AEs' hyperparameters to achieve a better performance?
- In which data contexts have the AEs presented good results?

To answer these questions, 26 works that use AEs for imputation of tabular data were considered, published between 2014 and 2020 (the selection criteria is described below).

The analysis is focused on how the AEs were used, considering aspects such as network structure, hyperparameters tuning, training approaches, extensions of the algorithm, comparison with other methods and data context characterization. The analysis conducted shows that AEs outperform several state-of-the-art methods, and therefore constitute a better alternative to perform missing data imputation.

To select the research studies for the analysis, an extensive search was conducted through the *Web of Science* platform. The articles were searched by title and content keywords, where a combination of the sentence fragments "autoencoders" and "missing data" was applied. Moreover, no time restrictions were set since the goal was to include as much relevant papers as possible. Initially, 54 works were selected with the described search criteria but 33 were discarded for being out of scope (e.g. some were only focused on dimensionality reduction, others were related with generative approaches not applicable for imputation purposes). At the end, 20 works were selected encompassing the period between 2014 and 2019. This search was later extended through the *Google Scholar* platform, using the same criteria already described. To the 20 works already selected, 6 other articles were added, extending the encompassed period to 2020. All these works are focused on tabular data (i.e., structured data stored in table formats, such as relational databases or comma-separated values files). However, 5 additional works that use unstructured data (in this case, images) were also found during the search. Therefore, aiming to provide a more correct and trustworthy analysis, the study is focused on the 26 works that use tabular data, but to avoid ignoring the remaining 5 works an independent analysis is presented in Section 3.4.

The remainder of the chapter is organized in the following way: Section 3.1 presents the theoretical background and technical details related to the hyperparameters and training of the AEs; Section 3.2 a characterization of the datasets used in the works; Section 3.3 the comparison with other methods regarding imputation and classification/regression; Section 3.4 an analysis of the works that use non-tabular data; and Section 3.5 the conclusions with a discussion of the results, recommendations and challenges found throughout the survey.

3.1 Autoencoders

AEs are Artificial Neural Networks (ANNs) trained in an unsupervised way and used to reproduce, as good as possible, the data supplied to the input layer in the output layer. To better understand this type of ANN, its theoretical background is presented in this section. Moreover, the works under analysis revealed tendencies regarding the architecture of the AEs and their training approaches. Theses tendencies can be seen as a standard to be followed, and are also presented in this section, along with extensions to AEs proposed by some authors.

3.1.1 Theoretical Background

AEs are ANNs composed by at least three layers (input, hidden and output layer) which can be divided into two parts: encoder, which goes from the input layer to the output of the hidden layer, and decoder, that goes from the hidden layer to the end of the output layer (Figure 3.1).

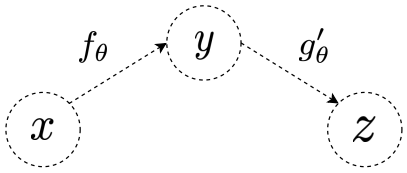

Figure 3.1: Simplified structure of an Autoencoder. f represents the encoder and g' the decoder. x is the input of the network and z is the output. y represents the results of the encoding process.

The encoder part of an AE maps an input vector $\boldsymbol{x}$ to a hidden representation $\boldsymbol{y}$, through a nonlinear transformation $f_\theta(\mathbf{x}) = s(\mathbf{x}\mathbf{W}^T + \mathbf{b})$ where θ represents the weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$. The resulting $\boldsymbol{y}$ representation is then mapped back to a vector $\boldsymbol{z}$ which has the same shape of $\boldsymbol{x}$, where $\boldsymbol{z}$ is equal to $g'_\theta(\boldsymbol{y}) = s(\boldsymbol{W'y} + \boldsymbol{b'})$. The training of an AE consists in optimizing the model parameters ($\boldsymbol{W}$, $\boldsymbol{W'}$, $\boldsymbol{b}$ and $\boldsymbol{b'}$) to minimize the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{z}$ with a Stochastic Gradient Descent variant, using a loss function such as the Mean Squared Error or the Binary Cross-Entropy for binary scenarios [18]. Therefore, this type of network is trained in an unsupervised way to reproduce its input at the output layer.

The AE has one variant often used for missing data imputation: the Denoising Autoencoder (DAE) [118]. This variant is designed to recover noisy data ($\widetilde{\boldsymbol{x}}$), which can exist due to data corruption via some additive mechanism or by the introduction of missing data [18] (Figure 3.2).

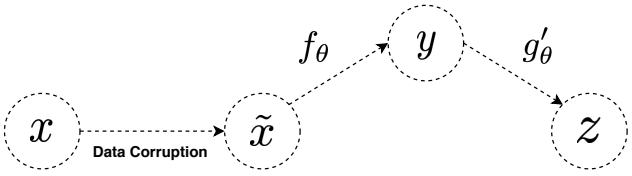

Figure 3.2: Simplified structure of a Denoising Autoencoder. f represents the encoder and g' the decoder. x is the uncorrupted version of the data, $\widetilde{x}$ is the corrupted input of the network and z is the output. y represents the results of the encoding process.

The DAE is similar to a basic AE, but the main difference is the application of a stochastic corruption to the inputs of the model during the training phase, meaning that $\boldsymbol{z}$ becomes a deterministic function of $\tilde{\boldsymbol{x}}$ rather than $\boldsymbol{x}$. One of the possible corruption techniques consists in setting to 0 a fixed amount of features for a set of observations, which can be seen as a dropout on the input layer (dropout is a regularization technique that randomly

drops units from an ANN to avoid overfitting [105]). There are other possible corruption processes, such as adding Gaussian noise or salt-and-pepper noise to the input data [119].

Recently, another variant of the AE family with generative capabilities has been used for missing data imputation: the Variational Autoencoder (VAE). While a vanilla AE learns a compressed representation of the input data, the VAE learns a set of distribution parameters which describe the data, usually the mean and variance of a Gaussian probability function. By sampling from these parameters, the VAE is capable of generating data instances with the same characteristics [44]. The VAE loss function contains two terms: the reconstruction error (as the vanilla AE) and a regularizer (see Equation 3.1, where $q(z|X)$ is the encoder output, $p(X|z)$ is the decoder output, X is the input data and z represents the generated instances). The regularizer used in the second term is the Kullback-Leibler divergence applied to the encoder and decoder distributions. Such regularization is used to ensure the latent space is well structured, leading to similar input data being represented by similar latent spaces [44].

$$L(X) = -E_{z\sim Q(z|X)}[logp(X|z)] + KL(q(z|X) \parallel p(z)) \tag{3.1}$$

Another generative variant of the AE is the Adversarial Autoencoder, which also relies on variational inference but has a different loss computation: it uses the same concept of adversarial loss as in the well-known Generative Adversarial Networks [58].

AEs have two types of representations regarding the number of nodes of the hidden layers: overcomplete, when the hidden layers have more nodes than the input layer, and undercomplete, when the hidden layers are smaller than the input layer. In a multilayer scenario, undercomplete representations are more often found, in which the number of nodes decreases through each encoding layer and increases again through each decoding layer. Such strategy is important to force the network to learn a lower-dimensional representation of the input data. For vanilla AEs, an overcomplete architecture only learns the identity function by copying the input to the output, which creates an overfitting scenario that requires additional handling. However, the DAEs can be overcomplete without having this problem because they compare the original input to its corrupted version. An undercomplete architecture never presents this type of problem since the AE is forced to learn a more concise representation of the input data.

3.1.2 Network Structure

An important aspect of the ANNs structure is the number of hidden layers, since they are directly related with the learning capabilities of the network. The works under study present different values for this parameter, as Figure 3.3 shows (the latent space is considered as a hidden layer in the count). Among the 26 papers, only 5 use a single-layer architecture while the remaining use between 2 and 13 layers. An interesting aspect is

that, with the exception of the work from Zhao et al. [129], only DAEs and VAEs use more than one layer. This parameter is defined in an empirical way for all works, with the exception of the work from El Esawey et al. [27], where a hyperparameter optimization was conducted using the Hyperopt[1] library.

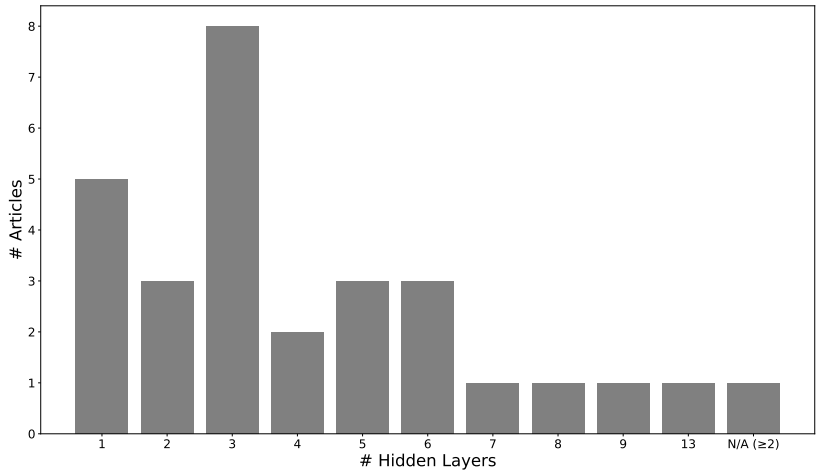

Figure 3.3: Number of hidden layers used in each work. N/A stands for Not Available.

Regarding the number of nodes, only 15 of the 26 works under analysis describe their type of representation, as Figure 3.4 shows. From this subset, 10 use undercomplete representations and 5 use overcomplete, which means that most works force the network to learn only the most relevant data. A more detailed analysis of how the nodes are distributed is presented in Table 3.1. On the multi-layer networks, the overcomplete representations always use a symmetric approach where the number of nodes is increased by a fixed step through the first half of the layers, and decreased by the same step through the second half, using as baseline the input layer. The undercomplete representations often use the same strategy, but they decrease the nodes on the first half of the layers and increase them on the second half. Other approaches are also used for this latter representation, namely a constant number of nodes and different values defined in an empirical way. A hyperparameter optimization approach is also used twice for unknown representations.

The activation functions used by the nodes of the hidden and output layers are reported in 21 of the 26 works. For the standard AEs the Sigmoid function is always used [38, 43, 91], but in the work of Sakurai et al. [91] a custom linear function is used on the output layer. For multi-layer networks the Sigmoid is also used in some works [25, 26, 94, 125], but ReLU is the one more often applied [12, 13, 28, 35, 65, 89, 90, 125], sometimes through the Leaky ReLU variant [89, 90]. The Hyperbolic Tangent is used less often [20, 36], but these works concluded it presents better results than ReLU when the datasets are small. The Softsign function, which is very similar to the Tangent, is also used in the work of Jaques et al. [42].

[1] Available at `http://hyperopt.github.io/hyperopt`.

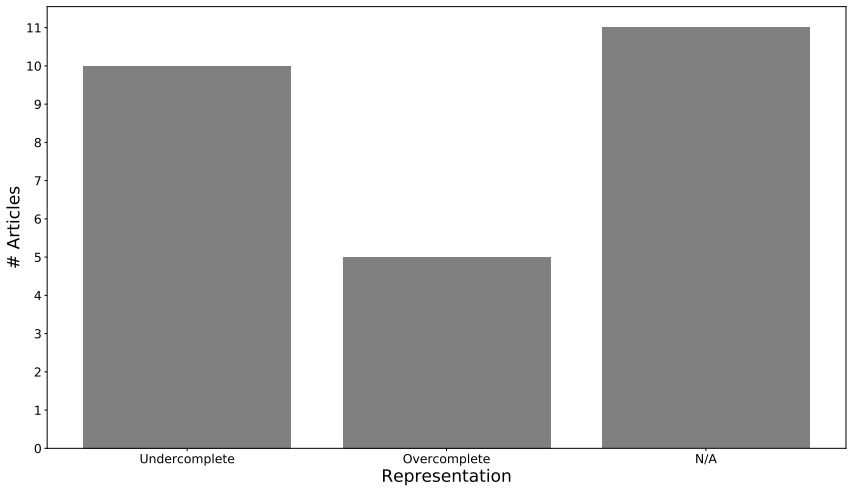

Figure 3.4: Representation used in each work. N/A stands for Not Available.

3.1.3 Training

The training phase of an AE depends on the same aspects of any other ANN: an optimization algorithm, a loss function and the maximum number of epochs. From the 17 works that describe the used optimization algorithm, 5 use the well-known Stochastic Gradient Descent [38, 46, 92, 93, 125] and 6 use one of its variants called Adam [12, 13, 27, 28, 89, 90]. Other algorithms are used less often, namely the Nesterov's Accelerated Gradient [36], the Scaled Conjugate Gradient Algorithm [91] and the RMSProp [65].

Regarding the loss functions, the majority of the works use the standard Mean Squared Error (MSE) [12, 27, 35, 36, 43, 90, 91] or the Squared Error [20, 38, 89, 93]. The exceptions are a work that uses a custom loss function roughly based on the Squared Error [129], 3 works that use the Cross-Entropy [42, 48, 94], another one that uses the Binary Cross-Entropy with a modification to handle the missing values [8], and finally 4 VAE-based works which use the log-likelihood [28, 65, 70, 125]. Notice that all VAE-based works also include the Kullback–Leibler divergence as a second term of the loss function.

With the exception of the modified Binary Cross-Entropy [8], the remaining functions are not able to deal with the missing values as they require all features to be complete. To solve this issue, pre-imputation of the missing values is often performed. Gondara and Wang [36] used the mean/mode method; Sánchez-Morales et al. [92] used and compared the methods zero imputation, kNN imputation (kNN) and Support Vector Machines (SVM) imputation, concluding that the kNN and SVM methods presented the best results; Jia et al. [43] compared constant and linear imputation, the latter performing better in general; Jaques et al. [42] and Saeed et al. [90] imputed all missing values with -1; Ryu et al. [89] replaced all missing values with 0; Sánchez-Morales et al. [94] used and compared the zero imputation method with Multiple Imputation by Chained Equations (MICE), with MICE presenting the overall best results; and finally Sánchez-Morales et al. [93]

Table 3.1: Nodes used in each work, grouped by organizational approach (see Section 3.1.2 for more details). N/A stands for Not Available.

Approach	**Article**	**Nodes**	**Representation**
Symmetric	[35]	Step of 5 over 4 layers	Overcomplete
	[36]	Step of 7 over 5 layers	Overcomplete
	[92]	25% to 75% growth over 3 layers	Overcomplete
	[93]	75% growth over 3 layers	Overcomplete
	[25]	Step of 72 over 3 layers	Undercomplete
	[26]	Step of 72 over 3 layers	Undercomplete
	[42]	Step of 200 over 4 layers	Undercomplete
	[89]	50% growth; latent space of 90	Undercomplete
	[94]	75% growth over 3 layers	Overcomplete
Constant	[65]	Constant 20; latent space of 10	Undercomplete
	[12]	Constant 512; latent space of 100	Undercomplete
	[8]	Constant 500	Undercomplete
	[90]	Constant 128	Undercomplete
Empirical	[125]	Between 3 and 13 over 7 layers	Undercomplete
	[125]	Between 3 and 10 over 5 layers	Undercomplete
	[20]	Between 5 and 36 over 6 layers	Undercomplete
	[28]	Variations over 4 and 6 layers	Undercomplete
	[91]	Between 4 and 70	Single-layer
	[38]	Between 200 and 400	Single-layer
	[48]	Between 200 and 400	Single-layer
Optimization	[27]	Between 240 and 370 over 5 layers	N/A
	[46]	Between 5 and 30 over 1 or 2 layers	N/A

applied and compared the methods Multi-layer Perceptron imputation (MLP), Singular Value Decomposition (SVD) and MICE, being the results generally better when MICE was used.

Some works also define a maximum number for the training epochs: less than 50 [28, 90], 100 [8], 300 [38], 500 [36], 1000 [35, 91], 2000 [70] and 10000 [46, 65]. Although early stopping rules are common on ANNs training, they were only applied by Gondara and Wang [36], stopping the training when the MSE achieves $1 * 10^{-6}$ or no improvement exists in an average of 5 epochs.

As previously stated, avoiding overfitting of AEs is mandatory, particularly with over-

complete representations, otherwise the network will lose its generalization ability. To avoid this behavior, the objective function can be modified to include a regularization term. The one used more often is the L2 regularization, which is also known as Frobenius norm regularization or "weight decay", because it forces the weights to decay towards zero without achieving it [100]. The L2 term consists of the sum of the squared values of the weights that is multiplied by an attenuation coefficient, which results on a higher error for larger weights, causing the training algorithm to favor smaller weights. Considering all research works, 7 use this approach [12, 13, 36, 38, 42, 90, 129] with the coefficient values varying between 0.01 and 0.001. Gondara and Wang [35] used batch normalization, where the output of each node of the hidden layers is normalized using the mean and variance of the batch, and 20% dropout in each hidden layer, meaning that 20% of the nodes are randomly set to 0 (they are dropped, becoming inactive) in each layer. This latter approach is also used by Beaulieu-Jones and Moore [8] and Saeed et al. [90]. Nazabal et al. [70] also applied batch normalization, but with a mean of 0 and a variance of 1. An interesting observation is that, in theory, DAEs can be overcomplete without the need for any regularization since they compare the original data with the corrupted version. Nevertheless, several works that use this type of AE still apply it [35, 36, 42].

A common procedure applied before the training step is to normalize the input data [2]. This normalization is known to provide several improvements: the training is often faster, which happens as a consequence of the faster convergence of the weights from the networks, and it also reduces the chances of the training being stopped on a local minimum. Only 8 works describe how this question was addressed, being the normalization output between $[0, 1]$ [12, 13, 36, 42, 90, 94] and $[-1, 1]$ with zero mean [20]. Xie et al. [125] also normalized the data but the scale is unknown.

In the specific case of DAEs, an important aspect is the type of noise added to the input data. Although such data already contains missing values, some works reinforce the data corruption by adding additional noise. Among all works, 14 use DAEs and 7 report the type of noise additionally generated, which is always dropout. As described, dropout is used to avoid overfitting by selecting a random percentage of nodes in each hidden layer and randomly setting them to 0 (i.e., making them inactive). However, this concept can also be seen as a type of noise when it is applied to the input layer, since setting several of the input features values to 0 is a corruption of the data, and forces the network to learn with that level of incompleteness. Some works propose different percentages for this random selection, such as 5% [42], 20% [8, 35], 30% [27] (although this work shows that using values between 10% and 60% produces equally good results) and, finally, 50% [36]. The work from Sánchez-Morales et al. [94] also uses additive Gaussian noise with zero mean and a dynamic variance that is a percentage of the variance for each feature. The authors concluded that this percentage should be between 10% and 20% to achieve the best results.

A layer-wise unsupervised pre-training strategy is used by 7 of the works under analysis [25, 26, 27, 72, 92, 94, 129]. In this approach, the representation of the k^{th} layer is the input for $(k+1)^{th}$ layer, which is trained after the k^{th} layer. When k is trained, it will have as input the uncorrupted output from the previous layers. After some training, the fine-tuning will be performed, as the current network parameters will be used to initialize a new network that will be trained using a regular supervised training criterion. In theory, this strategy improves the initial solution for the optimization problem solved during the training procedure, since the weights are no longer randomly initialized. The approach was proposed by Vincent et al. [119], and is called Stacked Autoencoder.

3.1.4 Extensions

Although AEs can be used without any modification for missing data imputation, several extensions have been proposed on the reviewed works, introducing changes or including them only in part of the process. In this section, such extensions are described individually. Considering the diversity of strategies followed by the authors, it is not possible to aggregate the analysis in similar ways to the remaining parts of this study.

Gondara and Wang [36] used a multiple imputation approach, where several runs of the model are executed with different initial weights of the network, defined randomly. The approach results in several datasets with different imputed values, attenuating the variability of the model. The different datasets can be analyzed and combined through the use of the mean value for numeric features or a voting mechanism for categorical ones.

Duan et al. [26] generated individual DAE models for different sources of data but trained each model with all sources. However, the respective data source of each model must have a bigger impact on the data imputation. To ensure this, a hierarchical training approach is proposed where each model is trained a second time only with the data from that source, being the network's weights influenced twice by it.

El Esawey et al. [27] performed imputation on missing temporal information, which requires strategies to ensure that the network is able to deal with the time-series data. The proposed approach is to include recurrent connections on a DAE, transforming it into a recurrent network instead of a feedforward one, because this type of network is commonly used in scenarios where temporal relations exist in the data. As a consequence of this change, the training algorithm had to be adjusted to the Backpropagation Through Time algorithm. Sakurai et al. [91] also dealt with time-series data, but used a different approach. A window of n days is defined and multiple input matrices are created from the training data by shifting the window day by day. The resulting matrices are finally combined in a single one through a composite transformation operation, which becomes the network input for training. Jia et al. [43] dealt with spatio-temporal data through the application of a standard AE for the spatial features and a Long Short-Term Memory

(LSTM) autoencoder for the temporal ones. The latter is a direct adaptation of the well-known LSTM model to incorporate the AE characteristics. Both AEs are treated as one, where the output of the regular AE serves as input for the LSTM. Fortuin et al. [28] used a VAE to map time-series data with missing values to a complete latent space, and the time-series are then modeled with a gaussian process using this latent representation instead of the original data. Comparing to existent approaches, this method is suitable for multivariate scenarios because it accounts for the correlation between different channels.

Zhao et al. [129] applied an AE together with the Fast Clustering algorithm to enhance the clustering accuracy, which increased almost the double. The imputation is made by the Top k-Nearest Neighbor Hybrid Distance Weighted algorithm.

Jaques et al. [42] used a DAE to deal with the missing data issue but also as a classifier by adding 2 new layers at the end of the network, which are used for prediction purposes. Saeed et al. [90] proposed a similar approach for the same purpose.

Chen et al. [20] used Dynamic Movement Primitives (DMPs) to generate new humanoid movements, although a dimensional reduction of the input data is required to handle the context appropriately. A DAE is used for this purpose and some tests of imputation of missing humanoid joints are conducted.

Sánchez-Morales et al. [93] used a DAE for imputation purposes but applied a deletion strategy before the training that forces some new missing values on observations that already have missing data. By doing this, the network should learn better how to reconstruct incomplete data. However, the dataset becomes unbalanced after this deletion, and for that reason a compensation strategy is also applied. This compensation is achieved through a small change on the loss function that includes a balancing parameter that is applied to the error of the complete and incomplete observations.

Lee and Lee [48] proposed a new collaborative filtering approach based on AEs that is able to perform the recommendation of the top N items using feedback data from users. The approach has 2 steps: a first one where a regular AE is trained to impute the missing values with positive feedback and a second one where a DAE is trained with the previous imputed data and used to perform the recommendation tasks. Hau et al. [38] introduced a similar but simpler approach, where an AE is trained and used without changes for recommendation purposes.

Nazabal et al. [70] proposed a VAE adaptation to handle heterogenous data by modeling different data types with different likelihood models appropriate for the respective types. To allow for different likelihoods, each feature has its own independent neural network. To account for relations and dependencies between features, a hierarchical structure is used to share network parameters among the different dimensions. Such approach addresses the limitation of a vanilla VAE modeling all data through a Gaussian distribution, which is not the most appropriate solution for non-real-valued data.

Sánchez-Morales et al. [94] proposed a multitask learning approach where a DAE is modified to perform the imputation of missing values while it simultaneously solves a classification task. The target of the classification problem is supplied as an additional feature to the DAE, and the loss function uses a double weighting approach between the classification and the imputation errors (such weights must be empirically adjusted for different datasets).

Lai et al. [46] introduced the concept of a tracking-removed AE (TRAE), which removes the connection between each output neuron and its corresponding input neuron. Such technique weakens the self-trackability of the network and helps it learn better the relations between different features of the dataset. Moreover, the authors also proposed a training scheme that includes the missing values in the training procedure. The approach uses the missing values estimates of the k iteration in the input of the $k+1$ iteration, which allows the training to consider the entire dataset while the estimates improve over time. Pre-imputation is still required, but since its impact is not significant the authors use a random number within the features domain.

3.2 Datasets Characterization

All the 26 analyzed works describe the datasets used in the experiments, as Table 3.2 summarizes. This table presents a description of the datasets used in each work, containing their number of instances and attributes, the missing rates applied and if they are public or private.

Table 3.2: Datasets used in each work. N/A stands for Not Available.

Article(s)	**Public**	**Dataset(s)**	**# Instances**	**# Features**	**Missing Rates (%)**
[35]	Yes	10 synthetic (5 with single outcome and 5 with multiple outcome) and 4 real-world	17703 to 40382 for the synthetic and 861 to 52000 for the real-world	25 to 50 for the synthetic and 4 to 8 for the real-world	60, 80
[36]	Yes	15 real-world	101 to 58000 (the majority are small-sized)	5 to 180	20
[8]	Yes	ALS Pooled Resource Open-Access Clinical Trials (PRO-ACT)	10723 (but only 1824 are used)	23	10, 20, 30, 40, 50
[25]	Yes	Caltrans Performance Measurement System (PeMS)	250	288	1, 10, 20, 30, 40, 50, 60, 70, 80, 90

Article(s)	Public	Dataset(s)	# Instances	# Features	Missing Rates (%)
[26]	Yes	Caltrans Performance Measurement System (PeMS)	104544 (data of weekdays and non-weekdays is separated through K-means Clustering)	288	5, 10, 15, 20, 25, 30, 35, 40, 45, 50
[92]	Yes	Cloud Dataset, Blood Transfusion Service Center and Boston Housing (all from UCI)	1024, 683 and 506 (respectively)	10, 4 and 13 (respectively)	10, 20, 30
[27]	Yes	City of Vancouver data of bicycle traffic, for 22 counter locations over 3 years	12986	2	N/A
[129]	Yes	5 academic from UCI (Iris, Wine, Pima, Yeast and Housing) and air quality monitoring from China	150 to 1484	4 to 14	3, 6, 9, 12, 15
[38]	Yes	Real-world Web Service QoS	1974675	2	N/A
[91]	No	Real time series showcase data of one month	N/A	N/A	N/A
[43]	Yes	EEG data extracted from UCI	600	16384	30
[42]	No	Mood data from SNAPSHOT project	6180	343	30
[20]	Yes	CMU Graphics Lab Motion Capture Database	5	50	N/A
[72]	No	Quality inspection data collected from social networks and e-commerce websites	N/A	N/A	1, 5, 10, 15, 20, 25, 30
[93]	Yes	Magic Gamma Telescope, Pima Indians Diabetes, Sensorless Drive Diagnosis, Gas Sensor Array Drift, Activity Recognition system based on Multisensor data fusion (AReM) and Twonorm	768 to 58509	6 to 128	10, 20, 30
[48]	Yes	MovieLens (ML-100K and ML-1M)	100000 and 1M	N/A	N/A

Article(s)	**Public**	**Dataset(s)**	**# Instances**	**# Features**	**Missing Rates (%)**
[89]	No	Real-world residential customers from South Korea, collected over 360 days (March 1, 2016, to February 23, 2017)	520200	96	5, 10, 30, 50
[65]	No	Simulated Milling Circuit	104	96	20, 90
[12]	Yes	Caltrans Performance Measurement System (PeMS)	210432	288	10, 20, 40
[13]	Yes	Caltrans Performance Measurement System (PeMS)	210432	288	10, 20, 40
[125]	No	Industrial data collected from the Distributed Control System of a polyester plant in China	10000	15	10, 30, 50
[90]	Yes	ExtraSensory	$\approx$ 300000	166	5
[70]	Yes	6 academic from UCI (Adult, Breast, DefaultCredit, Letter, Spam and Wine)	699 to 32561	10 to 58	10, 50
[28]	Yes	2012 Physionet Challenge real-world medical data	4000	37	60
[94]	Yes	Activity Recognition system based on Multisensor data fusion (AREM), Activity Recognition from Single Chest-Mounted Accelerometer (CHEST), Sensorless Drive Diagnosis (DRIVE), Rectangles (RECT), Skin Segmentation (SKIN) and Sloan Digital Sky Survey RD14 (SKY)	10000 to 1926896	3 to 784	20, 50
[46]	Yes	Iris, Leaf, Friedman, Concrete Slump Test, Stock portfolio performance, Seeds, Cloud, Glass, Yacht and Vertebral Column	103 to 1200	4 to 16	5, 10, 15, 20, 25, 30

The data contexts are diverse but some areas can be identified, namely medical and human-related data [8, 20, 28, 42, 43, 90], quality of service [38, 72, 91, 129], traffic data [12, 13, 25, 26, 27] and automation systems [65, 89, 125]. The remaining works use synthetic and real-world datasets from miscellaneous contexts, the majority available at public repositories such as the UCI Machine Learning.

The missing data mechanism is rarely a point explored by the authors. Among the 26 articles in analysis, only 9 identify the type of missing data that is being used. The articles from Beaulieu-Jones and Moore, Boquet et al., and Gondara and Wang [8, 12, 13, 35, 36] address the MCAR and MNAR mechanisms, while the papers from Duan et al., McCoy et al., and Sánchez-Morales et al. [25, 65, 92, 93] address only MCAR. The MAR mechanism is not stated by any of the works.

3.3 Comparison and Evaluation

To evaluate the imputation results of the AEs, the works under analysis frequently compare them with other imputation algorithms. The evaluation metric most frequently used is the Root Mean Square Error (RMSE), applied in about 50% of the articles, as Figure 3.5 shows. AEs are also compared with other methods in what concerns their impact on the performance of classification and regression tasks. In this scenario, accuracy is the metric more often used, as shown in Figure 3.6.

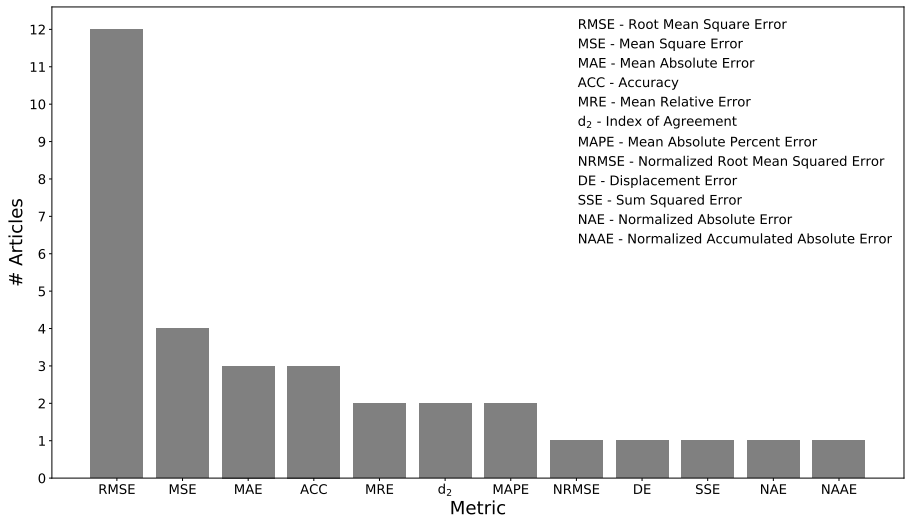

Figure 3.5: Metrics used for the evaluation of imputation tasks.

In both comparative scenarios, the data needs to be divided into training and test sets. Considering the 26 works, only 11 report how this task is performed, and 8 use hold-out validation while only 2 use cross-validation. The works that apply hold-out validation use similar divisions, such as 65%-35% [42], 66.666%-33.333% [25], 70%-30% [27, 36, 94], 80%-20% [46, 92, 93, 125] and 90%-10% [91], for training and test partitions, respectively.

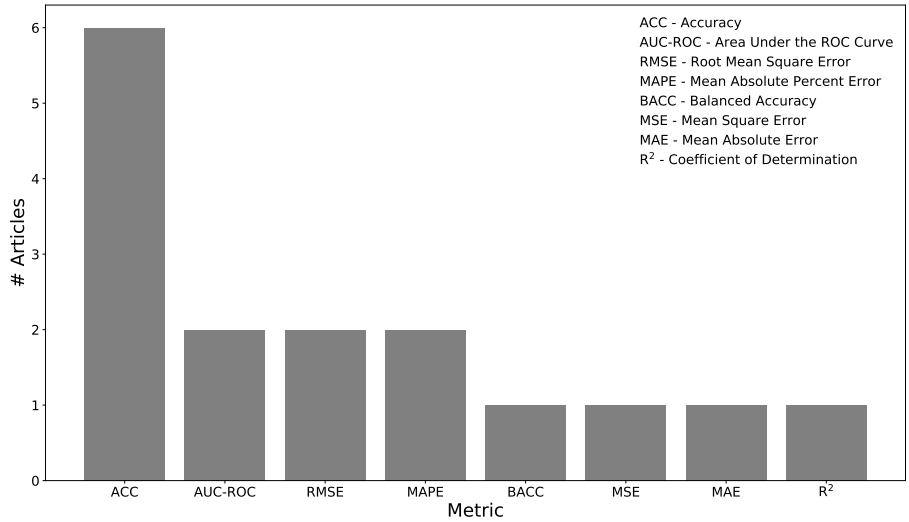

Figure 3.6: Metrics used for the evaluation of classification and regression tasks.

The works from Jia et al. [43] and Saeed et al. [90] use k-fold cross validation with $k = 5$. Some of these works also obtain average results over 10 experimental runs of the respective validation strategy [43, 91].

3.3.1 Imputation

AEs were compared in several works with other methods for imputation, and presented better results in most scenarios. In this section an analysis of these results is presented. The discussion is individualized for the Denoising and Variational AE variants. Vanilla AEs are left out since no works use them directly for imputation.

To summarize the results of the works described for each variant, a taxonomy was created where the algorithms from the studies are grouped by their families. The comparison is made through a scale between 1 and 4, where each value has the following meaning: 1 means the AE presents worse results; 2 means the AE presents equal results; 3 means the AE presents marginally better results; and 4 means the AE presents better results.

3.3.1.1 Denoising Autoencoders

Gondara and Wang [35] compared the use of a DAE with the Multiple Imputation by Chained Equations (MICE), using accuracy to evaluate binary imputation and RMSE to evaluate continuous time imputation, and concluded that the DAE consistently outperforms MICE in all datasets, for both metrics, sometimes achieving an improvement over 20%. For example, the DAE outperforms MICE with a minimum and maximum difference of 2.4% and 64.9%, respectively, for real-world datasets. The work of Gondara and Wang [36] is very similar to the latter, yet considers two different scenarios of missing data: all

the features are subjected to have missing values (uniform synthetic generation) and only half of the features are set to be missing (random synthetic generation) [96]. The results show that the DAE approach outperforms MICE for all the uniform scenarios and in 7 cases for the random scenario (this can be seen for 2 datasets under MCAR and for 5 datasets under MNAR).

Beaulieu-Jones and Moore [8] compared the DAE with the Iterative Singular Value Decomposition (SVD), the k-Nearest Neighbours (kNN) imputation, the SoftImpute and the mean/mode imputation, using RMSE for evaluation. The work shows that the DAE obtains the best results under the MCAR mechanism, with a minimum and maximum difference from the second best method (SoftImpute) of 0.005 (for a missing rate of 50%) and 0.1 (for a missing rate of 30%). For the MNAR mechanism, the work proved that the DAE achieves the best results, although very similar to the ones obtained by kNN imputation, Softimpute and SVD: the DAE differs from the second best method by a minimum of 0.0045 (kNN imputation under a missing rate of 20%) and a maximum of 0.0125 (SoftImpute under a missing rate of 40%).

Duan et al. [25] compared the DAE with a vanilla ANN, and the RMSE values vary between 16.9 and 20.3 for the DAE and between 17 and 21 for the ANN. The DAE also proves to be a better imputation method than ANN regarding the MAE metric. The work was later extended [26] to compare the same approach with ANNs, the History Model and the Autoregressive Integrated Moving-Average (ARIMA). Again, the DAE outperforms the remaining models: in terms of RMSE, the values range from 13.5 to 14.9, while for ANNs the range is between 15.2 and 17.5.

Ning et al. [72] compared the DAE with variants of the kNN imputation, namely the Weighted k-Nearest Neighbours data filling algorithm based on Grey correlation analysis (GBWkNN) and the Mutual k-Nearest Neighbours Imputation (MkNNI). The DAE surpasses the remaining imputation strategies, followed by GBWkNN, and the RMSE metric ranges from 13.4 to 14.5 and 15.2 to 17.5 for these two approaches, respectively. A time complexity analysis study is also conducted and the DAE is once again the one with the best results, presenting an average running time of 24.1 seconds against 30.7 for GBWkNN and 35.7 for MkNNI.

Sánchez-Morales et al. [92] used the DAE to improve the final imputation estimates of datasets pre-imputed with the algorithms zero, kNN and SVM imputation, achieving good results as the quality of imputation increases from 17% to 96% in all studied datasets.

Jaques et al. [42] compared the DAE with Principal Component Analysis (PCA) for multimodal data imputation, and the results show lower RMSE values for DAEs in all scenarios, which was expected considering that PCA works only in a linear space, performing worse with more complex data.

Sánchez-Morales et al. [93] compared the DAE and its variant with deletion and compen-

sation against the methods Multilayer Perceptron Imputation (MLP), SVD and MICE. The results show that the DAE with deletion and compensation and the MICE algorithms consistently outperform the remaining, presenting similar imputation squared errors for all the datasets, being the maximum difference 0.09.

Ryu et al. [89] compared the DAE with Linear Interpolation and the Historical Average (i.e., the average of highly correlated periods), while applying a binary mask vector to the DAE loss in order to give more weight on the accuracy of the imputed missing values. The Linear Interpolation method presents best results when the missing values are randomly generated (presumably a MCAR scenario), but when averaging over all missing data generation mechanisms the DAE shows better RMSE by up to 28.9% for point-wise error and by up to 56% for accumulated error.

A summary of the results obtained with DAEs in the reviewed works is presented in Table 3.3. The following families of algorithms were considered: Statistical, Matrix Completion, Distance and Connectionist.

Table 3.3: Comparison of DAEs with other methods. Score is a value between 1 and 4: 1 means the DAE presents worse results; 2 means the DAE presents equal results; 3 means the DAE presents marginally better results; and 4 means the DAE presents better results.

Algorithms' Family	**Algorithm & Article**	**Score**
Statistical	Multiple Imputation by Chained Equations [35]	4
	Multiple Imputation by Chained Equations [36]	3
	Multiple Imputation by Chained Equations [93]	2
	Mean/Median imputation [8]	4
	Autoregressive Integrated Moving-Average [26]	4
	Principal Component Analysis [42]	4
	Linear Interpolation [89]	3
	Historical Average [89]	4
Matrix Completion	Iterative Singular Value Decomposition [8]	3
	Singular Value Decomposition [93]	3
	SoftImpute [8]	3
Distance	k-Nearest Neighbours [8]	3
	History Model [26]	4
	Weighted kNNs based on Grey Correlation Analysis [72]	3
Connectionist	Artificial Neural Networks [25]	3
	Multilayer Perceptron Imputation [93]	3
	Artificial Neural Networks [26]	4

3.3.1.2 Variational Autoencoders

McCoy et al. [65] compared a VAE with the PCA method (applied in a multiple imputation scenario) and the mean imputation. The VAE outperformed the remaining methods, with average improvements of 45% over the PCA and 46% over the mean approach.

Saeed et al. [90] compared a (AAE) (which is not a VAE but is also based on variational inference) with the PCA method (mean, median and constant imputation with -1 were also considered, but the results were not reported). The AAE outperformed the PCA, achieving a RMSE of 0.227 compared with 0.937 for the latter method.

Nazabal et al. [70] compared their VAE extension that is capable of handling different data types with mean imputation, MICE, a general latent feature model for heterogeneous data and a Generative Adversarial Network for imputation. The proposed approach is outperformed in most scenarios with real-valued variables, but shows promising results with categorical variables. In this latter case, it outperforms the remaining methods in 4 of the 6 datasets used, with an error decrease over 50% for some settings. Therefore, the method appears to be more suitable for datasets that mostly contain categorical features.

A summary of the results obtained with VAEs in the reviewed works is presented in Table 3.4. The families of algorithms considered for this variant were: Statistical and Connectionist.

Table 3.4: Comparison of VAEs with other methods. Score is a value between 1 and 4: 1 means the VAE presents worse results; 2 means the VAE presents equal results; 3 means the VAE presents marginally better results; and 4 means the VAE presents better results.

Algorithms' Family	**Algorithm & Article**	**Score**
Statistical	Principal Component Analysis [65]	4
	Mean Imputation [65]	4
	Principal Component Analysis [90]	4
	Multiple Imputation by Chained Equations [70]	3
	Mean Imputation [70]	4
	General Latent Feature Model [70]	2
Connectionist	Generative Adversarial Network [70]	3

3.3.2 Classification and Regression

When considering the impact on classification and regression tasks, 10 of the works under analysis presented experimental studies that try to assess the impact of data imputed with AEs in such tasks.

Gondara and Wang [36] showed that the accuracy of the Random Forest classifier is higher when MNAR data is imputed with a DAE, rather than with MICE. The accuracy improvement varies from less than 1% to 20%, depending on the dataset.

Jia et al. [43] used a spatio-temporal AE with an adapted LSTM to impute missing values and learn a compact representation of the data, testing the impact of this pre-processing step on SVMs, Decision Trees (DTs) and Convolutional Neural Networks (CNNs) classifiers, using the AUC-ROC and accuracy metrics for evaluation. The experiments showed that both metrics present better results when the AE is used, with accuracy improvements between 4% and 13% for all classification algorithms.

Jaques et al. [42] connected additional classification layers to a DAE, which turns it into a classifier after the encoding part, and compared its accuracy results with a SVM, a Logistic Regression (LR) and a Feedforward ANN, using different imputation methods like PCA or filling the missing values with -1. The experiments showed that this new approach produces very similar results to the remaining algorithms, with varying accuracy results, although always between 58% and 64%. Saeed et al. [90] followed the same approach with an AAE, but only compared the method with different imputation approaches (mean, median, constant imputation with -1 and PCA). The results showed that the AAE achieves the best balanced accuracy, although the improvement is not significant (smaller than 5% in most settings) when compared to the mean and median imputations.

Sánchez-Morales et al. [93] trained a linear classifier with the imputed data from the DAE, MICE and SVD algorithms. The best results were obtained with the DAE, showing accuracy values almost always above 90% and with 1 to 8% differences from the remaining imputation methods.

Nazabal et al. [70] compared their VAE extension that is suitable for heterogeneous data with a deep logistic regression model and a conditional VAE. The obtained results showed similar accuracies between all imputation strategies, with the proposed method outperforming the remaining only in 2 of 5 datasets, and by a very narrow difference (smaller than 0.1).

Fortuin et al. [28] trained a logistic regression with data imputed through different methods: a VAE extension proposed in the work, forward and mean imputation, a simple gaussian process, a vanilla VAE, the VAE method from Nazabal et al. [70], and two recurrent neural network-based methods (GRUI-GAN and BRITS). The proposed VAE approach provided the best AUC-ROC results, but the improvement is insignificant since

the differences between the methods are minor (in general smaller than 0.05).

Boquet et al. [12] trained a 2-layer MLP with data imputed through a VAE, a vanilla Autoencoder and the PCA method. The obtained results show the VAE outperformed the remaining methods, providing improvements in the regression RMSE of at least 40% for MNAR values and 17% for MCAR values. Boquet et al. [13] extended the latter study and presented the same results and conclusions.

Xie et al. [125] proposed an approach which combines 2 VAEs to perform regression tasks, and is able to automatically deal with missing values. The approach was compared with deletion of the missing values, mean imputation and the PCA method, all combined with a vanilla VAE for the regression. The proposed approach outperformed the remaining for all settings, achieving MSE values below 0.05 for all missing rates. However, the differences between the methods are small, considering that the worst MSE value obtained was under 0.2 with the PCA method.

3.4 Autoencoders for Non-Tabular Data

As previously stated, this study mostly focuses its analysis on 26 works that use tabular data. Nevertheless, during the selection of the articles, 5 works that use non-tabular data were also found (in this case, they all use 2D images). For the sake of a fair and proper comparison between works, they were not considered in the remaining sections of this work. However, to avoid neglecting them, they are discussed in this section.

From the 5 works, 4 address imputation of missing data modalities with a vanilla AE. The key idea is that information about a feature can be obtained from different sources, and by combining them the feature will have more information.

Shao et al. [103] proposed the use of an AE to deal with missing modalities on image classification. Two approaches were introduced: the first mixes all the data into one AE whereas the second uses a bagging strategy to combine n AEs (each trained with a part of the data) and uses a Sparse Low-Rank Feature Fusion approach to refine the results from the multiple AEs. The authors focus the evaluation on an image classification problem, comparing the proposed approach to 4 methods: the Transfer Subspace Learning (TSL), the Low-rank Transfer Subspace Learning (LTSL), the Robust Domain Adaptation with Low-rank Reconstruction (RDALR) and the Geodesic Flow Kernel (GFK). The results showed that the new approach is better in all scenarios, with an average accuracy between 73.47% and 89.83%, while the remaining algorithms never surpassed the 70% threshold.

Malek et al. [59] proposed two similar approaches for the recovery of missing parts of multispectral images, which can be seen as different modalities of the same image. The first approach uses a vanilla AE that receives as input the images' pixels, referred to as pixel-based reconstruction. The second approach defines a central pixel and creates n

patches using a grid and changing the position of that pixel on the grid in each patch. This latter approach uses n AEs, one for each patch, and their results are fused through a weighted average. The method was compared to the Basis Pursuit, the Orthogonal Matching Pursuit and Genetic Algorithms. The conclusions showed that the proposed approach outperforms the remaining algorithms for all datasets, achieving Peak Signal-to-Noise Ratio (PSNR) values between 28.47 and 43.94, greater than the remaining by an average of 6.

Shang et al. [102] used a DAE, although only for the fusion of the modalities, whereas the imputation is performed afterwards with Generative Adversarial Networks. The AEs use fully connected layers for the imputation of numeric data and convolutional layers for imputing images. The approach was compared with the a matrix completion method, a vanilla AE, the pix2pix and the CycleGAN. The results showed that the proposed approach outperforms the remaining algorithms for all datasets, presenting an average RMSE error of 3.84 for the MNIST dataset and an average accuracy of 80.03% and 64.50% for each of the remaining datasets.

Tran et al. [111] followed an approach similar to Shang et al. [102] regarding the types of layers for numeric data and images, although Residual Autoencoders (RAEs) are used instead. These are very similar to DAEs, with the difference of the output layer. DAEs produce a replica of the input data without noise while RAEs produce the difference between the input data with and without noise. By taking advantage of this difference, a cascade architecture is proposed instead of the common stacked one. While the standard approach adds layers to the AEs, the cascade model uses independent RAEs and stacks the entire networks. Moreover, it uses a joint learning scheme where the minimization of the loss function during the training stage is done with an overall strategy to all RAEs. The approach was compared with the Singular Value Thresholding, the SoftImpute, the OptSpace, Genetic Algorithms, DAEs and other AE variants. The results showed that stacking RAEs to build a deep architecture improves the imputation, particularly when convolutional layers are used, since the proposed approach outperforms the remaining methods in all datasets, with the resulting PSNR values being between 26.12 and 31.04, greater than the remaining by an average of 1.5. The DAE methods have close errors to this new approach, but the authors claim it can recover more individual characteristics of the data.

Ma et al. [57] is the only work from the 5 that does not address missing modalities in images. Instead, the authors proposed a Partial VAE for imputation of images, which is similar to a vanilla VAE but is trained only with the observable data (i.e., complete data). The method was compared with two variants of a vanilla VAE pre-imputed with zeros (one is standard and the other one uses a mask matrix indicating which variables are complete), and the experiments used the MNIST dataset injected with missing pixels randomly selected (rates vary between 1% and 70%) and removed from the upper 60%

region of the images in the test set. The approach outperformed the remaining methods, showing bigger improvements when an entire region of the image is missing (an average improvement of approximately 10%).

A summary of the image datasets used in these 5 works, similar to the one presented before for the tabular datasets, is available in Table 3.5. Moreover, a summary of the results obtained from the 3 works that evaluate the imputation quality is presented in Table 3.6. This summary uses the same taxonomy introduced previously for the works using tabular data. Note that the families marked with △ were compared with the Convolutional Residual AE, the ones marked with □ with vanilla AEs and the one marked with ‖ with a VAE.

Table 3.5: Image datasets used in each work. N/A stands for Not Available.

Article(s)	**Public**	**Dataset(s)**	**# Instances**	**# Features**	**Missing Rates (%)**
[103]	Yes	BUAA and Oulu-CASIA images databases	15 and 80 (respectively)	2 image modalities (NIR and VIS)	N/A
[111]	Yes	2013 GRSS Data Fusion Contest, RGB-D Object, Multi-PIE and the Hyperspectral Face from Hong Kong Polytechnic University	200, 683, 2258 and 114 (respectively)	2 modalities with 111 and 37, 2 modalities with 2500, 5 modalities with 1024 and 24 modalities with 625 (respectively)	40, 45, 50
[59]	No	2 synthetic from the Taiwanese FORMOSAT-2 and the French SPOT-5 satellites, and 1 real from the European Sentinel-2 satellite	N/A	160000, 202500 and 560000 (respectively)	N/A
[102]	Yes	MNIST and 2 databases with information about patients of substance use disorders	70000 and 12158 (respectively)	784 for the MNIST and N/A for the remaining	N/A
[57]	Yes	MNIST	70000	784	1 to 70

Table 3.6: Comparison of AEs with other methods for imputation in images. Score is a value between 1 and 4: 1 means the AE presents worse results; 2 means the AE presents equal results; 3 means the AE presents marginally better results; and 4 means the AE presents better results.

Algorithms' Family	**Algorithm & Article**	**Score**
Matrix Completion$^{\triangle}$	Singular Value Thresholding [111]	4
	SoftImpute [111]	4
	OptSpace [111]	4
Evolutionary$^{\triangle\square}$	Genetic Algorithms (GA) [111]	4
	Genetic Algorithms [59]	4
Connectionist$^{\triangle\parallel}$	Denoising Autoencoder [111]	4
	Stacked Denoising Autoencoder [111]	4
	Multi-modal Autoencoder [111]	4
	Deep Canonically Correlated Autoencoders [111]	3
	Variational Autoencoder [57]	4
Other$^{\square}$	Orthogonal Matching Pursuit [59]	4
	Basis Pursuit [59]	4

From these 5 works it is possible to draw some initial conclusions about the use of AEs to impute missing parts of images: the results are encouraging, especially when AEs use convolutional layers. Furthermore, the application of this method to multimodal data also seems promising, particularly since the AEs can perform the necessary data fusion steps in a direct way and with a small effort.

Driven by the encouraging results for non-tabular data, we decided to explore the use of VAEs for missing image data imputation in a new work. In this work we propose a variant of the VAE, called Variational Autoencoder with Weighted Loss (VAE-WL), which has a custom loss function that prioritizes the reconstruction of the missing values. Using this method we tackled the imputation of missing values under MCAR in images. We compared our approach to other state-of-the-art generative methods and the results show clear improvements in the imputation quality, achieving in several settings error improvements above 40%.

Although a VAE can be used in its original form to perform missing data imputation tasks, its loss function is not the most suitable for this purpose [65]. As Equation 3.2 shows, the

default VAE loss function contains two terms: the first is the reconstruction error and the second is a regularizer. Moreover, $q(z|X)$ is the encoder output, $p(X|z)$ is the decoder output, X is the input data and z is the new sampled data from the learned distribution.

$$L(X) = -E_{z\sim Q(z|X)}[logp(X|z)] + KL(q(z|X) \parallel p(z)) \tag{3.2}$$

The reconstruction error is the basis (and often the only term) of every loss function used with neural networks. Some of the most frequent used functions here are the Mean Squared Error or the Binary Cross-Entropy for scenarios with only two possible outcomes. This term is essential for the decoder to learn how to reconstruct the data. On the other hand, the regularizer from the second term is the Kullback-Leibler divergence between the encoder and decoder distributions. This term is needed to ensure that the latent space is well structured, meaning that similar input data should be represented by similar representations of the latent space [44]. When considering the use of a VAE for missing data imputation, this loss function poses two problems. First the reconstruction error gives the same importance to the available values and to the missing ones. Although this error should consider all the data to ensure a complete learning of the network, for imputation purposes the reconstruction of the missing values should have a heavier weight on this process. Second, considering the importance of both terms from the loss function, in imputation tasks it is admissible to lose some of the structure from the latent space to ensure a better reconstruction of the missing values [14]. In other words, the Kullback-Leibler divergence may have a smaller impact on the learning process, which will lead to better reconstructions and, as consequence, better imputation results.

To address these issues we propose in this work the Variational Autoencoder with Weighted Loss (VAE-WL), consisting in a VAE with an extension of the default loss function that is presented in Equation 3.3. In this new function, the reconstruction error is split between the data containing missing values (X_{mv}) and the data that is complete (X_{av}), assigning a heavier weight to the first one through a coefficient $\gamma > 1$. Moreover, the Kullback-Leibler divergence is penalized by using another coefficient β within the range $[0, 1[$.

$$\begin{aligned} P_E(X) &= E_{z\sim Q(z|X)}[logp(X|z)] \\ L(X) &= -(P_E(X_{av}) * \boldsymbol{\gamma}\, P_E(X_{mv})) + \boldsymbol{\beta}\, KL(q(z|X) \parallel p(z)) \end{aligned} \tag{3.3}$$

An example of the impact of the proposed changes in the VAE-WL loss function is presented in Figure 3.7. The first image is an original character from the MNIST[2] dataset and the second one is the same image with 50% of its pixels missing completely at ran-

[2]Available at `http://yann.lecun.com/exdb/mnist/`.

dom. The last two images represent the imputation with a regular VAE and with the VAE-WL (respectively). The use of the new loss function shows obvious improvements in the image reconstruction. Notice that in both imputation scenarios the VAEs have the same architecture and hyperparameters and were trained with the same data.

Figure 3.7: Example of image imputation. From left to right: the 1st image is the original one, the 2nd image has 50% of its pixels missing completely at random, the 3rd image was imputed with a regular VAE, and the 4th image was imputed with the VAE-WL.

In order to properly evaluate the impact of the proposed approach in an imputation task, an experiment was conducted to compare the VAE-WL with a regular VAE. Also, the Generative Adversarial Imputation Nets (GAIN) [126] method was also considered in the study, being this another generative state-of-the-art model for missing data imputation.

Regarding the VAEs, the used architecture was obtained through experimentation and its main aspects are presented in Figure 3.8: the encoder has two convolutional layers with 32 filters, a kernel size of three, ReLu as the activation function and a stride length of two (which avoids the use of max pooling layers); the encoder also has two fully connected layers with 392 and 196 units, which also use ReLu as the activation, while the layers for the mean and variance have 32 units; the train used the optimization algorithm Adam with a learning rate of 0.001, batches of 64 images and a maximum of 200 epochs; to avoid overfitting each layer uses the L2 regularizer and is followed by a dropout layer with a 20% rate; and finally the decoder presents the inverse architecture of the encoder.

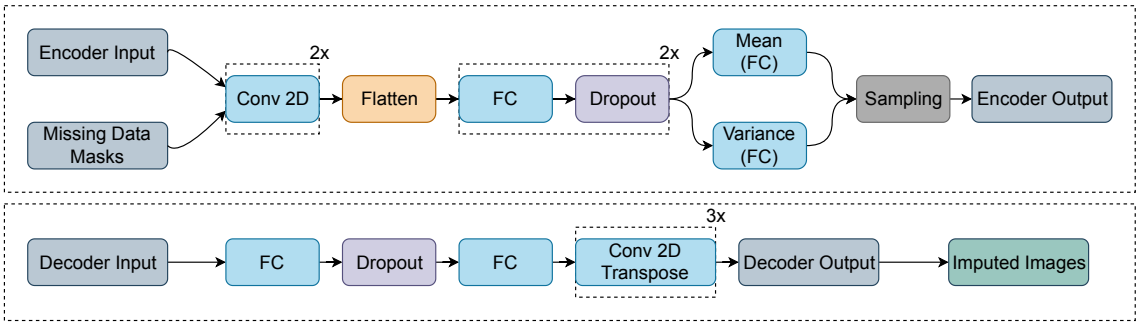

Figure 3.8: Architecture of the VAEs used in the experiments. The top rectangle represents the encoder and the bottom one the decoder.

Regarding the parameters for the VAE-WL loss function, after some experimentation with stable results they were define as $\gamma = 5$ and $\beta = 0.1$. Both VAEs use Binary Cross-Entropy for the reconstruction error of the function.

The experiment considered three datasets: MNIST, CBIS-DDSM Mass and Calcification[3]. The first is a well-known dataset for benchmarking in image related works, and it contains 70000 greyscale handwritten digits with a size of 28 by 28 pixels. The second and third datasets contain greyscale scanned mammography studies with 1696 and 1872 images, respectively, which where resized to 64 by 64 pixels. These two datasets were used because they represent a domain of medical imaging where missing values are frequent. All datasets where normalized within $[0, 1]$ and split in train, validation and test sets with 60%-20%-20% proportions. The experiment considered four missing rates (20%, 30%, 40% and 50%), with the missing values being assigned randomly to each image (following therefore the MCAR mechanism) and pre-imputed with zero. To mitigate bias and stochastic behaviors, each method was executed 30 times (the average was used as the final result) with the datasets being shuffled in each run. The imputation results were assessed through the Mean Absolute Error (MAE) metric calculated with the original images and the imputed ones.

The results obtained from the experiment are presented in Table 3.7. The VAE-WL outperforms the standard VAE and the GAIN method in all experimented scenarios. This allows for the conclusion that the imputation of missing values with the VAE-WL is in fact better than the state-of-the-art generative models. Moreover, the VAE-WL presents stable results across the different missing rates, with insignificant error increases in higher rates (the same behavior is observed in the regular VAE). On the other hand, the GAIN method presents in general worse results for smaller missing rates. An analysis of the percentage results for the VAE-WL shows average improvements of 43%, 12% and 13% for the MNIST, CBIS-DDSM Mass and Calcification datasets when comparing with a standard VAE, and 47%, 34% and 23% when comparing with the GAIN method.

3.5 Conclusions, Recommendations and Open Challenges

This study presents a technical analysis of AEs used for imputation of missing data, and compares their results with other algorithms used for the same purpose. This section concludes the analysis with a discussion of the results found, focusing on why AEs are a very powerful and promising method for missing data imputation, while also giving recommendations about the network architecture and hyperparameters to be used.

3.5.1 Architecture and Training

Defining the architecture and the respective hyperparameters of ANNs is never an easy task. Most decisions are usually based on empirical guesses or expensive grid search approaches. The vast majority of the analyzed works do not present justifications for the

[3]Available at `https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM`.

Table 3.7: Results from the experiment. The first three columns present the MAE values for the used methods. The last two columns present the percentage improvement of the VAE-WL compared with the VAE and the GAIN, respectively. The best results for each combination of dataset with missing rate are bolded.

			VAE-WL	VAE	GAIN	↑ % VAE	↑ % GAIN
MNIST		**MR 20%**	**0.036**	0.063	0.091	43%	61%
		MR 30%	**0.036**	0.064	0.066	44%	45%
		MR 40%	**0.037**	0.065	0.064	44%	43%
		MR 50%	**0.039**	0.067	0.064	42%	39%
CBIS-DDSM	**Mass**	**MR 20%**	**0.044**	0.051	0.084	13%	47%
		MR 30%	**0.045**	0.051	0.067	12%	32%
		MR 40%	**0.046**	0.052	0.055	11%	17%
		MR 50%	**0.046**	0.052	0.075	11%	39%
	Calc.	**MR 20%**	**0.046**	0.054	0.078	16%	42%
		MR 30%	**0.047**	0.054	0.066	12%	29%
		MR 40%	**0.048**	0.054	0.055	10%	13%
		MR 50%	**0.047**	0.054	0.051	12%	8%

decisions performed at this level. Instead, they tend to follow trends considered to be state-of-the-art, while just a few exceptions use grid search. As stated before, this might be due to the fact that grid search approaches are computationally expensive, and most authors do not have the resources to do it. Considering that training AEs is already a heavy process, further performing grid search could rapidly make the experimental setup more complex, time-consuming, and impractical.

From the state-of-the-art trends considered by most authors, some recommendations can be given. Regarding the number of hidden layers, although the spectrum goes from single-layer architectures to 13 multi-layered networks, most works use small-sized networks, with the most common amount of layers being 3. A possible explanation is that most datasets have a level of complexity that can be easily mapped by a small amount of layers. The benefits of using more layers in such scenarios may not compensate the increased time performance costs. Within each layer, the number of nodes varies between works, but the use of a regular step between layers is the approach most often applied, followed by the use of empirical criteria. Choosing a fixed step is recommended, since it allows a certain level of symmetry between the encoder and decoder, which appears to benefit the learning process. For each of the nodes, the activation function more often used is the ReLU, with the Sigmoid also appearing in a few works. This is an obvious state-of-the-art choice, since ReLU and its variants have been presenting good results in deep learning domains.

Focusing on the training process, it is usually performed with the Stochastic Gradient Descent or its Adam variant, with the most used loss function being the Mean Squared Error (MSE). Regularization terms are often applied to avoid overfitting, especially the

L2. Once again, such choices are the ones commonly found in state-of-the-art works that use ANNs and deep learning, being therefore recommended. This training process requires the data to be complete, otherwise the optimization of the network can not be performed. Therefore, most of the time a pre-imputation strategy is required, since the missing values cannot exist. Not many studies address this topic (and those who address it do not often elaborate on the used strategy), but as expected more complex imputation methods (e.g., kNN and MICE) tend to outperform simpler methods (e.g., mean and constant imputation). The reasoning is quite simple: if such methods perform better, they will generate better initial guesses for the network. Therefore, the use of these more robust imputation approaches should be preferred to simpler methods. Finally, for the specific case of DAEs, several works use dropout to add extra noise to the input data. This appears to reduce overfitting and improve the network robustness to missing values.

Regarding the purpose of the AEs, some works extend the basic architecture for different tasks. A purpose that stands out is using AEs for time-series imputation. Standard approaches for this type of data are applied (e.g., recurrent connections, LSTM models and a time window), but the aspects explored in this study are mostly shared with these extensions.

3.5.2 Data Contexts

Being a type of ANN, AEs can be applied to any type of data that is, or can be transformed to, numeric values. From the works under analysis, 3 predominant data contexts were detected: medical and human-related data, quality of service and traffic data. However, several works use datasets from miscellaneous contexts, most of them available in public repositories such as the UCI. Therefore, there is not enough information to say that AEs work better for those specific 3 contexts. Instead, the analysis shows that AEs are very versatile, since they obtain similar and stable results when used across a large variety of contexts. ANNs are, in general, applied to solve very different problems in the most heterogenous scenarios, and that is why the versatility of AEs comes as no surprise.

The vast majority of datasets used are small and medium-sized, mostly below 100000 observations. This may be due, once again, to the computation resources and the time complexity needed to trained AEs (and any deep learning model) in big data scenarios. Therefore, the generalization of these conclusions is limited to small and medium-sized datasets, while further work in big data contexts is required.

3.5.3 Comparative Analysis

The works under analysis report very promising results about the use of AEs for missing data imputation, with both the DAE and VAE variants surpassing their competitors in

the vast majority of works that report imputation results or its impact on classification and regression tasks. In fact, when focusing on the imputation error, AEs outperform the remaining families of algorithms in the following way:

- DAEs clearly outperform most statistical methods (the single exception is the work from [93], which obtains similar results with MICE), and marginally outperform the remaining families (matrix completion, distance and connectionist algorithms);
- VAEs outperform most methods considered (both statistical and connectionist), with the single exception being the work from Nazabal et al. [70] (the proposed method is more suitable for categorical features).

There are several aspects that may justify why AEs show such promising results and are a good alternative to other imputation methods. By being a ANNs-based model, AEs can model data patterns much more complex when compared to other simpler methods. For example, the kNN method is based on simple distances that only detect global similarity between instances, and the MICE method relies internally on simple regressions that may not be able to map complex patterns. Therefore, all known benefits of ANNs are inherited to AEs. Moreover, AEs natively support multivariate scenarios, meaning they perform the imputation of all features with missing values while accounting for the relations between them. This is something that most imputation methods are unable to do (the main exception would be the MICE method), and is the main difference between using AEs and vanilla ANNs to perform imputation. From a more practical perspective, although training AEs may be time-consuming, using a trained AE is exceptionally fast, even for the most time-restrained environments, while other methods need more time since they are executed on-the-fly (e.g., kNN) or need multiple executions (e.g., MICE).

Addressing the reported results of both variants, although they both are AEs there is a major difference between them: DAEs are discriminative models while VAEs are generative. In other words, DAEs try to reconstruct the exact input data while VAEs try to generate new instances with the same characteristics of the input data. Both variants are suitable for imputation of missing data, although VAEs may be more adequate for scenarios where the uncertainty surrounding the missing values is higher (e.g., missing data under the MNAR assumptions) because the new instances are samples of a distribution that describes the input data, and with an increased number of samples the uncertainty will be reduced.

Focusing on the impact of imputation in classification and regression tasks, this is an understudied topic with only 10 works touching upon this aspect. Nevertheless, the current results are very encouraging for DAEs, since the classifiers/regressors present in general better accuracy results. The same is not true for VAEs, which show in general very slight improvements in such tasks. Since VAEs generate new instances, these may contain differences that change the relation with the label, although such behavior would be highly

influenced by the missing data mechanism. One conclusion that can be drawn from these results is that the best method for imputation may not necessarily be the one that leads to better classification/regression results. Therefore, it is important to evaluate both the imputation error and the impact on predictive tasks when dealing with missing data.

Finally, a trend was also found in the evaluation metrics: RMSE is the one more often used for imputation, and accuracy is usually used for classification tasks. While the accuracy comes with no surprise since it is widely used for classification evaluation, the choice of RMSE for imputation may be questionable. RMSE is the square root of the average of squared errors, meaning that different values have a disproportional impact on final aggregated result. In fact, higher errors will have more impact on the metric's value. If this behavior is not desired, another metric may be more suitable (e.g., MAE).

3.5.4 Open Challenges

The works under analysis in this study provide important information about using AEs for imputation purposes, and show that they present good results when compared to other state-of-the-art methods. Nevertheless, several open challenges have also been identified that require further study.

An important but almost unaddressed topic by these works is the missing data mechanism. Only 9 of the 26 works describe the missing mechanism associated with the datasets, which does not allow conclusions to be drawn about how well AEs perform individually for missing values under MCAR, MAR and MNAR assumptions. Future studies should consider the missing mechanisms more often, and individual results should be presented for them. Moreover, the 9 works that report the mechanism only address MCAR and MNAR, leaving a gap for the MAR mechanism to be addressed. Furthermore, in order to have a controlled experiment most works rely on missing values artificially generated. But, the approach followed to perform this generation is rarely presented. Future works should describe this aspect, in order to make the experiments reproducible and to allow a more comprehensive discussion of the results.

The use of VAEs is also understudied, both for imputation quality and classification/regression impact. This variant was only used recently to address missing data issues, which may explain the smaller amount of works available. Nevertheless, future studies should address more actively this AE variant, since it presents suitable characteristics for missing data imputation [77] (e.g., it can generate new values following the same distribution as the data used for training). Another understudied topic is the impact of AEs imputation in classification and regression tasks. This is probably the most urgent aspect to be actively address by the community. Performing the imputation of missing values is often an intermediate step for another goal, which usually is a classification or regression problem. Therefore, it is important to assess the imputation quality, but evaluating the

impact of such operation in the other goals is of equal importance. If applicable, further studies should focus their evaluation in both components. Furthermore, the current works compare AEs with very different imputation methods in the experiments, but some state-of-the-art algorithms are neglected (e.g., SVM), and therefore they should be included in the baseline of studies.

The pre-processing operations required for the data to be used with AEs are a limitation that could also be mitigated in the future. The pre-imputation step and the approaches applied to categorical features (e.g., one-hot encoding) can add bias to the results. New strategies that could reduce the impact of such operations are desired.

Only a small set of works use AEs with data containing relevant structural information (i.e., data where the position and order of the features have a meaning, such as images). Although the study mostly provides an analysis over tabular data, the results presented in Section 3.4 show this is a promising direction, with AEs outperforming all the remaining methods. Therefore, such application of this type of ANN should also be further explored.

Finally, only 2 of the works under analysis report the execution time of the algorithms, and both use small datasets. Considering that deep learning methods are usually very computationally expensive, the existence of analysis focused on time complexity would be important to understand the environments in which AEs can be applied or must be discarded for being too heavy. Therefore, the resources required to train the network should also be addressed in future studies, particularly in big data contexts.

This page is intentionally left blank.

Chapter 4

Variational Autoencoder Filter for Bayesian Ridge Imputation

VAEs have been used to address missing data by performing the entire imputation. However, the results are limited and lack significance when comparing to other state-of-the-art methods, with stronger evidence in structured data [12, 65].

In this work a new approach called Variational Autoencoder Filter for Bayesian Ridge Imputation (VAE-BRIDGE) is introduced, and it comprises two main parts:

- A VAE is used in the initial steps of the imputation pipeline to filter the instances that will be considered for the generation of new values (therefore performing a kind of instance selection);
- The final imputation is performed by a Bayesian ridge regression fitted with the filtered instances.

The approach is compared with other state-of-the-art methods in an experiment that uses 10 public datasets from clinical trials that were injected with missing values under the MNAR mechanism. This data context and mechanism were used because healthcare studies suffer frequently from missing values, and these ones are usually under the MNAR assumptions. Moreover, four different missing rates were considered in the study (10%, 20%, 30% and 40%). The results from the experiment show that the VAE-BRIDGE approach outperformed all the remaining state-of-the-art methods with an overall improvement of 26% and 67% comparing to the second best and worst methods of each dataset, respectively. These results were proved to have a statistically significant of 5% through the Three-Way ANOVA and post-hoc Tukey's HSD tests. To the best of the authors' knowledge, the proposed approach is novel in the missing data field since VAEs were never used for this purpose.

The remainder of the chapter is organized in the following way: Section 4.1 presents the

background concepts of the Bayesian ridge regression; Section 4.2 describes the VAE-BRIDGE approach here proposed; Section 4.3 describes the experiment and the obtained results; and Section 4.4 shows the conclusions and future directions of this work.

4.1 Bayesian Ridge Regression

The frequentist approach to perform linear regression is based on assigning a weight/coefficient to each independent variable, reflecting therefore their effect on the dependent variable. An error term is also considered to account for noise and other external factors. The model is usually fitted through the Ordinary Least Squares (OLS) approach, which minimizes the residual sum of squares (see Equation 4.1, were y is the dependent variable, w the coefficients and x_i one of the M instances) [69]. In other words, the goal is to minimize the model error by adjusting the w coefficients.

$$min \sum_{i=1}^{M} (y_i - w^T x_i) \tag{4.1}$$

A problem found in this type of regression is overfitting, particularly when data displays multicollinearity patterns. To mitigate this issue regularization is often used. A common strategy is to penalize the size of the w coefficients with a L2 regularizer, creating the so-called ridge regression [45, 69].

A limitation of the OLS method is the w coefficient values having single point estimates, meaning these values are the ones most likely to be correct given the training data. However, the uncertainty surrounding the model results are not accounted at all.

The Bayesian approach addresses this issue by modeling the regression with probability distributions instead of single value estimates. Assuming the use of a Gaussian distribution, the formulation of the Bayesian regression is presented in Equation 4.2 (using the same definitions from Equation 4.1, with X being a matrix with all M instances) [45, 69].

$$y \sim \mathcal{N}(w^T X, \sigma^2) \tag{4.2}$$

The outcome will therefore be the posterior distribution of the w coefficients, instead of their exact values. This change allows the use of priors, which can be helpful if relevant information about the model is already known. Moreover, the model uncertainty is accounted for, since the posterior distribution gives a range of possible w coefficients based on the data and the prior [69]. This last aspect is particularly relevant when the amount of instances used to fit the model is reduced. In fact, when the number of instances increases the w coefficients converge to the ones obtain from the OLS method, since the level of uncertainty is decreasing. Moreover, when the prior also follows a Gaussian distribution,

the L2 regularization is implicitly applied, creating the Bayesian ridge regression concept [69, 110].

One of the most common ways to fit a Bayesian regression is drawing samples from the posterior distribution to improve and approximate it, using, for example, Monte Carlo methods. Another common approach is to use the Maximum A Posteriori (MAP) method [69, 110].

4.2 Proposed Approach

The VAE-BRIDGE approach starts by training a VAE with all data instances that are complete, but excluding the feature containing missing values (meaning that no pre-imputation is required). The model will learn the multidimensional parameters of the Gaussian distribution that represents the data, which will then be used for filtering purposes. Afterwards, each instance having missing values is encoded with the previously trained VAE, and its multidimensional Gaussian parameters are compared to the ones from each complete instance. The goal is to obtain the k percent instances that are described by the most similar Gaussian distributions, following the formula from Equation 4.3. This distance is an adaptation of the euclidean metric to include both Gaussian parameters (mean and variance).

$$d_{p,q} = \sqrt{\sum_{i=1}^{n}(\mu_{p_i} - \mu_{q_i})} + \sqrt{\sum_{i=1}^{n}({\sigma^2}_{p_i} - {\sigma^2}_{q_i})} \tag{4.3}$$

The selected k percent instances are finally used to fit a Bayesian ridge regression. Any imputation model could be used in this step, but the Bayesian ridge is known to provide better long term predictions through regularization strategies that deal with overfitting (see Section 4.1), being for that reason used by state-of-the-art methods such as MICE [15]. An additional aspect of this regression model is the easiness to interpret its results, something that is often important in sensitive contexts (e.g., healthcare). For datasets with more than two features this regression will be multivariate, with all the features without missing values being the independent variables.

For a proper generalization of this method the following aspects must also be considered:

- If the missing data scenario is multivariate (i.e., two or more features have missing values), the described process must be repeated individually for each of these features;
- To avoid issues with the domain of the features and to speed up the VAE training convergence, all features should be normalized within $[0, 1]$. Consequently, the VAE

output layer should use sigmoid as the activation function;

- Categorical features must be transformed to binary ones through a one-hot encoding process, otherwise the VAE training procedure and the distance formula from Equation 4.3 will not be valid. The imputation of these features will be a real value within $[0, 1]$ (as the previous point states), which can be converted to a binary value assuming a fixed threshold (e.g., 0.5).

The complete VAE-BRIDGE approach is summarized in Algorithm 1.

Algorithm 1 Pseudocode of the VAE-BRIDGE algorithm.

Input: *complete_rows*, *incomplete_rows*, k
Output: *imputed_rows*

```
Normalize all data within [0, 1]
Apply one-hot encoding to the categorical features
for each feature having missing values (md_i) do
    Train a VAE with complete_rows \ md_i
    enc_data ← Encode complete_rows \ md_i with VAE
    for each instance z in incomplete_rows do
        enc_z ← Encode z \ md_i with VAE
        sim_z ← Find k similar rows to enc_z from enc_data using the distance formula from Eq. 1
        br ← Fit a Bayesian ridge regression with sim_z
        imputed_rows ← Predict missing data in z with br
    end for
end for
return imputed_rows
```

The key aspect of this method is the filtering step based on the VAE encoding capabilities. By choosing only the k percent instances that have similar distribution parameters to the one that is being imputed, the method ensures that noisy data not relevant for the imputation task is ignored by the Bayesian ridge regression during the final prediction step. Therefore, for this reason, different k values will have a major impact on the approach results. To properly understand this impact a sensitivity analysis study was conducted. The experiment used the well-known Breast Cancer Wisconsin dataset, with missing values under MNAR with different rates (10%, 20% and 30%). Only this dataset was used given the number of variables to consider in the study, and the choice was based on its popularity among the healthcare public datasets. The focus on the MNAR mechanism is justified by the fact that most missing values in healthcare are usually under MNAR assumptions, as previously stated. The results were evaluated through the Mean Absolute Error (MAE). Figure 4.1 presents the conclusions of the study, where each bar shows the average number of times each k value presented the best MAE results for the three missing rates. The study considered 10 different percentages for k (10% to 100%, with 10% steps).

From the obtained results, the k values $\{20, 30, 40\}$ clearly outperformed the remaining ones, with $k = 20$ presenting the best overall score. Such results not only show the best k

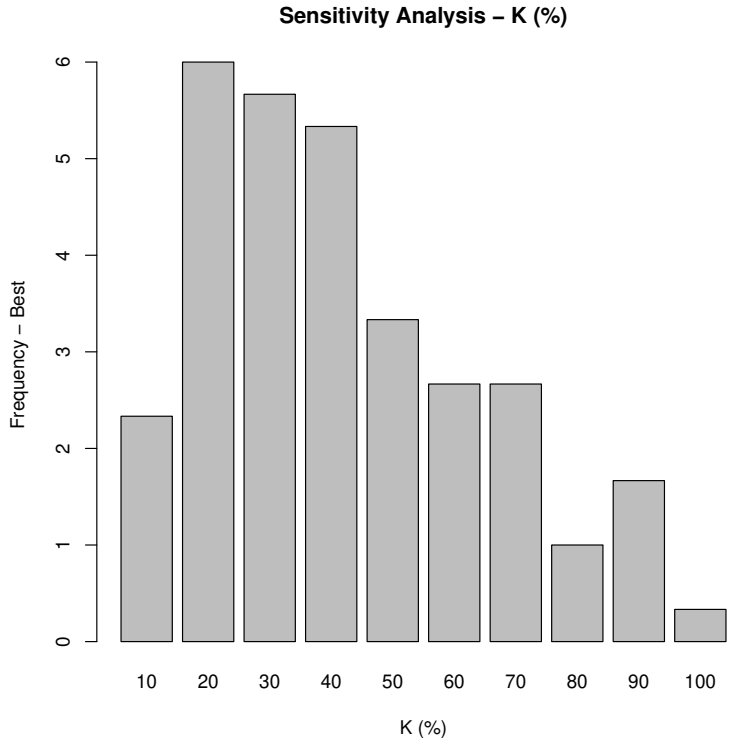

Figure 4.1: Sensitivity analysis of the k parameter. Each bar shows the average number of times each k value presented the best MAE results for the three missing rates.

values to be used, but also prove the impact of using only relevant data for the imputation. In general, when k increases over the 40% threshold, the results tend to be worse. This can be explained by the fact that the Bayesian ridge regression is using more data that is not relevant to the imputation, which creates additional noise in the fitting process. Using $k = 10$ also presented bad results since the filter probably discarded more information than it should, leading to the lost of data that was in fact relevant. Therefore, the used k value must neither be too small or too big, with percentages between 20 and 40 showing a good balance for this criteria. Nevertheless, to have accurate k values this study should be conducted for the different datasets. However, given the computation complexity of this task, only the best values obtained for the Wisconsin dataset ($k = \{20, 30, 40\}$) were used for the remaining ones in the experiments.

4.3 Experimental Results

To evaluate the quality of the imputation performed by the VAE-BRIDGE method an experiment was conducted, aiming to compare it to the following state-of-the-art methods:

- Standard VAE, which performed the entire imputation [65];

- Denoising Autoencoder (DAE), which is a basic discriminative Autoencoder that learns a compressed representation of the input data with noise (in this context the missing values are the noise) [118];
- Multiple Imputation by Chained Equations (MICE), which uses a multiple imputation approach to fit regression models using the features with missing values as the dependent variables [15];
- K-Nearest Neighbors (KNN) Imputation with $K = 5$, which finds the five nearest neighbors of an instance through a distance metric (usually the Euclidean one) and uses their average values to impute the missing data [95].

Regarding the Autoencoder-based methods (VAE-BRIDGE, VAE and DAE), they all used a similar architecture:

- A single hidden layer with 100 units and the ReLU activation function for the DAE, while the VAEs had two "parallel" layers (mean and variance) with 100 units and linear activation;
- The optimization algorithm used was Adam with a learning rate of 0.001, batches of 64 instances and 200 epochs;
- To avoid overfitting each layer applies the L2 regularizer with a factor of 0.01;
- The weights of the layers are initialized using the glorot normal approach, which follows a truncated normal distribution centered on zero;
- The output layer always uses the sigmoid activation function, as explained in the previous section.

Moreover, the VAE-BRIDGE approach used $k = \{20, 30, 40\}\%$, since these were the best overall values obtained from the sensitivity analysis study presented in the previous section. However, the best results were once again obtained for $k = 20$, being therefore the ones presented here.

The experiment considered 10 public datasets from the healthcare context, covering clinical studies of different pathologies. The choice of this context lies in the fact that it often suffers from missing data, which compromises severely the studies' results [76]. All datasets were obtained from the UCI[1] repository, and they have different sizes and both continuous and categorical features, as Table 4.1 shows.

All datasets were complete and were latter injected with missing values under the MNAR mechanism, using the method from [112] (the lowest values are set to be missing). This missing mechanism was used since it is the one more often found in healthcare contexts, and

[1] `https://archive.ics.uci.edu/ml/datasets.php`

Table 4.1: Characteristics of the datasets used in the experiments.

Dataset	**# Instances**	**# Features**	
		Continuous	**Categorical**
wisconsin	569	31	0
ctg	2126	21	2
pima	768	9	0
liver	583	10	1
hcv-egy	1385	19	10
parkinsons	195	23	0
bc-coimbra	116	10	0
thoracic-surgery	470	14	3
spine	310	13	0
mammographic-masses	830	2	4

it is also the one that poses more challenges for the imputation task [76]. Each feature of each dataset was iteratively injected with missing data and imputed, with the imputation quality being assessed through the Mean Absolute Error (MAE) metric, calculated between the ground truth and the imputed data. The final MAE of a dataset is the average MAE of all its features.

The data was normalized within $[0, 1]$ and split in train, validation and test sets (with 60%-20%-20% proportions) for all methods except the KNN, since it does not require training. Four missing rates were considered (10%, 20%, 30% and 40%), with the missing values being pre-imputed with the mean for the DAE and standard VAE methods. Notice that the missing values were injected independently for the train, validation and test sets, in order to ensure the same missing rate and MNAR assumptions for each one. To mitigate bias and stochastic behaviors, the experiment was executed five independent times, with the datasets being shuffled in each run. Moreover, the average results from the runs were considered for comparison. A graphical representation of the experimental setup here described is presented in Figure 4.2.

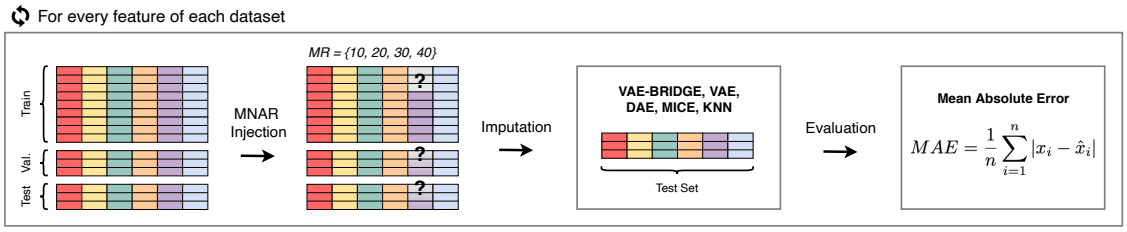

Figure 4.2: Graphical representation of the experimental setup used in this work.

The results from the experiment are presented in Table 4.2. The VAE-BRIDGE approach outperformed all state-of-the-art methods for every dataset and missing rate. The overall improvement from the second best and worst methods to VAE-BRIDGE was 26% and 67%, respectively. The second best method varied between VAE and MICE depending on the dataset, while the DAE presented the overall worst results.

Table 4.2: Results from the experiment. The first two columns identify the dataset and the missing rate percentage. The next five columns present the MAE values (mean and standard deviation) for the used methods. The last column shows the percentage improvement from the best method to the second best. The best results for each dataset and missing rate are bolded.

Dataset	MR (%)	VAE-BRIDGE	VAE	DAE	MICE	KNN	↑ %
wisconsin	10	**0.019 ± 0.019**	0.078 ± 0.037	0.213 ± 0.049	0.03 ± 0.034	0.074 ± 0.037	37%
	20	**0.017 ± 0.018**	0.062 ± 0.028	0.189 ± 0.045	0.028 ± 0.034	0.079 ± 0.037	39%
	30	**0.017 ± 0.017**	0.055 ± 0.023	0.173 ± 0.044	0.029 ± 0.036	0.085 ± 0.038	41%
	40	**0.017 ± 0.017**	0.052 ± 0.021	0.158 ± 0.045	0.032 ± 0.042	0.094 ± 0.037	47%
ctg	10	**0.028 ± 0.048**	0.051 ± 0.063	0.198 ± 0.149	0.043 ± 0.072	0.06 ± 0.092	35%
	20	**0.025 ± 0.043**	0.042 ± 0.05	0.179 ± 0.133	0.043 ± 0.076	0.065 ± 0.086	40%
	30	**0.024 ± 0.041**	0.037 ± 0.045	0.161 ± 0.106	0.066 ± 0.139	0.091 ± 0.142	35%
	40	**0.023 ± 0.038**	0.035 ± 0.042	0.148 ± 0.09	0.065 ± 0.118	0.094 ± 0.125	34%
pima	10	**0.161 ± 0.077**	0.176 ± 0.075	0.237 ± 0.078	0.195 ± 0.093	0.177 ± 0.1	9%
	20	**0.136 ± 0.052**	0.142 ± 0.043	0.208 ± 0.065	0.189 ± 0.067	0.174 ± 0.069	4%
	30	**0.124 ± 0.051**	0.127 ± 0.032	0.191 ± 0.069	0.203 ± 0.073	0.188 ± 0.071	2%
	40	**0.112 ± 0.05**	0.117 ± 0.024	0.173 ± 0.074	0.216 ± 0.082	0.199 ± 0.069	4%
liver	10	**0.096 ± 0.19**	0.158 ± 0.196	0.28 ± 0.223	0.117 ± 0.219	0.131 ± 0.201	18%
	20	**0.09 ± 0.187**	0.139 ± 0.185	0.263 ± 0.225	0.127 ± 0.25	0.145 ± 0.235	29%
	30	**0.083 ± 0.178**	0.123 ± 0.171	0.238 ± 0.203	0.198 ± 0.321	0.218 ± 0.31	33%
	40	**0.073 ± 0.153**	0.108 ± 0.146	0.212 ± 0.171	0.167 ± 0.241	0.189 ± 0.234	32%
hcv-egy	10	**0.141 ± 0.202**	0.155 ± 0.181	0.347 ± 0.163	0.156 ± 0.224	0.303 ± 0.142	9%
	20	**0.127 ± 0.181**	0.132 ± 0.159	0.333 ± 0.156	0.162 ± 0.229	0.331 ± 0.137	4%
	30	**0.113 ± 0.16**	0.117 ± 0.142	0.319 ± 0.152	0.167 ± 0.234	0.372 ± 0.142	3%
	40	**0.098 ± 0.138**	0.106 ± 0.126	0.303 ± 0.151	0.169 ± 0.237	0.441 ± 0.181	8%
parkinsons	10	**0.066 ± 0.089**	0.134 ± 0.131	0.284 ± 0.104	0.096 ± 0.145	0.108 ± 0.092	31%
	20	**0.059 ± 0.078**	0.117 ± 0.112	0.254 ± 0.094	0.103 ± 0.162	0.132 ± 0.122	43%
	30	**0.053 ± 0.068**	0.103 ± 0.089	0.231 ± 0.079	0.107 ± 0.156	0.15 ± 0.145	49%
	40	**0.049 ± 0.06**	0.094 ± 0.074	0.21 ± 0.07	0.105 ± 0.133	0.159 ± 0.12	48%
bc-coimbra	10	**0.142 ± 0.106**	0.188 ± 0.109	0.269 ± 0.069	0.185 ± 0.132	0.172 ± 0.116	17%
	20	**0.131 ± 0.088**	0.16 ± 0.085	0.243 ± 0.054	0.184 ± 0.125	0.183 ± 0.108	18%
	30	**0.122 ± 0.079**	0.15 ± 0.079	0.229 ± 0.047	0.197 ± 0.129	0.197 ± 0.113	19%
	40	**0.11 ± 0.071**	0.141 ± 0.07	0.208 ± 0.041	0.211 ± 0.133	0.212 ± 0.112	22%
thoracic-surgery	10	**0.084 ± 0.166**	0.114 ± 0.162	0.259 ± 0.211	0.096 ± 0.191	0.122 ± 0.194	13%
	20	**0.077 ± 0.148**	0.099 ± 0.142	0.254 ± 0.208	0.1 ± 0.192	0.139 ± 0.217	22%
	30	**0.07 ± 0.127**	0.087 ± 0.117	0.244 ± 0.191	0.135 ± 0.233	0.159 ± 0.237	20%
	40	**0.062 ± 0.108**	0.079 ± 0.097	0.232 ± 0.171	0.158 ± 0.249	0.178 ± 0.248	22%
spine	10	**0.274 ± 0.219**	0.326 ± 0.154	0.363 ± 0.137	0.297 ± 0.231	0.359 ± 0.191	8%
	20	**0.25 ± 0.205**	0.289 ± 0.157	0.326 ± 0.14	0.31 ± 0.251	0.362 ± 0.213	13%
	30	**0.219 ± 0.186**	0.251 ± 0.136	0.285 ± 0.134	0.323 ± 0.277	0.368 ± 0.227	13%
	40	**0.191 ± 0.162**	0.218 ± 0.117	0.249 ± 0.12	0.313 ± 0.252	0.36 ± 0.201	12%
mammographic-masses	10	**0.025 ± 0.072**	0.05 ± 0.064	0.242 ± 0.175	0.056 ± 0.161	0.072 ± 0.154	50%
	20	**0.022 ± 0.065**	0.044 ± 0.058	0.229 ± 0.153	0.045 ± 0.116	0.062 ± 0.111	50%
	30	**0.02 ± 0.062**	0.041 ± 0.059	0.223 ± 0.149	0.042 ± 0.108	0.059 ± 0.1	51%
	40	**0.02 ± 0.059**	0.04 ± 0.059	0.219 ± 0.149	0.045 ± 0.119	0.071 ± 0.113	50%

To assess the statistical significance of the results, the Three-Way ANOVA test was applied with a significance level of 5%. The factors considered were the dataset, the missing rate and the algorithm, while the dependent variable was the MAE. This statistical test can only be applied if the data follows a normal distribution, which was confirmed visually through a Q-Q plot. Moreover, the same analysis was conducted on the data subgroups.

The p-values from the test show the results are statistically significant for the datasets and algorithms ($p = 2e-16$). However, no evidence of sensitivity to the missing rates was found ($p = 0.365$).

To conclude if the VAE-BRIDGE approach outperformed the remaining algorithms with a statistical significance of 5%, the post-hoc Tukey's HSD test was applied for this particular factor. The VAE-BRIDGE significantly outperformed VAE ($p = 0.006732$), DAE ($p < 0.000001$), MICE ($p = 0.000001$) and KNN ($p < 0.000001$).

4.4 Conclusions

In this work a new approach for missing data imputation called Variational Autoencoder Filter for Bayesian Ridge Imputation (VAE-BRIDGE) is introduced. It uses a VAE to filter the instances that are relevant for the imputation, while a Bayesian ridge regression is fitted with them and predicts the new values. The method relies on the fact that instances that are not relevant for the imputation may compromise its results by adding unnecessary noise. The approach was compared with four state-of-the-art methods (standard VAE, DAE, MICE and KNN) in an experiment with 10 datasets from clinical trials that were injected with missing values under MNAR (including 10%, 20%, 30% and 40% missing rates). The VAE-BRIDGE approach outperformed all the remaining methods, achieving an overall improvement between 26% and 67%. The results were validated with a statistical significance of 5%.

In the future new experiments will be conducted to test the VAE-BRIDGE approach with datasets from other contexts (some containing pre-existent missing values) and with the remaining missing data mechanisms (MCAR and MAR), considering in this last scenario the impact of having different mechanisms in the train, validation and test sets. Moreover, a study on the impact of the imputation in classification tasks will also be conducted.

This page is intentionally left blank.

Chapter 5

Partial Multiple Imputation with Variational Autoencoders

A recent trend in imputation that has been showing good results particularly for MNAR is the use of generative models, namely Variational Autoencoders (VAE) [13, 65, 77, 78]. By generating samples that are based on the training data but diverge towards new concepts, the VAE tends to meet the MNAR assumptions by reducing the bias. However, this method still poses a limitation when dealing with MNAR: it only provides a single imputation result, which may have random bias given the stochastic behavior of the reparameterization trick. To tackle this limitation, we propose in this work an extension to the vanilla (i.e., standard) VAE that uses a partial Multiple Imputation (MI) strategy, which we call Partial Multiple Imputation with Variational Autoencoders (PMIVAE). Several studies show the benefits of applying MI strategies in MNAR scenarios [9, 84], but such strategies were never combined with VAEs. Moreover, considering that the VAE's latent space is a probability distribution, MI can be applied through sampling operations, which would not be possible with other generative models (e.g., Generative Adversarial Networks).

Our experimental setup considered 34 public datasets from the medical context, 8 state-of-the-art imputation approaches that were compared with the PMIVAE, and 5 missing rates from 10% to 80% achieved by artificially generating the missing values. With this high number of datasets, we aim to cover data scenarios with different characteristics and create an experimental setup that is representative of the missing data issue in clinical research. The missing values were artificially generated under the MNAR assumptions [97, 112], and the imputation results were evaluated through the Mean Absolute Error (MAE) between the imputed data and the ground truth. These results were proved to have a statistical significance of 5% through the Three-Way ANOVA on ranks and post-hoc Tukey's HSD tests. Moreover, an analysis of the imputation results on different probability distributions is also presented, followed by an experiment in a multivariate setting with

other deep learning-based models and a case study for a heart failure dataset, where the impact of the imputation in a classification task is addressed.

The remainder of the chapter is organized in the following way: Section 5.1 describes the PMIVAE method here proposed; Section 5.2 describes the experimental setup and results; Section 5.3 extends the experimental setup to encompass a multivariate setting; Section 5.4 presents a case study of the imputation impact on a classification task with heart failure patients; and Section 5.5 presents the conclusions and future directions of this work.

5.1 Proposed Approach

The VAE model has a loss function with two terms: the reconstruction error and a regularizer (see Equation 5.1, where $q(z|X)$ is the encoder output, $p(X|z)$ is the decoder output, X is the input data and z represents the new samples). The regularizer is the Kullback-Leibler divergence, which forces the disentanglement of the latent space, leading to a well-structured space where similar input data generates similar latent representations. Moreover, it also helps mitigate overfitting [44].

$$L(X) = -E_{z\sim Q(z|X)}[logp(X|z)] + KL(q(z|X) \parallel p(z)) \tag{5.1}$$

In order for the loss function to be differentiable the sampling process must be stochastic. Therefore, the reparameterization trick is applied. This process adds a random variable, sampled from a normal distribution with $\mu = 0$ and $\sigma = 1$, to the sampling process. This ensures the distribution parameters remain as the learnable ones of the network [44].

Although the VAE has presented good results in several missing data imputation scenarios, it still poses a major limitation: the imputed values are still the result of a single imputation procedure. Given the stochastic nature of the model, caused by the reparameterization trick, the new values being generated can have a random bias. To tackle this issue we propose a partial MI approach, based on a Monte-Carlo sampling, that is applied to the VAE generative phase. We call it Partial Multiple Imputation with Variational Autoencoders (PMIVAE).

The main aspects of the proposed approach are described in Algorithm 2 through pseudocode. To generate a new instance, the VAE takes a sample from the Gaussian parameters previously learned and decodes it. In our approach, the sampling process is repeated N times, leading to N latent space representations of new instances. These samples are aggregated into a single one by an averaging operation, and the resulting sample is then decoded. We denote this procedure as a partial MI strategy since for it to be a standard MI the N results from the sampling process need to be independently analyzed before the

aggregation, which is not applicable in this context. The goal is to reduce bias and uncertainty in the generative process by accounting for the stochastic behavior of the model with this partial MI strategy. Furthermore, the integration of the MI with the VAE is a key aspect to achieve better results. By applying this procedure directly to the latent space of the VAE model, the multiple samples are generated only from the relevant information learned by it. If the MI was performed in a pre-processing stage, the additional noise in the input data would propagate to the MI results.

Algorithm 2 Pseudocode for the PMIVAE method.

Input: *train_rows*, *rows_to_impute*, N
Output: *imputed_rows*

pre_train_rows ← Pre-impute *train_rows* with the features' mean
pre_rows_to_impute ← Pre-impute *rows_to_impute* with the features' mean
vae_model ← Train a VAE with *pre_train_rows*
enc_rows_to_impute ← Encode *pre_rows_to_impute* with *vae_model*
μ_{vae}, σ_{vae} ← Extract Gaussian parameters from *enc_rows_to_impute*
for each i in $\{1, ..., N\}$ **do**
    $x_i = \mu_{vae} + y_i * \sigma_{vae},\ y_i \sim \mathcal{N}(0, 1)$
    Append x_i to X
end for
X_{mean} ← Average of X
imputed_rows ← Decode X_{mean} with *vae_model*
return *imputed_rows*

The PMIVAE method requires a number of iterations (N), as most MI procedures do. To understand the impact of this parameter in the results, we conducted a sensitivity analysis study that considered 5 values for N (10, 20, 50, 100 and 200), 5 datasets and a missing rate of 40% (which represents a moderate amount of missing values). The remaining settings of the experiment are the same used for the main experimental results of this work (see Section 5.2). We only considered a subset of the datasets and missing rates used in the main experiments because the time complexity, imposed by testing several N values, was very high. Furthermore, we settled the maximum N value as 200 since it is a reasonable limit when comparing to the iterations used by other works [122]. The study results, evaluated through the Mean Absolute Error (MAE), are presented in Table 5.1. As expected, we see that the best results tend to be achieved with higher N values, although some differences are small. This indicates that the results should be similar with any of the N values. Nevertheless, $N = 200$ iterations still lead to the best results, and therefore we decided to use it for the main experiments.

5.2 Experimental Results

To evaluate the imputation results of the PMIVAE method, an experiment was conducted where this method was compared with its established state-of-the-art competitors, pre-

Table 5.1: Sensitivity analysis for the number of iterations. The best MAE results are bolded.

Dataset	Number of Iterations (N)				
	10	**20**	**50**	**100**	**200**
alzheimer-v1	0.1397	0.1402	0.1394	0.1395	**0.1390**
bupa	0.0868	**0.0854**	0.0855	0.0856	**0.0854**
ecoli	0.1127	**0.1125**	0.1126	0.1128	0.1126
pima	0.1072	0.1066	0.1069	0.1065	**0.1064**
wisconsin	0.0598	0.0585	0.0580	0.0578	**0.0577**

sented in Table 5.2 together with their respective configurations. All Autoencoder-based methods (PMIVAE, VAE and DAE) used a similar neural-network architecture, which is described in Table 5.3. This architecture and the configurations for the remaining methods were defined through a grid search procedure. Furthermore, the Autoencoder-based methods were implemented with the Keras library, while the remaining methods were directly used from the Scikit-learn library. The implementation of PMIVAE method is available on GitHub[1].

Table 5.2: Baseline of imputation methods used for comparison.

Method	Description
Mean Imputation	The imputed value is the mean of the available values for each feature.
Multiple Imputation by Chained Equations (MICE)	Applies a MI strategy in the fitting process of several Bayesian ridge regressions using a round-robin approach with 10 iterations, and assuming as the dependent variables the features containing missing values [15].
k-Nearest Neighbors (kNN) Imputation with $k = 5$	Finds the k nearest neighbors of the instance being imputed using the Euclidean distance between the features without missing data, and imputes using the average values from the k instances [95].
SoftImpute	Performs matrix completion through an iterative approach based on nuclear-norm regularization, where the missing values are replaced by estimations from a soft-thresholded SVD [64].
Denoising Autoencoder (DAE)	Vanilla Autoencoder that learns from a noisy variant of the input data. This variant is created through a corruption process, which in this context is the amputation of values [118].
Variational Autoencoder (VAE)	Vanilla VAE (i.e., the basis for the PMIVAE), but used to perform single-sample estimation [65].

The experiment was executed with 34 public datasets from the medical context, which cover different pathologies and domains of clinical research based on routinely collected

[1] `https://github.com/ricardodcpereira/PMIVAE`

Table 5.3: Network architecture of the Autoencoder-based methods.

Aspect	Configuration
DAE Structure	Single hidden layer with 100 units and the ReLU activation function.
VAE Structure	Three layers with 100 units: two "parallel" layers (mean and variance), followed by the layer that samples points from the latent Gaussian distribution. This is the basic configuration for any VAE.
Optimization Algorithm	Adam, with the Mean Squared Error as the loss function and a learning rate of 0.001.
Training Procedure	Batches of 64 instances, a maximum of 200 epochs, an early stopping rule that stops training if the validation loss as no improvements over 100 epochs, and a reduction of the learning rate by 80% using again the previous criteria.
Overfitting Mitigation	Each layer uses L2 regularization with a 0.01 factor.
Network Initialization	Network weights are randomly initialized by sampling from a truncated normal distribution centered on zero.
Output	The output layers use the sigmoid activation function, since the data is normalized between $[0, 1]$.

healthcare data. All datasets are public (most of them can be downloaded from the UCI[2] repository), and they are very heterogeneous in regard to the number of instances and types of features, as Table 5.4 shows: they range from 68 to 2126 instances, from 1 to 19 categorial features, and from 1 to 22 continuous features.

Moreover, the 34 datasets include only complete data: to perform the experiment in a controlled manner, these datasets were injected with missing values under the MNAR mechanism, following the strategy proposed in [112], where the lowest values of the continuous features are removed upon a certain missing rate (see Figure 5.1 for a high-level representation of this approach). Each feature was iteratively injected with missing values and imputed, while the imputation results were assessed through the MAE metric applied between the ground truth (i.e., complete dataset) and the imputed data. This strategy was chosen to maximize the coverage of the experiments and to avoid bias by randomly selecting features. Also, this metric was chosen so that all errors had the same weight in the final results. The overall MAE for a dataset is the mean MAE of all its features.

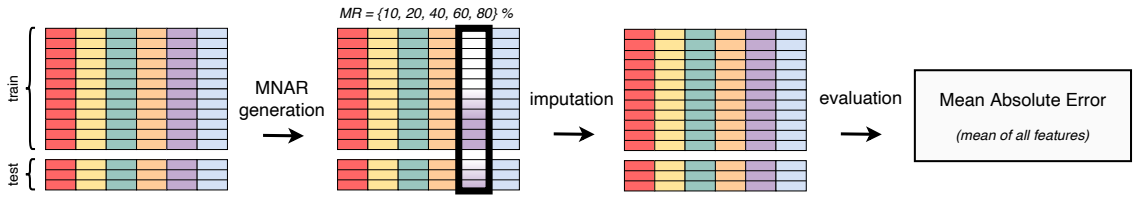

Figure 5.1: High-level representation of the experimental setup.

[2] `https://archive.ics.uci.edu/ml/datasets.php`

Table 5.4: Characteristics of the public medical datasets.

Dataset	# Instances	# Features	
		Categorical	Continuous
acute-inflammations-nephritis	120	6	1
acute-inflammations-urinary	120	6	1
alzheimer-v1	317	3	7
autism-adolescent	98	19	1
autism-adult	701	16	1
autism-child	288	16	1
bc-coimbra	116	1	9
biomed	194	1	5
breast-tissue-2c	106	1	9
bupa	345	2	5
cleveland_0_vs_4	173	1	13
cmc	1473	8	2
cryotherapy	90	3	4
ctg-2c	2126	1	21
diabetic-retinopathy	1151	4	16
ecoli	336	1	7
fertility-diagnosis	100	8	2
haberman	306	1	3
heart-statlog	270	7	7
immunotherapy	90	3	5
kala-azar	68	2	5
lymphography-v1	142	16	3
new-thyroid-N-vs-HH	215	1	5
newthyroid-v1	185	1	5
parkinson	195	1	22
pima	768	1	8
postoperative-SvsA	86	8	1
relax	182	1	12
saheart	462	2	8
thoracic	470	14	3
thyroid_3_vs_2	703	1	21
transfusion	748	1	4
vertebral-2c	310	1	6
wisconsin	683	1	9

All data was normalized within $[0, 1]$ and split in train and test sets following a 70%-30% ratio. For the Autoencoder-based methods, 20% of the training set is used for validation purposes. The categorical features were transformed with the one-hot encoding procedure (i.e., dummy coding). The injection of missing values previously described was performed independently for each of these sets, in order to ensure equal missing rates and MNAR assumptions for all of them. The experiment considered 5 missing rates (10%, 20%, 40%, 60% and 80%), which cover different magnitudes of the missing data issue. For the Autoencoder-based methods the missing values were pre-imputed with the mean of the respective features. To mitigate bias and stochastic behaviors, the entire pipeline

was executed in 30 independent runs, with the datasets being shuffled in each one. The results considered for comparison are the mean of these runs. Each run was executed in a computer running Windows 10, which had the following specifications: CPU AMD Ryzen 5600X, 16GB RAM, and GPU NVIDIA GeForce GTX 1060 6GB. The deep learning-based methods took roughly the same time to train in all settings (around four minutes per dataset), while the remaining methods took between one and five seconds to be executed.

The obtained results were analyzed from 3 different perspectives. The main analysis is a global perception of the imputation error provided by all methods, achieved by comparing the MAE results and validating their statistical significance. To complement this analysis, the previous results were also grouped by the number of datasets where each method outperformed the remaining. Finally, a study to identify the most suitable method for different distributions was conducted. These 3 different analyses are presented in the following sub-sections (5.2.1, 5.2.2 and 5.2.3).

5.2.1 Gross Imputation Error Comparison

Regarding the MAE produced by the imputation methods, the average results for all datasets are presented in Table 5.5. Moreover, Figure 5.2 displays the density plots of these results for each imputation method, considering the individual values obtained from each of the 30 runs, which allows for a more comprehensive analysis of the variance among all methods. In an overall analysis, the PMIVAE method surpassed the remaining ones, achieving an average MAE of 0.142, being followed by the VAE and SoftImpute methods with an average MAE of 0.161 and 0.175, respectively. When analyzing the results by missing rate, the PMIVAE method shows the lowest MAE values for the 20%, 40%, 60% and 80% missing rates. For the 10% rate it is surpassed by the SoftImpute and MICE methods.

Table 5.5: MAE results obtained for each imputation method and grouped by missing rate. The reported values are the averages and standard deviations of all datasets. The best results for each missing rate are bolded.

Imputation Method	**Missing Rate**				
	10%	**20%**	**40%**	**60%**	**80%**
DAE	0.280 ± 0.08	0.247 ± 0.07	0.203 ± 0.06	0.168 ± 0.05	0.155 ± 0.05
VAE	0.209 ± 0.10	0.181 ± 0.09	0.149 ± 0.07	0.133 ± 0.06	0.131 ± 0.06
kNN	0.195 ± 0.10	0.198 ± 0.11	0.224 ± 0.12	0.279 ± 0.14	0.342 ± 0.13
Mean	0.280 ± 0.10	0.275 ± 0.11	0.291 ± 0.12	0.321 ± 0.12	0.373 ± 0.12
MICE	0.192 ± 0.11	0.194 ± 0.12	0.217 ± 0.14	0.264 ± 0.15	0.319 ± 0.16
PMIVAE	0.195 ± 0.10	$\mathbf{0.165 \pm 0.09}$	$\mathbf{0.129 \pm 0.07}$	$\mathbf{0.111 \pm 0.06}$	$\mathbf{0.110 \pm 0.06}$
SoftImp	$\mathbf{0.173 \pm 0.09}$	0.167 ± 0.10	0.171 ± 0.11	0.180 ± 0.11	0.182 ± 0.10

To understand how these results change according to the size of the datasets, Table 5.6 presents the Pearson correlation coefficients calculated between the datasets' size and

the respective MAE values for the imputation methods. In this context, the size of the dataset is its total number of values ($size = \#instances * \#features$). All coefficients are negative, which indicates that the imputation results tend to be worse when the size of the dataset increases. However, the coefficients are all within $[-0.5, 0]$, therefore representing low correlation and similar consistence among the different imputation methods.

Table 5.6: Pearson correlation coefficients calculated between the size of the datasets and the respective MAE values for the imputation methods. The coefficients are independently presented for each missing rate.

Imputation Method	**Missing Rate**				
	10%	**20%**	**40%**	**60%**	**80%**
DAE	-0.39	-0.34	-0.32	-0.33	-0.31
VAE	-0.45	-0.42	-0.43	-0.45	-0.46
kNN	-0.31	-0.27	-0.26	-0.26	-0.26
Mean	-0.30	-0.27	-0.25	-0.24	-0.25
MICE	-0.32	-0.30	-0.28	-0.31	-0.30
PMIVAE	-0.40	-0.38	-0.35	-0.34	-0.35
SoftImp	-0.29	-0.25	-0.19	-0.15	-0.11

The benefits of using the PMIVAE method in MNAR scenarios with moderate and high missing rates are related to two main factors: the generative capabilities of the VAE, which lead to imputed values that have similar characteristics to the available data but are diverging towards new concepts, and the reduced level of uncertainty achieved through the partial MI procedure, which allows higher stability in the imputed values. Furthermore, the PMIVAE is surpassed in the 10% missing rate because the amount of information still available in the features is enough for a method more suitable for MAR scenarios (such as the SoftImpute or MICE) to still produce good results. In other words, the MNAR assumptions are weak in this low missing rate scenario, and therefore the level of uncertainty surrounding the missing values is lower since the features probably contain enough information to ease the imputation procedure.

An interesting aspect to highlight is that all Autoencoder-based methods reduce their MAE values when the missing rate increases, while the remaining methods present the opposite behavior. This a clear indicator that Autoencoders are suitable for MNAR scenarios, particularly with high missing rates, since they deal well with corrupted data, mostly because the additional noise leads to less overfitting and improved generalization [120].

To validate the statistical significance of the obtained results, the Three-Way ANOVA on ranks test was applied with a significance level of 5%. The factors considered were the dataset, the missing rate and the imputation method, while the dependent variable was the MAE. The standard Three-Way ANOVA test can only be applied if the normality assumptions are met. In this study such assumptions were violated, and therefore the data was transformed into rankings using the Ordered Quantile normalization [83], which always produces normally distributed transformed data. The same assumptions were

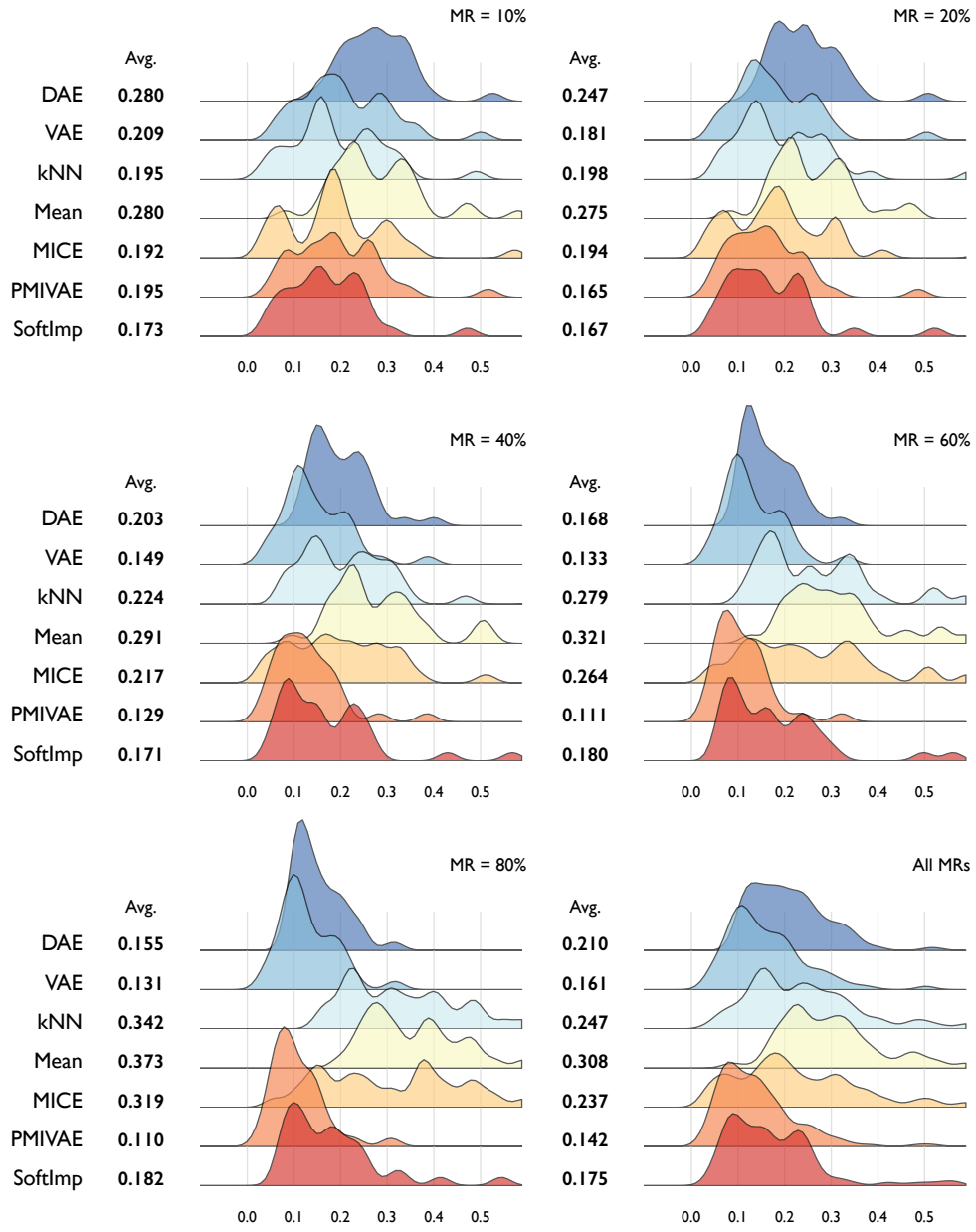

Figure 5.2: Density plots of the MAE results obtained for each imputation method. The analysis is presented by missing rate, with the overall results being displayed in the last plot. The average MAE for each setting is also displayed.

ensured for the data subgroups. From the obtained p-values we can conclude the results are statistically significant for all factors: the dataset and imputation method factors show $p = 2e^{-16}$, while the missing rate shows $p = 3.33e^{-06}$.

Finally, to conclude if the PMIVAE outperformed the remaining methods with a statistical significance of 5%, the post-hoc Tukey's HSD test was applied for the imputation method factor. The PMIVAE significantly outperformed all the remaining methods, showing $p < 0.001$ for all settings.

5.2.2 Best Imputation Method per Dataset

Table 5.7 presentes an alternative analysis where the results are grouped by the amount of datasets where each imputation method obtained the lowest MAE. We can conclude that PMIVAE was the overall best method in 71% of the datasets, followed by the SoftImpute method in 15% of them. When analyzing the results by missing rate, we see that PMIVAE also presents the best results, with an exception for the 10% missing rate where the SoftImpute surpassed it (the reasons behind this behavior are the same already presented for the MAE results analysis). Moreover, we see clear improvements in the PMIVAE results when the missing rates increase: for a 20% missing rate it is better in 56% of the datasets, for the 40% and 60% rates it is better in 88% of them, and finally for a 80% rate PMIVAE is better in 97% of the datasets.

Table 5.7: Percentage of datasets where each imputation method was the best. The data is grouped by missing rate, with the overall results being presented in the last column. The best results for each missing rate are bolded.

Imputation Method	**Missing Rate**					**Overall Balanced**
	10%	**20%**	**40%**	**60%**	**80%**	
DAE	0%	3%	3%	6%	3%	3%
VAE	0%	0%	0%	0%	0%	0%
kNN	9%	6%	0%	0%	0%	3%
Mean	0%	0%	0%	0%	0%	0%
MICE	18%	15%	6%	3%	0%	8%
PMIVAE	26%	**56%**	**88%**	**88%**	**97%**	**71%**
SoftImp	**47%**	21%	3%	3%	0%	15%

5.2.3 Sensitivity to Different Data Probability Distributions

Considering that our experimental setup addressed iteratively each feature of the 34 datasets, we decided to perform an analysis focused on the most suitable imputation methods for each Probability Density Function (PDF) available in the data. The goal was to understand if some imputation methods were more suitable for specific distributions. We developed an automated approach to find the PDF of each feature in each dataset, where the most well-known PDFs[3] were iteratively fitted to the data, and the resulting fitting error was calculated through the residual sum of squares between the distribution and the histogram of the original data. The PDF with the lowest error for each feature was chosen as the one that better describes it. Finally, we crossed this information with the best imputation method for each feature, leading to the results available in Figure 5.3. This heat-map presents the best imputation method for each PDF identified in the process previously described. To ensure the results were trustworthy, only distributions followed by at least 4 features were considered. The analysis was performed individually for each

[3]See `https://docs.scipy.org` for the list of PDFs considered.

missing rate, and each cell contains the percentage of features that lead each method to be the best.

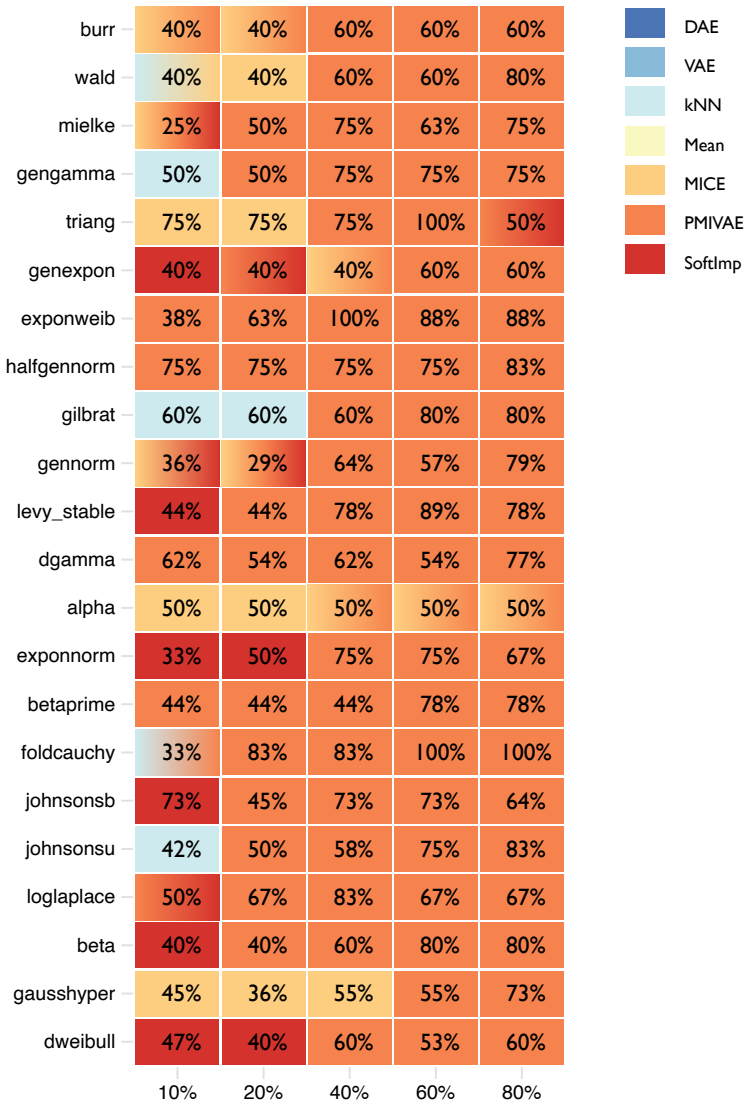

Figure 5.3: Heat-map of the best imputation methods for each PDF followed by, at least, 4 features of the 34 datasets. The results are displayed independently for each missing rate (x-axis), and the color legend is displayed in the top-right corner. When 2 methods are tied for the same PDF, a color gradient of both is used. The percentage of features that lead a method to be the best is displayed in each cell.

The obtained results showed a clear dominance of the PMIVAE method in all PDFs when the missing rate is 40% or higher. For the smaller missing rates (10% and 20%) the variance between the best methods is high, which can be explained by a smaller impact of the imputation procedure: the amount of values generated is not enough to have an impact on the distribution. Nevertheless, for moderate and high missing rates, PMIVAE shows stable results for all PDFs, therefore being a versatile method that can be applied on data with different characteristics.

5.3 Multivariate Experimental Results

Based on the encouraging results from the main experimental setup, we decided to extend these experiments to encompass a multivariate missing data setting. Furthermore, we also decided to compare the best method from the main experiments (PMIVAE) with two other state-of-the-art generative deep learning-based models: the Missing Data Importance-Weighted Autoencoder (MIWAE) [63] and the GAIN [126] methods. Both models were parametrized according to the best hyperparameters reported by their authors. In this multivariate scenario, the missing data generation follows the same MNAR pattern from the main experiments, but is now applied to the entire dataset. Therefore, the missing rate is global for the dataset, and several features are injected with missing values simultaneously. Such multivariate setting usually leads to increased imputation errors, and to understand if the models present large deviations from the ground truth in specific features we also calculated the Root Mean Squared Error (RMSE) metric. For the remaining aspects, these experiments followed the same configurations of the main experimental setup. The obtained results are presented in Table 5.8. The PMIVAE method surpassed the remaining two deep learning models in both MAE and RMSE metrics. PMIVAE was the best method in 75% of the settings combining the five levels of missingness and the 34 datasets. Such results show that the PMIVAE scales well for multivariate scenarios, while keeping the best imputation quality.

Table 5.8: MAE and RMSE results obtained for each deep-learning imputation method and grouped by missing rate. The reported values are the averages and standard deviations of all datasets. The best results for each missing rate are bolded.

Metric	**Imputation Method**	**Missing Rate**				
		10%	**20%**	**40%**	**60%**	**80%**
MAE	GAIN	0.317 ± 0.11	0.329 ± 0.11	0.389 ± 0.14	0.627 ± 0.34	$\mathbf{0.800 \pm 0.48}$
	MIWAE	0.409 ± 0.17	0.424 ± 0.18	0.492 ± 0.21	0.748 ± 0.42	0.959 ± 0.61
	PMIVAE	$\mathbf{0.285 \pm 0.10}$	$\mathbf{0.293 \pm 0.11}$	$\mathbf{0.354 \pm 0.15}$	$\mathbf{0.599 \pm 0.36}$	$\mathbf{0.800 \pm 0.55}$
RMSE	GAIN	0.453 ± 0.12	0.473 ± 0.13	0.558 ± 0.16	0.993 ± 0.62	$\mathbf{1.273 \pm 0.81}$
	MIWAE	0.560 ± 0.15	0.580 ± 0.17	0.673 ± 0.22	1.124 ± 0.67	1.472 ± 1.00
	PMIVAE	$\mathbf{0.424 \pm 0.11}$	$\mathbf{0.437 \pm 0.13}$	$\mathbf{0.523 \pm 0.18}$	$\mathbf{0.970 \pm 0.64}$	1.312 ± 0.98

5.4 A Case Study on Heart Failure

Encouraged by the positive results obtained from the experiments of Sections 5.2 and 5.3, we aimed to assess the impact of the imputation performed by the PMIVAE method in a classification problem. We focused this new experiment in a heart failure dataset available for medical research on the PhysioNet repository since late 2020 [34, 128]. This dataset was constructed with electronic health data collected from 2008 patients admitted with heart failure to a hospital in Sichuan, China, between December 2016 and June 2019. It contained 167 features (125 continuous and 42 categorical) covering regular clinical

characteristics (e.g., body temperature, pulse, respiration rate, systolic blood pressure, diastolic blood pressure, mean arterial blood pressure, weight, height, body mass index, type of heart failure, New York Heart Association cardiac function, Killip Grade and Glasgow Coma Scale) and several features extracted from echocardiograms performed on patients (e.g., left ventricular ejection fraction, left ventricular end diastolic diameter, mitral valve peak E and A wave velocities, tricuspid valve regurgitation velocity, and tricuspid valve regurgitation pressure). In a pre-processing stage, 3 features containing unique and constant values were dropped, since they were irrelevant for the classification task. A procedure to remove outliers from the continuous features was also applied, where the values below and above 3 standard deviations from the mean were removed.

The considered classification target was a binary label indicating if the patient would be readmitted to the hospital within 6 months after discharge. Heart failure is a frequent reason for hospitalization of elderly people that leads to a high mortality [128]. Therefore, having a decision support system that can help physicians know in advance what patients are more likely to be readmitted allows a better management of patients and discharges by the hospital systems.

Initially all the dataset instances contained missing values, with the missing rate range varying from 1% to 97.9% in different features. Although there is no clear way of identifying the missing mechanism behind missing data, we performed a few statistical and exploratory steps to find the more likely mechanisms. We started by applying the Little's test [53], which gave us a clear indication that the missing values were not under MCAR assumptions. We then performed exploratory data analysis, where we took into account meta-data from the features containing missing data and correlations with other features. After aggregating all the information, we concluded that most missing features are likely to be under MNAR assumptions.

In order to train the Autoencoder-based methods, a complete set of instances is required. Since all instances of the dataset had missing values, pre-imputation was needed. As in the previous experiments, the features' mean was used for this purpose.

To understand the impact on different types of classification algorithms, the 4 classifiers presented in Table 5.9 were considered in the study [11].

The remaining parameters used for the 4 classifiers are the recommended defaults by the Scikit-learn library. The dataset was divided into training and test sets in a stratified manner (70%-30% ratio), ensuring a balanced split between the label values in the two sets. Before training the classifiers, the SMOTE [19] over-sampling method was applied to solve the imbalance between labels on the training set (the global imbalance ratio between readmitted and other patients was 63%). To evaluate the classification results the F1 Score metric was considered. The same imputation methods used in the general experiments from Section 5.2 were used here, as well as the same parametrization, with a single excep-

Table 5.9: Classifiers used in the study and their configurations.

Classifier	Configuration
Artificial Neural Network (ANN)	The network had 2 hidden layers (with 100 and 50 nodes, respectively), ReLU as the activation function, Adam as the optimizer, 2000 epochs and 20% of the training samples for validation.
k-Nearest Neighbors (kNN)	$k = 5$ neighbors were considered.
Random Forest (RF)	100 trees were used.
Support Vector Machine (SVM)	RBF kernel and $\gamma = \frac{1}{N*\sigma^2}$ were used, being N the number of features.

tion: given the higher dimensionality of this heart failure dataset, the Autoencoder-based methods used two hidden layers (with 90 and 60 units), and a latent space with 30 units. The maximum number of epochs was also extended to 2000.

The experiment was executed 30 independent times, with the dataset being shuffled in each one. The obtained results, which are the mean of these 30 runs, are available in Table 5.10. This classification problem is very complex, leading in general to modest F1 Scores

Table 5.10: F1 Score results obtained for each imputation method applied to each classifier. The reported values are the averages and standard deviations of 30 runs. The best results for each classifier are bolded.

Imputation Method	Classifiers' F1 Score			
	ANN	kNN	RF	SVM
DAE	0.538 ± 0.02	0.476 ± 0.03	0.531 ± 0.02	$\mathbf{0.538 \pm 0.02}$
VAE	0.517 ± 0.03	0.471 ± 0.03	0.531 ± 0.02	0.523 ± 0.03
kNN	0.529 ± 0.02	$\mathbf{0.481 \pm 0.02}$	0.530 ± 0.02	0.534 ± 0.03
Mean/Mode	0.528 ± 0.02	0.480 ± 0.03	0.517 ± 0.02	0.530 ± 0.02
MICE	0.532 ± 0.03	0.478 ± 0.02	0.529 ± 0.02	0.531 ± 0.02
PMIVAE	$\mathbf{0.539 \pm 0.03}$	$\mathbf{0.481 \pm 0.02}$	$\mathbf{0.534 \pm 0.02}$	0.535 ± 0.03
SoftImp	0.523 ± 0.03	0.473 ± 0.03	0.532 ± 0.02	0.523 ± 0.02

(peaking at 0.539 for the ANN). Although the variance of results between the different imputation methods is small for all classifiers, we see improvements in 50% of them when PMIVAE is used (ANN and RF). For the kNN classifier, the best result is shared between the kNN imputation and PMIVAE. Only the SVM presented the best results with another imputation method (DAE). Therefore, in conclusion, we see general improvements when PMIVAE is used for the imputation task previous to the classification.

5.5 Conclusions

In this work a new method to perform missing data imputation, called Partial Multiple Imputation with Variational Autoencoders (PMIVAE), is introduced. PMIVAE is an extension to vanilla VAEs, and it applies a partial MI strategy to reduce bias and uncertainty in the new values generated. The method was compared with 8 state-of-the-art imputation approaches, in a very complete experimental setup that considered 34 public datasets from the medical context, missing data generation under MNAR assumptions and 5 missing rates between 10% and 80%. The results showed that the PMIVAE was the best overall method in 71% of datasets. Moreover, PMIVAE tends to perform better for moderate and high missing rates, being only surpassed in the 10% rate by the SoftImpute method. These results were validated with a statistical significance of 5%. The PMIVAE also surpassed the MIWAE and GAIN methods in a multivariate setting, achieving the best results for most missing rates and datasets. Furthermore, a case study of a classification task with a heart failure dataset was also performed. The PMIVAE method provided improvements in 50% of the considered classifiers.

In the future we want to address the limitation of only considering a single amputation procedure by testing the PMIVAE method with other strategies to generate missing values under MNAR assumptions. Furthermore, we want to extend PMIVAE by incorporating information from other datasets of the same context, which would allow adjustments to the distribution parameters of the VAE that could reduce bias in MNAR settings.

This page is intentionally left blank.

Chapter 6

Siamese Autoencoder-based Approach for Imputation

Several deep learning architectures have been explored to perform imputation assuming the different mechanisms. The state-of-the-art architectures in this scope are generative adversarial networks (GAN) [126], denoising autoencoders (DAE) [17, 118] and variational autoencoders (VAE) [65, 80]. In this work, we explore the use of siamese networks [21] to perform missing data imputation. We propose a new model called Siamese Autoencoder-based Approach for Imputation (SAEI), which extends and adapts the vanilla siamese network for the imputation task by adding a deep autoencoder architecture, a custom loss function and a custom triplet mining strategy. We compared our SAEI model with a baseline of seven state-of-the-art imputation methods in a experimental setup encompassing 14 datasets of the healthcare domain injected with MNAR values at different missingness levels (10% to 60% missing rates). The achieved results proved that our SAEI model significantly outperformed the remaining baseline methods in all experimented settings, achieving an average improvement of 35%.

The remainder of the chapter is organized in the following way: Section 6.1 describes with detail the proposed SAEI model; Section 6.2 presents the experimental setup used to validate the effectiveness of our SAEI model, and analyzes the results obtained from the experiments; and Section 6.3 presents our conclusions and the future work.

6.1 Proposed Approach

The Siamese Autoencoder-based Approach for Imputation (SAEI) is an extension of a vanilla siamese network, and it is tailored for missing data imputation by comprising three adaptations:

- A deep autoencoder architecture that allows the network to reproduce the input data at the output layer in an unsupervised fashion;
- A custom loss function that includes both the distance-based triplet loss and the reconstruction error of the autoencoder component of the network;
- A custom triplet mining strategy that was designed specifically for the missing data issue by creating hard triplets based on the existing missing values.

These three adaptations are independently described in the next subsections.

6.1.1 Deep Autoencoder Architecture

The architecture of our SAEI model is roughly inspired in the well-know ZFNet [127], although it presents several changes motivated by it being aimed at tabular data. Figure 6.1 and Figure 6.2 depict graphical descriptions of the encoder and decoder networks, respectively. The encoder network is composed of two one-dimensional convolutional layers with 16 filters, ReLU as the activation function, and kernel sizes of five and three. Such layers are followed by max-pooling layers with two strides, which are then followed by two regularization layers that perform batch normalization and dropout at a rate of 25%. Moreover, a residual connection is also used to skip the second convolutional layer. Finally, the output from the last pooling layer is flattened and passed through a hyperbolic tangent activation function, and the latent output is obtained from a final dense layer with 128 units and the latter activation function. The decoder network is symmetric to the encoder, presenting the same architecture but in reserve order (without the residual connection). To perform the deconvolution operation, the one-dimensional convolutional and the max pooling layers are replaced by one-dimensional transposed convolutional layers with two strides. The output dense layer uses the sigmoid activation function so that the data can be normalized within $[0, 1]$.

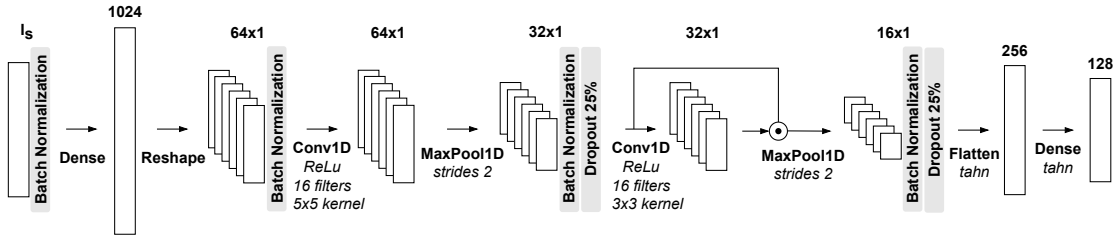

Figure 6.1: SAEI encoder architecture. I_s represents the input shape. The encoder network is composed of two one-dimensional convolutional layers with 16 filters followed by max-pooling layers with two strides. Regularization is performed through batch normalization and dropout at a rate of 25%. A residual connection is also used two skip the second convolutional layer. The latent output is obtained from a dense layer with 128 units.

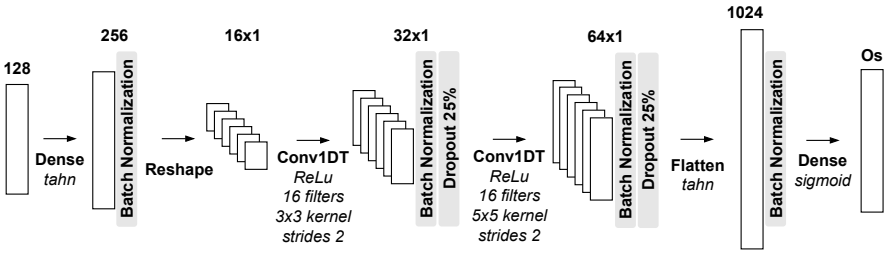

Figure 6.2: SAEI decoder architecture. O_s represents the output shape. The decoder network is symmetric to the encoder. Therefore, it has the same layers and regularization but in reserve order. The residual connection is not applied by the decoder.

Convolutional layers operate based on the spatial positioning of the data. In other words, the position where each value is placed is relevant for the feature extraction process. Tabular data does not present this behavior because the features' positions are irrelevant. To surpass this limitation, the SAEI model feeds the input data to a dense layer with 1024 units and without activation function. The purpose of this layer is to learn an abstract representation of the input data where the spatial relation between the new abstract features is meaningful. Therefore, our SAEI model delegates the task of learning the spatial structure of the features to the network. The first convolutional layer is then fed with the output of the mentioned dense layer.

6.1.2 Custom Loss Function

One of the original loss functions proposed to train a siamese network was the triplet loss. It tries to minimize the distance between an anchor and a positive sample which represent the same concept, while maximizing the distance between that same anchor and a negative sample of a different concept. The formulation is presented in Eq. 6.1, where $f(x)$ is a function of the embedding representation, N is the number of triples composed by an anchor (x_i^a), a positive sample (x_i^p) and a negative one (x_i^n) [101]. Furthermore, α is a margin used to ensure a minimum distance between positive and negative samples.

$$TL_i = \sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \tag{6.1}$$

Using the triplet loss function is ideal for a vanilla siamese network since the goal is for the model to distinguish between positive and negative concepts in comparison to the anchor, while outputting embedding representations that incorporate such differences. However, our SAEI model relies on an autoencoder architecture that outputs a reconstruction of the input data based on the anchor latent representation. Therefore, this reconstruction component must also be reflected in the loss function of the model. Eq. 6.2 presents our custom loss function, where the triplet loss (TL_i) from Eq. 6.1 is added to the mean

squared error between the anchor and positive sample, similarly to what would happen in a denoising autoencoder (i.e., the anchor is the corrupted version of the ground truth represented by the positive sample).

$$TL_i + \sum_{i}^{N} (\hat{x_i^a} - x_i^p)^2 \tag{6.2}$$

6.1.3 Custom Triplet Mining

The triplet selection is a key step for successfully training a siamese network. To ensure that the network is able to learn how to distinguish between the positive and negative concepts, it is imperative to select triplets that the network is unable to differentiate before being trained. These triplets are composed by the so-called hard positives and hard negatives, which are samples that violate the constraint of the triplet loss, presented in Eq. 6.3 [101]. If the network is trained with easy triples where the triplet loss constraint is already satisfied before training, it would not gain the capacity of distinguishing between positive and negative concepts.

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{6.3}$$

When extending this type of network and its respective training procedure to address missing data imputation, the definitions of anchor, negative and positive samples must be redefined within the missing data scope. Our SAEI model proposes the following definitions:

- Anchors are the samples containing missing values, which are pre-imputed with the mean of the available values in each feature (or the mode in categorical features). This pre-imputation step is required since neural networks are unable to be trained with data containing missing values;
- Positives are the original samples without containing missing values (i.e., the ground truth of the anchor samples). This implies that we must have a portion of complete data to train the SAEI model, but such assumption is common to all deep learning-based imputation methods. Furthermore, the anchor samples are created by artificially generating missing values in the positive samples, according to a pre-established missing data mechanism;
- Negatives are the same as anchor samples but with the missing values being replaced by Gaussian noise sampled according to Eq. 6.4, where D represents the entire dataset and λ represents the variance of the noise domain.

$$\mathcal{N}\left(\frac{\max(D) - \min(D)}{2}, \lambda\right) \tag{6.4}$$

The rational behind the mean value is to use the midpoint within the domain of the data, therefore keeping the Gaussian noise centered on that domain. Such strategy works specially well when the data is normalized within a specific range (e.g., $[0, 1]$). Moreover, the λ parameter acts as a control variable to define how hard or easy should the triplets be: a low λ value leads to Gaussian noise mostly or completely contained within the data domain, therefore leading to harder triplets since the negative sample is likely to be partially overlapped with the positive one; on the other hand, a high λ value increases the noise domain and creates a negative sample that is more dissimilar to the positive, therefore creating easier triplets where the negative and positive samples are immediately distinguishable by the model. As a consequence, this parameter should be defined aiming to generate hard triplets while considering for the data domain.

6.2 Experimental Results

The quality of the imputation results achieved by the SAEI model was assessed by comparing it with a baseline of seven state-of-the-art imputation methods: kNN with $k = 5$, Mean/Mode, MICE with 100 iterations, SoftImpute with 100 iterations, DAE, VAE and GAIN. The DAE and VAE were defined according to the following architecture and hyperparameters: a hidden layer with half of the input dimension and ReLU as the activation function (although VAE has two additional layers for the Gaussian distribution parameters), Adam as the optimizer with a learning rate of 0.001, Sigmoid as the activation function of the output layer (forcing the data to be normalized within $[0, 1]$), batches of 64 samples, 200 training epochs, a dropout rate of 10% for regularization, Mean Squared Error as the reconstruction loss, and early stop and learning rate reduction by 80% if the validation loss does not improve over 100 epochs. The VAE loss function also includes the Kullback–Leibler divergence for regularization. Furthermore, both DAE and VAE require pre-imputation, which is performed with the mean/mode of the features. The GAIN method was used with the hyperparameters proposed by its authors. Our SAEI model follows the architecture described in Section 6.1.1 while all the remaining hyperparameters are the same used by the DAE and VAE. Moreover, the Gaussian noise of the negative instances was sampled from $\mathcal{N}(0.5,\, 0.05^{0.5})$, and the margin of the triplet loss function was set to $\alpha = 0.2$ as the original authors proposed. With the exception of GAIN, the described hyperparameters of all methods were defined through a grid search process. This is a standard procedure that aims to achieve hyperparameters that conform to common use cases. Moreover, the DAE and VAE were implemented with the Keras library, the GAIN implementation was obtained from its original authors[1], and the remaining methods were obtained and used from the Scikit-learn library. Our SAEI model was also coded with the

[1] `https://github.com/jsyoon0823/GAIN`

Keras library and is available on GitHub[2].

The experiments were conducted over 14 datasets of the healthcare domain which cover different types of clinical data that was collected for several pathologies. We chose to cover this medical domain since it often suffers from the missing data issue, particularly missing values under the MNAR mechanism, which creates deep challenges to any subsequent analysis or task performed with the data [74]. In this health context, each instance usually represents a group of values collected for a patient, and each value belongs to a feature that could be a measurement, an exam result, or any other medical input. The missing values may appear at any feature and/or instance. All the 14 datasets are public and available at the UC Irvine Machine Learning[3] and Kaggle[4] repositories, and they present heterogeneous characteristics as seen in Table 6.1.

Table 6.1: Datasets characteristics.

Dataset	**# Instances**	**# Features**	
		Categorical	**Continuous**
diabetic-retinopathy	1151	4	16
ecoli	336	1	7
ctg-2c	2126	1	21
new-thyroid-N-vs-HH	215	1	5
kala-azar	68	2	5
immunotherapy	90	3	5
saheart	462	2	8
bc-coimbra	116	1	9
cleveland-0-vs-4	173	1	13
newthyroid-v1	185	1	5
biomed	194	1	5
cryotherapy	90	3	4
thyroid-3-vs-2	703	1	21
pima	768	1	8

The datasets were normalized within $[0, 1]$ and split into train and test sets with 70% and 30% of the instances, respectively. The scaler learns the minimum and maximum values from the train set and transforms both sets. This strategy ensures the test data is not biased with information from the training data. However, in the presence of high missing rates (usually above 50%), the test set may contain values unseen by the scaler, which leads to the normalization boundaries being slightly extended from the expected $[0, 1]$ domain. Also, the autoencoder-based methods use 20% of the train set as the validation set. Furthermore, the categorical nominal features were transformed through the one-hot

[2] `https://github.com/ricardodcpereira/SAEI`
[3] `https://archive.ics.uci.edu/ml`
[4] `https://www.kaggle.com/datasets`

encoding procedure so that they can be supported by all imputation methods. Finally, in order to be able to calculate the imputation error of the experimented methods, the datasets must be complete and the test set must be injected with artificially generated missing values so that we can compare the estimated values with the original ones (i.e., ground truth). We choose to generate MNAR values since it is the hardest mechanism to address and the one more often found in real-world contexts such as healthcare [74, 80]. Our generation strategy is based on removing the smaller values of several features at once (upon a certain missing rate) in a multivariate procedure. Therefore, the missing rate is defined for the entire dataset and the imputation is performed on all features simultaneously. The obtained results were evaluated with the Mean Absolute Error (MAE) between the estimates and the ground truth values. Regarding the missing rate, we considered four levels of missingness: 10%, 20%, 40% and 60%.

To avoid the impact of stochastic behaviors in the results, the experiment was executed 30 independent times, with the train/test split being randomly performed in every run. The MAE results of the experiment are the average of the 30 runs. Moreover, each run was executed in a computer with the following specifications: Windows 11, CPU AMD Ryzen 5600X, 16GB RAM, and GPU NVIDIA GeForce GTX 1060 6GB. The time complexity of each imputation method was not measured, but all methods were computed in a feasible amount of time. The imputation results obtained from the experimental setup are displayed with detail in Table 6.2. The MAE results (average and standard deviation of the 30 independent runs) are individually displayed for each dataset and missing rate. Furthermore, the overall results for each imputation method and per missing rate (i.e., considering all datasets) are displayed in Figure 6.3.

In an overall analysis, our SAEI model clearly outperforms all the remaining imputation methods, achieving smaller MAE values for every dataset and missing rate considered, as seen in Table 6.2. In fact, the average improvement of the SAEI model in comparison to the remaining baseline methods is 35%, peaking at the 20% missing rate with an improvement of 42%. For the lower missing rate (10%) the improvement is 36%. For the higher missing rates (40% and 60%), the improvement rates are 39% and 24%, respectively. Such results show the consistency of our SAEI model with different levels of missingness and with data presenting different characteristics.

Table 6.2: Experimental results per dataset and missing rate (average ± standard deviation of MAE). The best results are bolded and highlighted.

Dataset	MR (%)	kNN	Mean/Mode	MICE	SoftImpute	DAE	VAE	GAIN	SAEI
diabetic-retinopathy	10	0.133 ± 0.02	0.194 ± 0.01	0.119 ± 0.02	0.165 ± 0.02	0.152 ± 0.01	0.178 ± 0.01	0.192 ± 0.02	**0.109 ± 0.01**
	20	0.157 ± 0.02	0.213 ± 0.02	0.128 ± 0.02	0.185 ± 0.03	0.164 ± 0.02	0.175 ± 0.01	0.216 ± 0.03	**0.100 ± 0.01**
	40	0.207 ± 0.03	0.252 ± 0.03	0.179 ± 0.03	0.236 ± 0.03	0.199 ± 0.02	0.163 ± 0.01	0.264 ± 0.04	**0.131 ± 0.01**
	60	0.324 ± 0.05	0.358 ± 0.05	0.292 ± 0.05	0.354 ± 0.04	0.298 ± 0.05	0.242 ± 0.04	0.367 ± 0.06	**0.231 ± 0.04**
ecoli	10	0.213 ± 0.03	0.304 ± 0.04	0.229 ± 0.03	0.286 ± 0.04	0.265 ± 0.03	0.328 ± 0.03	0.337 ± 0.06	**0.173 ± 0.03**
	20	0.224 ± 0.04	0.309 ± 0.04	0.236 ± 0.04	0.297 ± 0.04	0.257 ± 0.03	0.306 ± 0.03	0.327 ± 0.06	**0.151 ± 0.03**
	40	0.363 ± 0.08	0.413 ± 0.07	0.363 ± 0.08	0.367 ± 0.05	0.347 ± 0.07	0.352 ± 0.05	0.456 ± 0.08	**0.229 ± 0.06**
	60	0.750 ± 0.21	0.759 ± 0.20	0.730 ± 0.20	0.703 ± 0.18	0.674 ± 0.21	0.672 ± 0.20	0.800 ± 0.22	**0.589 ± 0.20**
ctg-2c	10	0.151 ± 0.02	0.288 ± 0.03	0.137 ± 0.02	0.155 ± 0.02	0.188 ± 0.02	0.255 ± 0.02	0.261 ± 0.02	**0.121 ± 0.02**
	20	0.191 ± 0.02	0.296 ± 0.02	0.161 ± 0.01	0.167 ± 0.01	0.201 ± 0.02	0.227 ± 0.01	0.271 ± 0.02	**0.116 ± 0.01**
	40	0.318 ± 0.03	0.362 ± 0.03	0.286 ± 0.04	0.233 ± 0.02	0.293 ± 0.04	0.205 ± 0.02	0.347 ± 0.03	**0.191 ± 0.03**
	60	0.498 ± 0.05	0.526 ± 0.05	0.488 ± 0.05	0.399 ± 0.05	0.508 ± 0.11	0.356 ± 0.06	0.511 ± 0.06	**0.354 ± 0.06**
new-thyroid-N-vs-HH	10	0.264 ± 0.07	0.270 ± 0.05	0.266 ± 0.08	0.258 ± 0.05	0.272 ± 0.06	0.413 ± 0.04	0.361 ± 0.08	**0.124 ± 0.04**
	20	0.273 ± 0.04	0.264 ± 0.04	0.256 ± 0.05	0.227 ± 0.04	0.238 ± 0.04	0.408 ± 0.04	0.369 ± 0.08	**0.143 ± 0.03**
	40	0.269 ± 0.04	0.296 ± 0.04	0.269 ± 0.04	0.204 ± 0.03	0.247 ± 0.04	0.427 ± 0.04	0.392 ± 0.09	**0.156 ± 0.03**
	60	0.355 ± 0.10	0.432 ± 0.09	0.391 ± 0.09	0.257 ± 0.07	0.374 ± 0.11	0.502 ± 0.07	0.532 ± 0.11	**0.239 ± 0.12**
kala-azar	10	0.317 ± 0.07	0.358 ± 0.07	0.288 ± 0.06	0.300 ± 0.07	0.356 ± 0.06	0.488 ± 0.09	0.398 ± 0.08	**0.217 ± 0.06**
	20	0.409 ± 0.05	0.401 ± 0.05	0.369 ± 0.04	0.357 ± 0.05	0.404 ± 0.07	0.508 ± 0.05	0.455 ± 0.07	**0.232 ± 0.05**
	40	0.468 ± 0.07	0.472 ± 0.07	0.468 ± 0.08	0.408 ± 0.04	0.441 ± 0.09	0.589 ± 0.07	0.540 ± 0.12	**0.277 ± 0.05**
	60	0.951 ± 0.35	0.952 ± 0.35	0.953 ± 0.35	0.805 ± 0.31	0.908 ± 0.35	1.025 ± 0.34	0.983 ± 0.35	**0.796 ± 0.36**
immunotherapy	10	0.346 ± 0.08	0.384 ± 0.08	0.341 ± 0.09	0.362 ± 0.08	0.348 ± 0.08	0.433 ± 0.09	0.388 ± 0.09	**0.285 ± 0.08**
	20	0.370 ± 0.07	0.372 ± 0.07	0.349 ± 0.07	0.384 ± 0.06	0.337 ± 0.06	0.447 ± 0.06	0.386 ± 0.07	**0.268 ± 0.06**
	40	0.463 ± 0.08	0.463 ± 0.08	0.460 ± 0.09	0.469 ± 0.07	0.396 ± 0.08	0.524 ± 0.07	0.498 ± 0.08	**0.345 ± 0.08**
	60	1.026 ± 0.38	1.028 ± 0.38	1.015 ± 0.36	0.958 ± 0.36	0.929 ± 0.36	1.084 ± 0.36	1.027 ± 0.38	**0.871 ± 0.36**
saheart	10	0.289 ± 0.03	0.367 ± 0.03	0.295 ± 0.03	0.245 ± 0.03	0.315 ± 0.03	0.344 ± 0.03	0.387 ± 0.05	**0.231 ± 0.03**
	20	0.335 ± 0.04	0.389 ± 0.04	0.336 ± 0.03	0.268 ± 0.03	0.336 ± 0.04	0.332 ± 0.02	0.410 ± 0.05	**0.216 ± 0.03**
	40	0.464 ± 0.07	0.486 ± 0.06	0.464 ± 0.07	0.366 ± 0.05	0.442 ± 0.07	0.361 ± 0.04	0.531 ± 0.07	**0.296 ± 0.06**
	60	0.876 ± 0.28	0.884 ± 0.28	0.869 ± 0.28	0.732 ± 0.27	0.854 ± 0.29	0.729 ± 0.29	0.934 ± 0.29	**0.677 ± 0.26**
bc-coimbra	10	0.232 ± 0.04	0.300 ± 0.05	0.232 ± 0.04	0.209 ± 0.03	0.328 ± 0.04	0.464 ± 0.04	0.326 ± 0.06	**0.177 ± 0.05**
	20	0.277 ± 0.03	0.317 ± 0.03	0.269 ± 0.03	0.226 ± 0.02	0.319 ± 0.06	0.482 ± 0.02	0.357 ± 0.05	**0.175 ± 0.03**
	40	0.391 ± 0.06	0.411 ± 0.06	0.390 ± 0.07	0.293 ± 0.05	0.363 ± 0.08	0.559 ± 0.05	0.427 ± 0.08	**0.221 ± 0.05**
	60	0.661 ± 0.15	0.691 ± 0.16	0.656 ± 0.14	0.499 ± 0.13	0.675 ± 0.14	0.799 ± 0.14	0.697 ± 0.17	**0.459 ± 0.15**
cleveland-0-vs-4	10	0.340 ± 0.04	0.381 ± 0.04	0.346 ± 0.04	0.273 ± 0.04	0.332 ± 0.04	0.406 ± 0.04	0.347 ± 0.05	**0.255 ± 0.05**
	20	0.358 ± 0.04	0.401 ± 0.04	0.364 ± 0.03	0.280 ± 0.03	0.305 ± 0.03	0.399 ± 0.02	0.383 ± 0.04	**0.224 ± 0.03**
	40	0.526 ± 0.08	0.550 ± 0.08	0.526 ± 0.08	0.412 ± 0.08	0.434 ± 0.11	0.499 ± 0.09	0.524 ± 0.09	**0.303 ± 0.09**
	60	0.917 ± 0.17	0.931 ± 0.16	0.919 ± 0.17	0.761 ± 0.17	0.884 ± 0.18	0.922 ± 0.19	0.920 ± 0.17	**0.733 ± 0.22**
newthyroid-v1	10	0.250 ± 0.05	0.303 ± 0.05	0.249 ± 0.06	0.271 ± 0.06	0.295 ± 0.06	0.408 ± 0.04	0.368 ± 0.09	**0.157 ± 0.04**
	20	0.268 ± 0.05	0.300 ± 0.04	0.257 ± 0.05	0.302 ± 0.05	0.254 ± 0.05	0.409 ± 0.05	0.396 ± 0.11	**0.155 ± 0.03**
	40	0.299 ± 0.05	0.327 ± 0.05	0.311 ± 0.05	0.335 ± 0.05	0.240 ± 0.04	0.434 ± 0.04	0.449 ± 0.13	**0.161 ± 0.03**
	60	0.444 ± 0.15	0.494 ± 0.15	0.468 ± 0.15	0.437 ± 0.10	0.351 ± 0.12	0.554 ± 0.12	0.571 ± 0.19	**0.290 ± 0.12**
biomed	10	0.210 ± 0.05	0.288 ± 0.04	0.208 ± 0.05	0.277 ± 0.07	0.232 ± 0.04	0.409 ± 0.04	0.379 ± 0.10	**0.159 ± 0.03**
	20	0.228 ± 0.04	0.299 ± 0.03	0.211 ± 0.04	0.255 ± 0.04	0.217 ± 0.04	0.418 ± 0.03	0.381 ± 0.10	**0.149 ± 0.02**
	40	0.283 ± 0.04	0.343 ± 0.04	0.261 ± 0.04	0.274 ± 0.04	0.257 ± 0.05	0.448 ± 0.04	0.424 ± 0.08	**0.179 ± 0.03**
	60	0.474 ± 0.11	0.513 ± 0.11	0.448 ± 0.11	0.383 ± 0.10	0.399 ± 0.14	0.573 ± 0.12	0.572 ± 0.14	**0.335 ± 0.14**
cryotherapy	10	0.302 ± 0.08	0.371 ± 0.10	0.317 ± 0.09	0.295 ± 0.09	0.354 ± 0.10	0.453 ± 0.08	0.412 ± 0.11	**0.267 ± 0.09**
	20	0.347 ± 0.09	0.415 ± 0.07	0.339 ± 0.08	0.361 ± 0.07	0.401 ± 0.07	0.465 ± 0.08	0.425 ± 0.10	**0.284 ± 0.06**
	40	0.497 ± 0.10	0.521 ± 0.10	0.477 ± 0.10	0.455 ± 0.09	0.464 ± 0.09	0.573 ± 0.10	0.547 ± 0.11	**0.390 ± 0.10**
	60	1.373 ± 1.05	1.396 ± 1.05	1.371 ± 1.05	1.252 ± 1.02	1.352 ± 1.07	1.419 ± 1.06	1.392 ± 1.06	**1.218 ± 1.03**
thyroid-3-vs-2	10	0.094 ± 0.01	0.099 ± 0.01	0.085 ± 0.01	0.059 ± 0.01	0.068 ± 0.01	0.095 ± 0.01	0.111 ± 0.03	**0.045 ± 0.01**
	20	0.106 ± 0.02	0.109 ± 0.02	0.099 ± 0.02	0.062 ± 0.01	0.072 ± 0.02	0.092 ± 0.01	0.119 ± 0.03	**0.041 ± 0.01**
	40	0.137 ± 0.02	0.135 ± 0.02	0.138 ± 0.02	0.069 ± 0.01	0.093 ± 0.02	0.086 ± 0.01	0.145 ± 0.03	**0.051 ± 0.01**
	60	0.226 ± 0.04	0.224 ± 0.04	0.222 ± 0.04	0.112 ± 0.04	0.170 ± 0.06	0.119 ± 0.04	0.234 ± 0.05	**0.110 ± 0.04**
pima	10	0.266 ± 0.03	0.338 ± 0.03	0.273 ± 0.02	0.239 ± 0.02	0.297 ± 0.03	0.306 ± 0.03	0.384 ± 0.07	**0.207 ± 0.03**
	20	0.275 ± 0.02	0.336 ± 0.03	0.279 ± 0.02	0.234 ± 0.02	0.290 ± 0.03	0.269 ± 0.02	0.368 ± 0.06	**0.168 ± 0.02**
	40	0.360 ± 0.03	0.391 ± 0.04	0.354 ± 0.04	0.281 ± 0.03	0.341 ± 0.04	0.261 ± 0.03	0.452 ± 0.08	**0.199 ± 0.03**
	60	0.569 ± 0.12	0.575 ± 0.12	0.558 ± 0.12	0.435 ± 0.10	0.526 ± 0.13	0.406 ± 0.09	0.628 ± 0.14	**0.392 ± 0.13**

In order to validate if the obtained results were statistically significant, we applied the Three-Way ANOVA on ranks test with a significance level of 5%. We considered as factors the dataset, the missing rate and the imputation method, while the MAE was set as the dependent variable. The normality assumptions were not met, so the data was transformed into rankings using the Ordered Quantile normalization [83]. Such assumptions were also ensured for the data subgroups. The obtained p-values show that the results are statistically significant for all factors, with $p < 0.001$. Additionally, to validate if our

SAEI model outperformed the remaining methods with a statistical significance of 5%, the post-hoc Tukey's HSD test was applied to this factor. The obtained p-values show that the SAEI model significantly outperformed all the baseline of imputation methods, with $p < 0.001$ in all scenarios.

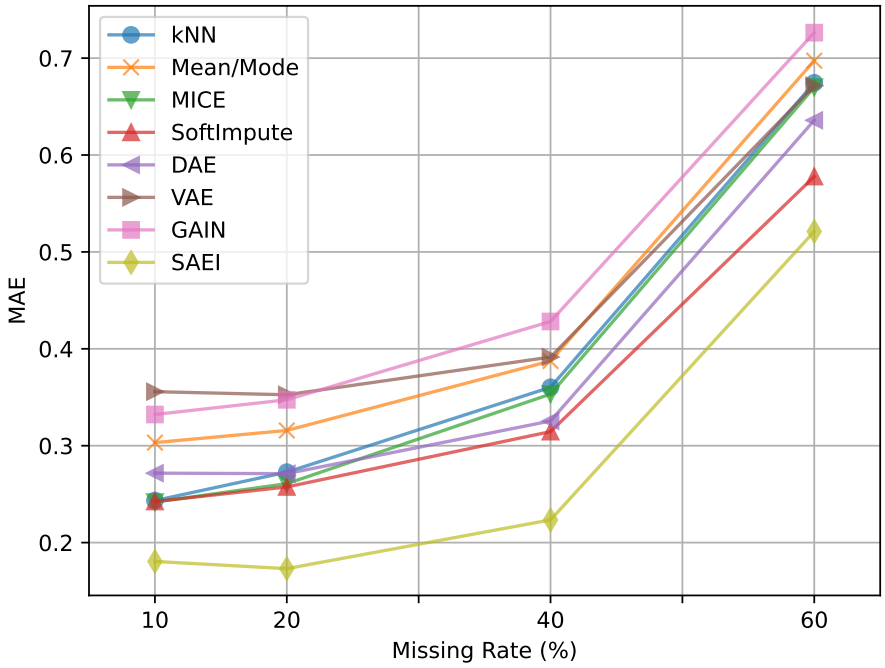

Figure 6.3: Overall Mean Absolute Error of all imputation methods per missing rate. Our SAEI model significantly outperforms the remaining methods in all settings.

6.3 Conclusions

In this work we propose a new model for missing data imputation called Siamese Autoencoder-based Approach for Imputation (SAEI), which is an extension of the vanilla siamese networks adapted for this imputation task. The model incorporates three main adaptations: a deep autoencoder architecture that is tailored for missing data reconstruction, a custom loss function that encompasses both the triplet and reconstruction losses, and a triplet mining strategy tailored for the missing data issue that is capable of generating hard triplets that are meaningful for the training procedure. To the best of our knowledge, this is the first time a siamese architecture is being used and adapted for missing data imputation. We compared our SAEI model with seven state-of-the-art imputation methods from both statistical and machine learning backgrounds, in an experimental setup that used 14 datasets of the healthcare domain injected with MNAR values in a multivariate fashion at 10%, 20%, 40% and 60% missing rates. Our SAEI model significantly outperformed

all the baseline methods used for comparison in all the datasets and missing rates, with a statistical significance of 5%, achieving an average improvement of 35%.

In the future we want to extend this work to test other missing data mechanisms, and we want to explore automatic approaches to evolve our deep autoencoder architecture so it can be even further optimized.

Part II

New Trends for Data Missing Not At Random

This page is intentionally left blank.

Chapter 7

Imputation with Distributed Data

To improve the imputation for the MNAR mechanism other sources of information of the same context can be used to help complete the missing values, since these sources may contain the related unobserved data. Having this idea in mind, the following research questions can arise:

- Could the error of the imputation methods used to estimate missing values under the MNAR mechanism decrease if they use more information of the same data context?
- Could the performance of the classifiers trained by this imputed data improve their precision and recall?

This work presents an experimental design that aims to answer these research questions. The data processing steps and the experimental pipeline are described with detail. The simulation results show that using other sources of information from the same context reduces the imputation error. Moreover, the classification performance is not biased by this imputation, and in fact improves with a statistical significance of 5%.

The remainder of the chapter is organized in the following way: Section 7.1 describes the design of the experiment, Section 7.2 presents the obtained results and their discussion, and Section 7.3 the conclusions and future directions of this work.

7.1 Experimental Design

An experimental design was proposed to prove the impact of using different sources of the same data context in the imputation quality and classification performance. The imputation evaluation is assessed through the Root Mean Square Error (**RMSE**) metric and the classification evaluation through the **F1 score**. In the experiment the following variables were considered:

- Features Similarity Rate ($FSRate$), which indicates the percentage of equal features that the datasets must have to be properly combined, and has the values 40%, 50%, 60% and 70%;
- Number of datasets ($NumDatasets$), which indicates how many datasets are being combined, and the values can vary between 1 and 5 (for the imputation evaluation the value 1 is discarded since no missing values exist, which leads to a constant RMSE value of 0);
- Imputation algorithms ($Algorithm$), which are the methods that generate the plausible values to replace the missing ones, and the algorithms used are the Mean Imputation, Multivariate Imputation by Chained Equations (MICE), K-Nearest Neighbors (KNN) Imputation, SoftImpute (SI) and Support Vector Regression (SVR).

Based on the presented variables, an experimental setup with the following pipeline of tasks was proposed (see Figure 7.1):

- A given database with F features and O observations is divided in N smaller datasets:
  - The observations are randomly divided in equal parts for all datasets, which means that each one will contain $\frac{O}{N}$ rows;
  - A percentage of the features is kept equal for all datasets ($FSRate$). These $FSRate{*}F$ features are chosen randomly from the total. The remaining features will only exist in one of the smaller datasets and are assigned sequentially and circularly.
- Each smaller dataset is combined gradually with the remaining ones by being concatenated with each one through $N-1$ iterations (e.g., for $N=4$ the combinations for the first smaller dataset would be $\{D1, D2\}$, $\{D1, D2, D3\}$ and $\{D1, D2, D3, D4\}$ over three iterations). This entire process is repeated N times, one for each of the datasets, and the final results are an average of these repetitions;
- Since all datasets have a set of features that are unique, the combination of two datasets generates missing values under MNAR for those features on the resulting dataset. Therefore, for each of the combinations described in the previous item, the missing values are predicted using 5 imputation algorithms, being the quality evaluated through the RMSE:
  - Mean Imputation, that uses the mean of the features to fill in the missing values;
  - Multivariate Imputation by Chained Equations (MICE) with 100 iterations, which uses a multiple imputation approach to create several regression models for the missing features based on the complete ones [5];

  - K-Nearest Neighbors (KNN) Imputation with $K = 1$, which finds the nearest neighbor through the Euclidean distance and uses its value of the missing feature to fill in the observation [95];
  - SoftImpute (SI), which is a matrix completion method based on nuclear-norm regularization [37];
  - Support Vector Regression (SVR), which is an adaptation of the well-known Support Vector Machines that can be applied for regression instead of classification, and it was configured with a RBF kernel and $\gamma = \frac{1}{\#Features}$ [104].

- The imputed datasets by each of the described algorithms are also used to train a classifier with the purpose of evaluating the impact of the imputation in the learning capabilities of the model. In this experiment a Decision Tree was used since it is widely applied in healthcare contexts for interpretability and explainability reasons. The quality of the classification is evaluated through the F1 score using hold-out validation (75% of the data for training and 25% for testing).

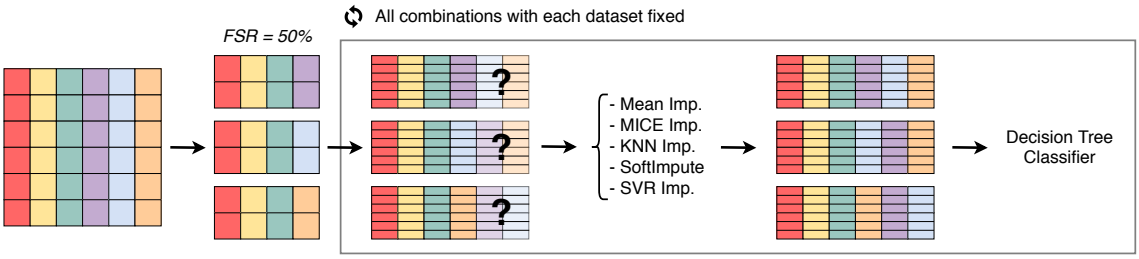

Figure 7.1: Pipeline of tasks for the experiment.

As stated before, merging the small datasets generates MNAR values because these missingness of a specific dataset is related with the information of the remaining ones. In other words, the missing values are related with the unobserved data that is being merged, and therefore are MNAR. The fusion approach here proposed is an attempt to mitigate the MNAR assumptions by transforming the MNAR mechanism into MAR, since the unobserved data will become observed after the fusion process. For that reason, the state-of-the-art imputation algorithms should now provide better and less biased results. Moreover, since the small datasets are generated from a complete one, the ground truth is available for error and accuracy calculations.

To reduce bias and mitigate stochastic behaviors the described pipeline was executed 100 times, being the average of the 100 results considered for analysis purposes. The pipeline was implemented through several Python scripts using the Pandas[1] framework to process the data and the scikit-learn[2] and fancyimpute[3] frameworks to assist the implementation of the imputation methods and the classifier. The experiment was conducted using 15

[1]Available at `https://pandas.pydata.org/`
[2]Available at `https://scikit-learn.org/stable/`
[3]Available at `https://github.com/iskandr/fancyimpute`

public databases of the medical context, available on the UCI[4] and Kaggle[5] repositories, that cover a variety of pathologies: ctg_2c, bc_coimbra, breast_tissue_2c, cleveland_0_vs_4, dermatology_6, ecoli, parkinson, pima, relax, spectf, thyroid_3_vs_2, transfusion, vertebral_2c, wisconsin and wpbc. These databases have a number of observations between 106 and 2126, and a number of features between 4 and 44.

7.2 Experimental Results

Starting the analysis by the imputation quality, the RMSE results obtained from the experiment are presented in Table 7.1. These results show that the imputation error increases a small amount when the number of combined datasets also increases, with the best results being obtained when 2 datasets are combined. However, it is import to consider that for each combined dataset, the amount of missing values increases according to the number of new unique features. This means that the error increase is not proportional to the growth in the missing values. Therefore, to proper analyze this error, the ratio of RMSE per missing value must be considered. Table 7.2 presents this ratio for the best RMSE results of each database, along with the average number of missing values that is presented in Table 7.3 for each combination of datasets. These results show that the lowest RMSE ratio is obtained when 5 datasets are combined. Moreover, this ratio decreases when the number of combined datasets increases, which is a clear indication of the improvements in the imputation achieved by using external information. Therefore, it is possible to conclude that the use of more information of the same medical context improves the imputation of missing values under MNAR.

Regarding the classification performance, the F1 score results for the Decision Tree classifier are presented in Table 7.4. For all the 15 databases, the best F1 score is obtained almost always when the number of datasets combined is 5 (the maximum). Moreover, the results also show that the classification performance increases gradually when the number of combinations of datasets also increases. To ensure that these results were statistically significant, the Friedman test was applied with a significance level of 5%. The treatment variable used was the number of datasets and the block variable was the imputation algorithm. This means that, although the p-values exist only for the number of datasets, the effect of the imputation algorithm is taken into account and is not being neglected. The inference was performed individually for each value of the feature similarity rate, and the results are presented in Table 7.5. The p-values show that the number of datasets combined have an impact on the F1 scores in 13 of the 15 databases.

To understand when the differences between the F1 score results obtained using a single dataset and a combination of 2, 3, 4, and 5 datasets were statistically significant, the

[4]Available at `https://archive.ics.uci.edu/ml/datasets.html`
[5]Available at `https://www.kaggle.com/`

Table 7.1: RMSE results for each combination of the number of datasets and imputation algorithms, applied to all the 15 databases and averaged over the 4 values of the features similarity rate. For each cell, the top value is the RMSE average and the bottom value the standard deviation. The yellow cells are the best results for each combination of variables and the bolded results are the best for the 15 databases.

Database	Num. Datasets = 2					Num. Datasets = 3					Num. Datasets = 4					Num. Datasets = 5				
	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5
ctg_2c	14.32 (0.54)	**8.55** (1.05)	8.77 (0.83)	16.44 (4.07)	14.53 (0.52)	14.94 (0.27)	9.30 (1.13)	9.09 (0.73)	23.78 (6.02)	15.17 (0.28)	15.17 (0.26)	9.38 (1.05)	9.22 (0.75)	29.05 (6.48)	15.41 (0.29)	15.32 (0.16)	9.55 (1.14)	9.37 (0.79)	32.96 (6.14)	15.55 (0.19)
bc_coimbra	**61.07** (10.33)	64.06 (10.01)	78.94 (13.49)	103.29 (20.87)	62.55 (10.74)	75.44 (5.90)	80.48 (5.64)	97.68 (7.76)	129.73 (14.40)	77.46 (6.36)	84.00 (9.05)	89.39 (9.86)	108.82 (12.35)	145.12 (19.37)	86.15 (9.34)	88.09 (12.80)	94.07 (14.03)	113.16 (16.88)	152.28 (24.66)	90.31 (13.00)
breast_tissue_2c	2717.14 (525.28)	**2032.60** (530.80)	2304.24 (550.39)	2706.11 (562.62)	2691.05 (530.54)	3565.39 (825.98)	2723.56 (810.70)	3045.60 (800.70)	3634.17 (863.80)	3552.68 (829.10)	4194.04 (1013.68)	3290.42 (977.52)	3676.55 (974.71)	4313.53 (1053.16)	4187.02 (1018.79)	4678.72 (1172.25)	3754.08 (1162.43)	4217.82 (1178.67)	4817.34 (1213.37)	4674.10 (1163.73)
cleveland_0_vs_4	**13.49** (0.70)	14.63 (0.65)	18.12 (1.10)	40.95 (8.62)	13.57 (0.76)	15.22 (0.58)	16.39 (0.63)	20.37 (1.07)	52.03 (9.21)	15.33 (0.62)	16.17 (0.43)	17.53 (0.50)	21.60 (1.13)	58.36 (8.29)	16.29 (0.42)	16.40 (0.37)	17.82 (0.43)	21.99 (1.05)	61.40 (7.26)	16.53 (0.38)
dermatology_6	1.99 (0.17)	1.85 (0.19)	2.34 (0.24)	3.78 (0.52)	**1.83** (0.17)	2.16 (0.21)	2.04 (0.23)	2.59 (0.30)	4.36 (0.68)	2.02 (0.21)	2.29 (0.24)	2.19 (0.26)	2.78 (0.34)	4.83 (0.76)	2.17 (0.24)	2.43 (0.32)	2.36 (0.36)	3.00 (0.47)	5.32 (1.01)	2.33 (0.34)
ecoli	0.15 (0.00)	0.14 (0.01)	0.16 (0.01)	0.25 (0.06)	**0.14** (0.01)	0.16 (0.00)	0.15 (0.00)	0.17 (0.01)	0.25 (0.04)	0.14 (0.01)	0.16 (0.00)	0.15 (0.00)	0.17 (0.01)	0.26 (0.04)	0.14 (0.01)	0.16 (0.00)	0.15 (0.00)	0.17 (0.01)	0.27 (0.03)	0.14 (0.01)
parkinson	16.04 (1.03)	**15.70** (1.31)	17.67 (1.75)	28.61 (4.68)	16.36 (1.07)	18.13 (1.30)	17.95 (1.78)	20.15 (2.22)	34.61 (5.27)	18.58 (1.33)	19.85 (1.22)	19.80 (1.82)	22.23 (2.20)	39.17 (5.17)	20.32 (1.23)	20.87 (1.40)	20.71 (2.00)	23.43 (2.71)	42.16 (5.07)	21.34 (1.38)
pima	30.20 (1.60)	**29.20** (1.32)	37.24 (1.48)	43.48 (3.10)	32.38 (2.00)	33.40 (0.93)	32.48 (0.49)	41.01 (0.74)	50.04 (2.92)	35.91 (1.15)	35.88 (1.50)	34.84 (2.03)	44.10 (3.24)	53.53 (4.24)	38.78 (1.98)	36.80 (2.33)	35.75 (3.00)	45.24 (4.54)	54.98 (4.79)	39.91 (2.93)
relax	0.42 (0.00)	**0.27** (0.06)	0.42 (0.03)	0.30 (0.03)	0.29 (0.04)	0.42 (0.00)	0.27 (0.06)	0.42 (0.03)	0.32 (0.03)	0.29 (0.04)	0.42 (0.00)	0.27 (0.06)	0.42 (0.03)	0.33 (0.03)	0.29 (0.04)	0.42 (0.00)	0.27 (0.06)	0.42 (0.03)	0.33 (0.03)	0.29 (0.04)
spectf	10.00 (0.06)	8.50 (0.31)	9.15 (0.18)	**7.83** (0.12)	10.23 (0.07)	10.05 (0.08)	8.56 (0.26)	9.17 (0.18)	10.27 (0.09)	10.04 (0.04)	8.61 (0.32)	9.18 (0.18)	10.61 (3.34)	10.27 (0.05)	10.03 (0.02)	8.67 (0.35)	9.17 (0.16)	12.34 (4.95)	10.26 (0.03)	
thyroid_3_vs_2	**0.17** (0.01)	0.17 (0.01)	0.23 (0.01)	0.19 (0.01)	0.18 (0.01)	0.17 (0.01)	0.18 (0.01)	0.24 (0.01)	0.20 (0.01)	0.18 (0.01)	0.17 (0.00)	0.18 (0.00)	0.24 (0.01)	0.21 (0.01)	0.19 (0.00)	0.17 (0.00)	0.18 (0.00)	0.24 (0.01)	0.21 (0.00)	0.19 (0.00)
transfusion	417.52 (68.62)	**138.13** (89.75)	231.32 (91.23)	567.86 (95.69)	432.72 (70.67)	528.88 (57.72)	174.99 (111.76)	291.79 (111.32)	719.21 (80.64)	549.14 (60.02)	537.69 (68.86)	179.07 (118.44)	296.39 (121.34)	732.53 (96.94)	558.39 (71.47)	537.00 (71.50)	180.07 (117.64)	297.65 (124.63)	731.54 (98.85)	557.44 (74.09)
vertebral_2c	18.93 (0.72)	**15.19** (1.62)	17.11 (1.55)	32.37 (4.97)	19.44 (0.76)	19.88 (0.93)	16.76 (1.09)	18.18 (1.64)	38.29 (3.73)	20.49 (0.96)	20.26 (1.01)	17.36 (1.44)	18.57 (1.84)	40.15 (4.27)	20.90 (1.07)	20.31 (0.94)	17.51 (1.40)	18.61 (1.83)	41.17 (4.15)	20.94 (1.00)
wisconsin	2.81 (0.02)	**1.89** (0.06)	2.39 (0.09)	2.84 (0.36)	2.02 (0.02)	2.83 (0.01)	1.92 (0.04)	2.41 (0.07)	2.76 (0.26)	2.03 (0.02)	2.83 (0.01)	1.92 (0.03)	2.42 (0.06)	2.73 (0.24)	2.03 (0.02)	2.83 (0.00)	1.92 (0.04)	2.43 (0.07)	2.76 (0.22)	2.03 (0.02)
wpbc	69.56 (8.83)	**27.35** (10.60)	40.04 (6.04)	115.79 (28.47)	70.55 (9.19)	82.96 (12.27)	32.72 (13.20)	47.51 (7.61)	153.84 (37.00)	84.25 (12.78)	93.70 (13.80)	37.21 (15.37)	53.13 (8.54)	183.58 (40.69)	95.00 (14.28)	99.93 (16.70)	41.28 (18.54)	56.66 (10.46)	203.12 (47.01)	101.39 (16.95)

Algorithms Legend - AL1: Mean, AL2: MICE, AL3: KNN, AL4: SI, AL5: SVR

Table 7.2: Best RMSE result for each combination of the number of datasets divided by the average number of missing values and applied to all the 15 databases. Each cell presents the ratio of the RMSE by missing value. The bolded results are the best for the 15 databases.

Database	Num. Datasets			
	2	**3**	**4**	**5**
ctg_2c	0.00452	0.00162	0.00086	**0.00055**
bc_coimbra	1.25788	0.52737	0.30747	**0.21074**
breast_tissue_2c	45.83089	20.84623	13.18806	**9.82743**
cleveland_0_vs_4	0.13310	0.05112	0.02895	**0.01893**
dermatology_6	0.00377	0.00143	0.00078	**0.00052**
ecoli	0.00119	0.00038	0.00020	**0.00014**
parkinson	0.08946	0.03461	0.01956	**0.01295**
pima	0.10580	0.03786	0.02077	**0.01454**
relax	0.00287	0.00097	0.00050	**0.00032**
spectf	0.01721	0.00627	0.00324	**0.00200**
thyroid_3_vs_2	0.00027	0.00009	0.00005	**0.00003**
transfusion	0.63757	0.27261	0.17781	**0.13361**
vertebral_2c	0.15313	0.05406	0.03164	**0.02353**
wisconsin	0.00660	0.00227	0.00119	**0.00078**
wpbc	0.11311	0.04561	0.02640	**0.01766**

Table 7.3: Average number of missing values for each combination of the number of datasets and applied to all the 15 databases.

Database	Num. Datasets			
	2	**3**	**4**	**5**
ctg_2c	1892	5611	10775	17009
bc_coimbra	49	143	273	418
breast_tissue_2c	44	131	250	382
cleveland_0_vs_4	101	298	559	867
dermatology_6	485	1417	2764	4512
ecoli	117	369	694	1009
parkinson	176	519	1012	1599
pima	276	858	1677	2459
relax	94	278	536	838
spectf	455	1366	2661	4327
thyroid_3_vs_2	625	1851	3553	5626
transfusion	217	642	1007	1348
vertebral_2c	99	310	549	744
wisconsin	286	845	1614	2460
wpbc	242	717	1410	2337

post hoc Nemenyi test was applied. Table 7.6 presents the results and they show that in general the statistical significance only exists when the number of datasets combined is 4 or 5, which reinforces the results previously described[6]. Therefore, it is possible to

[6]Although the Nemenyi test p-values are two-tailed, it is possible to ensure that these results always reflect improvement in the F1 scores by cross-analyzing them with the ones from Table 7.4.

conclude that the use of more information of the same medical context for the imputation of missing values under MNAR has a positive impact on the performance of the Decision Tree classifier. The use of more information could also be in part responsible for the improvements, but the main conclusion here is that the imputation was well performed, otherwise it would have a negative impact in the classification results.

Finally, an analysis of the results obtained from the imputation algorithms was also conducted, since they have an impact on the study (the Friedman test was also used here). Table 7.7 shows how many times each algorithm was the best. The results show different but consistent values for the classification and imputation evaluation. For the imputation the MICE algorithm clearly outperformed the remaining methods, but for the classification the KNN presented the best results. A possible explanation for this difference is that MICE creates several regressions based on all complete data, and therefore is able to generate values that better represent the overall population and that are more close to real ones. On the other hand, the KNN obtains the observation more similar to the one that is being imputed, which means that the generated value is a copy of the other value, and this benefits the classification because the dispersion of the observations reduces and the classifier is able to define the "boundaries" between the labels more easily.

7.3 Conclusions

Missing data is a problem often found in real-world datasets and, although several works have been published in this field, the MNAR mechanism is still the one less addressed and with worst results. Considering the nature of MNAR, this work tries to prove if using multiple sources of information from the same data context helps improving the imputation of MNAR missing values, using only medical databases. The results show that when the number of datasets increases the RMSE values increase just a small amount, which is a considerable improvement when the ratio of RMSE by missing value is considered. Moreover, the results also show that the precision and recall of the Decision Tree classifier increase and are not negatively biased by the imputation when more information is added, particularly when at least 4 datasets are combined, and the results are statistically significant with a significance level of 5%. Moreover, the use of the MICE algorithm for the imputation benefits its error, while the KNN algorithm is the best for the classification performance. In the future the study should be extended to include more databases from different data contexts and more classifiers of different natures, in order to allow a generalization of the results for these scenarios.

Table 7.4: F1 score results for each combination of the number of datasets and imputation algorithms, applied to all the 15 databases and averaged over the 4 values of the features similarity rate. For each cell, the top value is the F1 score average and the bottom value the standard deviation. The yellow cells are the best results for each combination of variables and the bolded results are the best for the 15 databases.

Database	Num. Datasets = 1					Num. Datasets = 2					Num. Datasets = 3					Num. Datasets = 4					Num. Datasets = 5				
	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5	AL1	AL2	AL3	AL4	AL5
ctg_2c	0.66	0.66	0.66	0.66	0.65	0.68	0.67	0.68	0.67	0.67	0.69	0.67	0.70	0.68	0.69	0.71	0.68	0.71	0.69	0.70	0.72	0.69	**0.72**	0.70	0.71
	(0.02)	(0.03)	(0.02)	(0.02)	(0.02)	(0.03)	(0.04)	(0.03)	(0.03)	(0.03)	(0.03)	(0.04)	(0.03)	(0.03)	(0.03)	(0.03)	(0.04)	(0.03)	(0.03)	(0.03)	(0.03)	(0.04)	(0.03)	(0.04)	(0.03)
bc_coimbra	0.97	0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
breast_tissue_2c	0.94	0.94	0.94	0.94	0.95	0.98	0.97	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	0.99	0.99	**0.99**	0.99
	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
cleveland_0_vs_4	0.96	0.96	0.96	0.96	0.96	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	0.99	0.99
	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
dermatology_6	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
ecoli	0.94	0.94	0.94	0.94	0.94	0.96	0.96	0.97	0.96	0.96	0.98	0.97	0.98	0.97	0.97	0.98	0.98	0.98	0.98	0.98	**0.98**	0.98	0.98	0.98	0.98
	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
parkinson	0.81	0.81	0.81	0.82	0.82	0.86	0.85	0.86	0.86	0.85	0.88	0.88	0.89	0.88	0.88	0.90	0.89	0.91	0.90	0.90	0.91	0.90	**0.92**	0.91	0.91
	(0.00)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)
pima	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	**0.66**	0.65	0.64	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
relax	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.58	0.57	0.57	0.57	0.57	**0.58**	0.57	0.57
	(0.01)	(0.00)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
spectf	0.8	0.81	0.81	0.81	0.81	0.94	0.93	0.94	0.93	0.94	0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.98	**0.99**
	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
thyroid_3_vs_2	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.95	0.94	0.95	0.94	0.94	**0.95**	0.95	0.94	0.95	0.95
	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
transfusion	0.69	0.70	0.69	0.69	0.69	0.69	0.70	0.69	0.69	0.70	0.70	0.69	0.69	0.70	0.70	0.70	0.69	0.70	0.70	0.70	**0.70**	0.69	0.70	0.70	0.70
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
vertebral_2c	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	1.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
wisconsin	0.93	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.94	0.93	0.93	0.94	0.94	0.93	0.94	0.93	0.94	0.94	0.93	**0.94**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)
wpbc	0.65	0.66	0.65	0.65	0.65	0.66	0.66	0.66	0.66	0.66	0.67	0.66	0.67	0.66	0.66	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	**0.67**	0.67
	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)

Algorithms Legend - AL1: Mean, AL2: MICE, AL3: KNN, AL4: SI, AL5: SVR

Table 7.5: *P*-values of the Friedman test for the F1 score, using the number of datasets as the treatment variable and the imputation algorithm as the block variable, and obtained individually for each feature similarity rate. The bolded *p*-values are the statistical significant ones, assuming a significance level of 5%.

Database	**FSRate**			
	40%	**50%**	**60%**	**70%**
ctg_2c	**0.001**	**0.000**	**0.000**	**0.000**
bc_coimbra	**0.001**	**0.001**	**0.001**	**0.001**
breast_tissue_2c	**0.001**	**0.001**	**0.001**	**0.000**
cleveland_0_vs_4	**0.000**	**0.001**	**0.000**	**0.001**
dermatology_6	**0.001**	**0.000**	**0.001**	**0.000**
ecoli	**0.000**	**0.000**	**0.000**	**0.000**
parkinson	**0.000**	**0.000**	**0.000**	**0.000**
pima	0.193	**0.023**	0.291	0.161
relax	0.364	0.218	0.142	0.051
spectf	**0.000**	**0.000**	**0.000**	**0.000**
thyroid_3_vs_2	0.205	**0.015**	**0.001**	**0.001**
transfusion	**0.039**	**0.003**	**0.015**	**0.013**
vertebral_2c	**0.000**	**0.002**	**0.001**	**0.002**
wisconsin	0.092	**0.001**	**0.001**	**0.000**
wpbc	0.218	**0.010**	**0.001**	**0.001**

Table 7.6: Number of times where the difference between the F1 score for Num. Datasets = 1 *vs* Num. Datasets = 2, 3, 4, 5 was statistical significant, according to the *p*-values of the post hoc Nemenyi test and assuming a significance level of 5%. The frequencies are presented individually for each value of the features similarity rate.

	FSRate			
ND	**40%**	**50%**	**60%**	**70%**
2	0	0	0	0
3	0	1	0	0
4	**9**	**10**	**10**	**12**
5	**9**	**13**	**13**	**13**

Table 7.7: Number of times that each imputation algorithm presented the best results for each number of datasets. The frequencies are presented individually for the F1 score (left) and the RMSE (right).

	F1 Score					**RMSE**				
ND	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
1	5	**18**	15	11	11	–	–	–	–	–
2	6	5	**25**	12	12	12	**34**	1	4	9
3	13	6	**20**	7	14	13	**34**	3	2	8
4	14	7	**20**	6	13	14	**34**	3	0	9
5	**21**	5	14	6	14	14	**34**	3	0	9
Sum	59	41	**94**	42	64	53	**136**	10	6	35

Algorithms Legend - AL1: Mean, AL2: MICE, AL3: KNN, AL4: SI, AL5: SVR

This page is intentionally left blank.

Chapter 8

Automatic Delta-Adjustment Method

In general terms, the imputation approaches tend to only be suitable for the MCAR and MAR mechanisms. Since the models base their estimations on the available data, the imputation tends to be biased when performed under MNAR assumptions because the existent data is not enough to properly model its missingness causes [50]. Nevertheless, there are a few approaches to reduce this bias, one being the delta-adjustment method [88]. Assuming that the imputation is performed with a model designed for the MAR mechanism, the missing values are likely to be either underestimated or overestimated. In other words, they are probably shifted towards a lower or higher domain. The delta-adjustment method provides a simple and transparent solution to correct this shift: add or multiply the imputed values by a correction factor. However, this factor must be manually defined by domain experts, which makes the delta-adjustment method often unfeasible to be applied considering the complexity and cost of consulting these experts. Moreover, the lack of scientific rigor in this process may also compromise the generalization of the results.

In this work we introduce an automatic procedure to estimate the approximate delta adjustment values for every feature of the dataset, which we call Automatic Delta-Adjustment Method (ADAM). The procedure explores the distance between the biased imputation and other estimations that are sampled from Gaussian distributions with extreme means that stretch the range of considered values. We conducted an experimental setup with seven state-of-the-art imputation methods, comprising 10 datasets of the healthcare domain that were injected with missing values under MNAR. We compared the results of the imputation with and without the adjustments provided by ADAM trough the Mean Absolute Error (MAE), and validated the results trough the Wilcoxon signed-rank test with a significance level of 5%. The use of ADAM's adjustment provided significant improvements for all imputation methods and missing rates.

The remainder of the chapter is organized in the following way: Section 8.1 describes in detail the proposed ADAM method; Section 8.2 presents the design of the experimental setup and displays the analysis of the obtained results; and Section 8.3 states the final conclusions and future directions.

8.1 Proposed Approach

In this work an automatic procedure that is capable of estimating approximate delta adjustment values is proposed. Considering the MNAR characteristics, it is impossible to find an optimal delta value since it would depend on the missing values themselves. Therefore, we rely on statistics to find an approximate value that will bring the estimates made under MAR assumptions valid under MNAR. We called our approach Automatic Delta-Adjustment Method (ADAM).

ADAM's goal is to find a factor comprised within $[0, 1]$ for each feature of the dataset. This factor will be multiplied by the imputed values in order to adjust them. The proposed procedure to estimate the factor for a specific feature X_i is presented in Algorithm 3. The procedure assumes that the missing values correspond to the smaller values of the feature (the opposite scenario is addressed later), and comprises these main steps:

1. The mean of the available values within the first quartile is calculated (μ_{Q1}). This value is representative of the lower tail of the feature, and it is used instead of the minimum because it better represents the group of smaller values and it is resilient to extreme factors. Additionally, the standard deviation of all the available values in the feature is also calculated (σ_{all});

2. The missing values are imputed three independent times by sampling from a Gaussian distribution where the standard deviation is one and the mean varies μ_{Q1} minus σ_{all} according to the empirical rule: $\mu_{Q1} - \sigma_{all}$, $\mu_{Q1} - 2\sigma_{all}$, and $\mu_{Q1} - 3\sigma_{all}$. Such variation is used to define a reasonable range for the missing values, since the imputed values with MAR models are likely to be shifted towards a higher domain. Moreover, when later calculating the adjustment factor, these three imputations are weighted so that the more extreme values (i.e., further away from μ_{Q1}) have a decreased impact on the calculation.

3. The Principal Component Analysis (PCA)[1] method is used to condense the data from the features into a single representative numeric value. This transformation is applied individually to the dataset imputed with the MAR model and the datasets imputed in the previous step, which leads to four different scalars by feature. To achieve these results the datasets must be transposed before the transformation since

[1]PCA is a feature extraction technique often used for dimensionality reduction. It computes the principal components of the data and returns the first n, which is a user-defined parameter [1].

we are condensing the data from the features and not the features themselves. The obtained values for each feature are then used to estimate how far away are the imputations from the previous step when compared to the original imputation with the MAR model. For this purpose, we calculate the euclidean distance between the value representing the imputation with the MAR model and each one of the remaining scalars;

4. To calculate the factor, the distances from the previous step are normalized so that their sum is equal to one (which is necessary to achieve a factor within $[0, 1]$), and a weighted mean is calculated so that the imputations performed according to the empirical rule have an impact in the factor proportional to their means.

Algorithm 3 Pseudocode for the ADAM procedure. The missing values are on the lower tail of the feature X_i. The algorithm receives as input the feature data containing missing values ($X_i\_missing$) and already imputed ($X_i\_imputed$), and it returns the adjusted data for the feature ($X_i\_adjusted$).

Input: $X_i\_missing$, $X_i\_imputed$
Output: $X_i\_adjusted$
1: μ_{Q1} = mean($Q1$ values from $X_i\_missing$)
2: σ_{all} = standard_deviation($X_i\_missing$)
3: **for** each j in $\{1, 2, 3\}$ **do**
4: $\quad gaussian\_imputed_j$ = missing data $\sim \mathcal{N}(\mu_{Q1} - j * \sigma_{all}, 1)$
5: **end for**
6: $scalar_0$ = PCA (1 comp.) applied to $X_i\_imputed$
7: **for** each j in $\{1, 2, 3\}$ **do**
8: $\quad scalar_j$ = PCA (1 comp.) applied to $gaussian\_imputed_j$
9: $\quad dist_j$ = euclidean_distance($scalar_0$, $scalar_j$)
10: **end for**
11: Normalize $dist_j, \forall j \in \{1, 2, 3\}$ so that $\sum_1^3 dist_j = 1$
12: $factor_i = (\, 3dist_3 + 2dist_2 + dist_1 \,) \,/\, 6$
13: $X_i\_adjusted = X_i\_imputed - factor_i * X_i\_imputed$
14: **return** $X_i\_adjusted$

After performing the previous step, we have an independent factor ($factor_i$) within $[0, 1]$ for each feature (X_i) of the dataset. To adjust the values imputed with the MAR model, the following equation is applied to each of the D features:

$$X_i = X_i - factor_i * X_i, \forall i \in \{1, ..., D\} \tag{8.1}$$

As previously stated, the described procedure assumes that the missing values are the smaller values of the feature. However, the opposite scenario where the missing data corresponds to the larger values is also valid. To address this scenario, the procedure suffers two specific changes:

- In step 1), the last quartile is instead calculated (μ_{Q4}) since it now represents the

higher tail of the feature.

- In step 2), the mean is varied on the opposite direction ($\mu_{Q4} + \sigma_{all}$, $\mu_{Q4} + 2\sigma_{all}$, and $\mu_{Q4} + 3\sigma_{all}$), since the imputed values with MAR models are now shifted towards a lower domain.

Finally, the adjustment operation also changes since the imputed values are now being shifted upwards:

$$X_i = X_i + factor_i * X_i, \forall i \in \{1, ..., D\} \tag{8.2}$$

The identification of the feature's missing tail (i.e., if the missing values correspond to the smaller or larger values of the feature) should be made manually through exploratory data analysis or even by using domain knowledge.

8.2 Experimental Results

To evaluate if ADAM is effective in adjusting the imputed values, an experiment was conducted to compare the imputation results before and after applying it. The experimental setup comprised the following seven state-of-the-art imputation methods:

- Mean imputation, where the mean of the available values of each feature are used to impute;
- Multiple Imputation by Chained Equations (MICE), which is a multiple imputation-based approach where several Bayesian ridge regressions are fitted in a round-robin procedure with 100 iterations. Each regression defines as the dependent variable one of the features containing missing values, and uses the remaining as the independent variables [15];
- k-Nearest Neighbors (kNN) imputation with $k = 5$, which selects the k nearest neighbors of the instance being imputed by calculating the Euclidean distance between the available values, and uses the mean of these k neighbors to impute the features with missing data [95];
- SoftImpute, which is a matrix completion iterative method based on nuclear-norm regularization, that estimates the missing values through soft-threshold singular value decomposition [64]. A maximum of 100 iterations was considered;
- Denoising Autoencoder (DAE), which is an autoencoder trained with data containing additional noise, which in this context is the existence of missing values. Incidentally, an autoencoder is a special type of neural network that tries to reproduce the input data at the output layer, usually by learning a compressed representation of

that data [79, 118]. The architecture of the networks was defined with the following hyperparameters: a single hidden layer with a number units equal to half of the input dimension; ReLU as the activation function; batches of 64 instances; a maximum of 200 epochs; Adam as the optimization algorithm; Mean Squared Error as the loss function; a learning rate of 0.001; a dropout layer with a rate of 25% for regularization; early stop if the validation loss has no improvements over 100 epochs; a reduction of the learning rate by 80% if there are no improvements over 100 epochs; and Sigmoid as the activation function for the output layer, so that the data is normalized within $[0, 1]$;

- Variational Autoencoder (VAE), which is a generative variant of the autoenconder that learns the multidimensional parameters of a Gaussian distribution (i.e., mean and standard deviation), and by sampling from these is able to generate new data with similar characteristics [65, 79, 80]. The architecture of the networks was defined with the same hyperparameters as the DAE, adding the two layers needed to represent the Gaussian parameters.

- Generative Adversarial Imputation Nets (GAIN), which is a direct application of the well-known generative adversarial networks to the problem of missing data [126]. The generator network performs the imputation, while the discriminator tries to distinguish between original and imputed data. The networks were parametrized with the hyperparameters reported by the authors of the method.

The architecture and hyperparameters used for the seven imputation methods were defined through a grid search procedure. This is a common strategy that aims to obtain optimal hyperparameters that conform to common use cases. Regarding the implementation of these algorithms, the autoencoders were implemented with the Keras library, the SoftImpute was coded from scratch, the GAIN code was obtained from the original authors[2], and the remaining methods were directly used from the Scikit-learn library. Furthermore, the implementation of ADAM is available on GitHub[3].

The experiment considered 10 public datasets from the healthcare domain, covering clinical research based on routinely collected data for different pathologies. This domain was chosen since it frequently suffers from missing data under the MNAR mechanism [74]. All datasets are available at the UC Irvine repository[4], and they cover different ranges of instances and features, as Table 8.1 shows.

In order to have a controlled experiment, the datasets were all complete (i.e., without missing data) and the missing values were artificially generated according to the MNAR mechanism. The used strategy removed the smaller values [112] or larger values [124]

[2] `https://github.com/jsyoon0823/GAIN`
[3] `https://github.com/ricardodcpereira/ADAM`
[4] `https://archive.ics.uci.edu/ml`

Table 8.1: Datasets used in the experimental setup.

Dataset	# Instances	# Features	
		Continuous	Categorical
wisconsin	569	31	0
ctg	2126	21	2
pima	768	9	0
liver	583	10	1
diabetic-retinopathy	1151	16	4
parkinsons	195	23	0
bc-coimbra	116	10	0
thoracic-surgery	470	14	3
spine	310	13	0
mammographic-masses	830	2	4

of the orderable features upon a certain missing rate (which excludes nominal categorical features since they are non-orderable). Such strategy was applied in a multivariate fashion where the missing rate is defined for the entire dataset and several features are injected with missing values simultaneously [97]. Consequently, each feature has a different number of missing values, which grouped together sum up to the desired global missing rate. The imputation is performed for all features at once, and the results are assessed through the Mean Absolute Error (MAE) calculated between the ground truth (i.e., original data) and the imputed values.

All datasets were normalized within $[0, 1]$ and split into train and test sets with 70% and 30% of the instances, respectively. The normalizer uses the minimum and maximum values from the train set and it is then applied to both sets. This strategy keeps the test data isolated from the training data, preventing bias in the test set. However, when dealing with high missing rates (usually above 50%), it is possible for the test set to contain unseen values, which makes its normalization boundaries go slightly beyond the aimed $[0, 1]$ domain. For the neural network-based methods, 20% of the train set was used for validation. The non-orderable features (e.g., categorical nominal) were transformed through one-hot encoding (i.e., dummy coding). The missing values were injected independently in each of the described sets in order to ensure that all of them had equal missing rates and MNAR assumptions. Five different missing rates were considered (5%, 10%, 20%, 40%, and 60%) in order to cover different levels of missingness. For the neural network-based methods the missing values were pre-imputed with the mean imputation.

To mitigate bias and stochastic behaviors the experiment was executed 30 independent times, with the data being randomly split into the train and test sets in each run. The results here presented are the mean of these 30 runs. Each run was executed in a computer with the following specifications: Windows 11, CPU AMD Ryzen 5600X, 16GB RAM, and

GPU NVIDIA GeForce GTX 1060 6GB. The time complexity of applying ADAM for each of the imputation methods was not directly measured, but the impact was not significant.

The obtained results for the adjustment of imputed values provided by ADAM are presented in Tables 8.2 and 8.3. For Table 8.2 the left tail was missing (i.e., smaller values of the features), and for Table 8.3 the right tail was removed (i.e., larger values of the features).

In an overall analysis, all methods achieved smaller imputation errors after the imputed values were adjusted through ADAM. This behavior is consistent for all missing rates and for both MNAR strategies with the smaller and larger values being removed, with the global error improvement being 11%. The enhancement also appears to be stable among the different levels of missingness, peaking at 18% for the 5% missing rate, which shows ADAM's resilience to variations on this factor.

To understand if the obtained results were statistically significant we applied the Wilcoxon signed-rank test with a significance level of 5%. This test was chosen because the normality assumptions were not met, and we have paired MAE values for each imputation method (before and after applying ADAM). The test was applied independently for each missing rate, and the one-sided alternative was used since we were only interested in evaluating if the MAE values after applying ADAM are significantly lower. The obtained p-values showed that the results are statistically significant with $p < 0.001$ for all settings, which corroborates the good performance obtained by ADAM.

8.3 Conclusions

In this work we proposed a procedure called Automatic Delta-Adjustment Method (ADAM) to automatically estimate the delta-adjustment values. We compared the results obtained by seven state-of-the-art imputation methods with and without the adjustments provided by ADAM. The experimental setup comprised 10 datasets from the healthcare context that were injected with missing values under MNAR in a multivariate fashion. We concluded that the adjustment performed by ADAM led to error improvements in all imputations methods and missing rates, achieving a global enhancement of 11%.

Motivated by the results achieved with ADAM, future work will be focused on integrating it with an auxiliary procedure to automatically identify the features' missing tails. A possible direction is to model this task as a binary classification problem and use machine learning to solve it. Furthermore, we want to incorporate information from other datasets in the adjustments calculations, so that external data can be used to help reduce bias in MNAR settings. Finally, we also want to compare ADAM results to imputed data that was manually adjusted by domain experts.

Table 8.2: MAE results of the imputation methods with and without the adjustment provided by ADAM. The left tail of the features was missing (i.e., smaller values).

Imp.	ADAM	Missing Rate				
		5%	10%	20%	40%	60%
AE	No	0.236 ± 0.09	0.238 ± 0.09	0.256 ± 0.10	0.322 ± 0.15	0.515 ± 0.34
	Yes	**0.192 ± 0.07**	**0.195 ± 0.06**	**0.212 ± 0.07**	**0.274 ± 0.12**	**0.464 ± 0.31**
GAIN	No	0.254 ± 0.08	0.252 ± 0.08	0.266 ± 0.10	0.324 ± 0.15	0.519 ± 0.34
	Yes	**0.206 ± 0.06**	**0.208 ± 0.06**	**0.220 ± 0.07**	**0.277 ± 0.12**	**0.469 ± 0.31**
MICE	No	0.184 ± 0.09	0.186 ± 0.09	0.204 ± 0.10	0.280 ± 0.14	0.475 ± 0.34
	Yes	**0.154 ± 0.07**	**0.157 ± 0.07**	**0.174 ± 0.07**	**0.245 ± 0.12**	**0.436 ± 0.31**
Mean	No	0.269 ± 0.08	0.264 ± 0.08	0.278 ± 0.09	0.334 ± 0.14	0.523 ± 0.33
	Yes	**0.217 ± 0.05**	**0.216 ± 0.06**	**0.229 ± 0.07**	**0.284 ± 0.11**	**0.472 ± 0.30**
SoftImp	No	0.178 ± 0.07	0.184 ± 0.07	0.197 ± 0.08	0.255 ± 0.11	0.429 ± 0.29
	Yes	**0.152 ± 0.06**	**0.160 ± 0.06**	**0.174 ± 0.06**	**0.233 ± 0.10**	**0.412 ± 0.28**
VAE	No	0.285 ± 0.11	0.279 ± 0.10	0.294 ± 0.12	0.348 ± 0.16	0.533 ± 0.35
	Yes	**0.227 ± 0.07**	**0.224 ± 0.07**	**0.238 ± 0.08**	**0.292 ± 0.12**	**0.477 ± 0.32**
kNN	No	0.185 ± 0.09	0.194 ± 0.09	0.219 ± 0.10	0.292 ± 0.14	0.487 ± 0.34
	Yes	**0.153 ± 0.07**	**0.162 ± 0.07**	**0.184 ± 0.07**	**0.254 ± 0.11**	**0.446 ± 0.31**

Table 8.3: MAE results of the imputation methods with and without the adjustment provided by ADAM. The right tail of the features was missing (i.e., larger values).

Imp.	ADAM	Missing Rate				
		5%	10%	20%	40%	60%
AE	No	0.710 ± 0.27	0.847 ± 0.52	1.417 ± 1.91	2.770 ± 3.37	4.011 ± 4.81
	Yes	**0.610 ± 0.26**	**0.744 ± 0.51**	**1.306 ± 1.91**	**2.655 ± 3.36**	**3.894 ± 4.80**
GAIN	No	0.725 ± 0.27	0.849 ± 0.52	1.410 ± 1.91	2.769 ± 3.37	4.036 ± 4.81
	Yes	**0.628 ± 0.26**	**0.747 ± 0.51**	**1.301 ± 1.91**	**2.654 ± 3.36**	**3.924 ± 4.80**
MICE	No	0.579 ± 0.23	0.716 ± 0.47	1.291 ± 1.87	2.678 ± 3.35	3.943 ± 4.80
	Yes	**0.496 ± 0.22**	**0.624 ± 0.46**	**1.183 ± 1.85**	**2.557 ± 3.34**	**3.813 ± 4.79**
Mean	No	0.774 ± 0.28	0.896 ± 0.53	1.448 ± 1.92	2.784 ± 3.36	4.019 ± 4.81
	Yes	**0.689 ± 0.27**	**0.804 ± 0.52**	**1.346 ± 1.91**	**2.671 ± 3.35**	**3.902 ± 4.80**
SoftImp	No	0.676 ± 0.27	0.826 ± 0.51	1.422 ± 1.90	2.834 ± 3.39	4.187 ± 4.84
	Yes	**0.579 ± 0.26**	**0.727 ± 0.50**	**1.319 ± 1.90**	**2.738 ± 3.38**	**4.119 ± 4.84**
VAE	No	0.758 ± 0.29	0.884 ± 0.54	1.441 ± 1.92	2.781 ± 3.37	4.018 ± 4.81
	Yes	**0.671 ± 0.28**	**0.790 ± 0.53**	**1.338 ± 1.92**	**2.669 ± 3.36**	**3.903 ± 4.80**
kNN	No	0.618 ± 0.24	0.761 ± 0.49	1.342 ± 1.89	2.722 ± 3.36	3.976 ± 4.81
	Yes	**0.509 ± 0.23**	**0.650 ± 0.48**	**1.224 ± 1.88**	**2.600 ± 3.35**	**3.852 ± 4.80**

Chapter 9

Artificial Generation of Missing Data

To achieve the best imputation results possible, it is vital to select the imputation method more suitable for the data characteristics and domain. To make such decision an experimental study is often required where several methods are used to generate the estimated values that will replace the missing data. The selection is then performed through the imputation error, which is calculated between the estimates and the ground truth (i.e., the original values) for each method [77]. However, to calculate this error the dataset must be complete (i.e., it can not have missing values) since the ground truth values are required. Therefore, the complete dataset must be injected with missing data under a specific mechanism through a process of artificial generation of missing values. Such process must accurately mimic the chosen mechanism, otherwise the obtained results will be biased. This is easy to achieve with the MCAR and MAR mechanisms, but for MNAR the existing approaches to generate the missing values are rather limited - they are mostly applicable to continuous features, mimicking scenarios where the missingness of values only depends on themselves [97]. Such limitations become a major issue when considering that MNAR is the mechanism more often found in several critical contexts, such as healthcare [74]. In this work, we tackle and overcome these limitations by introducing four novel artificial generation strategies for MNAR, which mimic real-world scenarios that follow the assumptions behind this type of mechanism. Overall, the proposed strategies can be used with continuous and categorical features (including nominal ones), and are able to generate missing values related to themselves and to unavailable/unobserved data. This is the first time that artificial generation strategies for MNAR with such realistic characteristics have been proposed.

Furthermore, a benchmark study is also conducted to compare the imputation error obtained by a comprehensive baseline of six state-of-the-art imputation methods. The main goal is to evaluate how these methods perform when dealing with the different MNAR

characteristics introduced by the proposed generation strategies. The conclusions obtained from this benchmark study can also help to understand which imputation methods may be more suitable for the different MNAR settings, leading to a set of guidelines that can be used by other researches and practitioners.

The experimental setup uses 10 public datasets from the medical context, which was chosen because it suffers frequently from missing data under MNAR assumptions, making it an ideal test bed for a study such as this one [74]. Furthermore, since all datasets are public, the study can be easily replicated. The setup considers five levels of missingness (10%, 20%, 40%, 60%, and 80%) and the results are evaluated individually for continuous, categorical, and mixed data types, using the mean absolute error (MAE) metric. A detailed discussion is presented about the conclusions obtained with the experiment, containing a set of guidelines as the take-home message.

The remainder of the chapter is organized in the following way: Section 9.1 introduces the new artificial generation strategies to generate MNAR values; Section 9.2 describes the experimental setup used in the benchmark study; Section 9.3 presents the obtained experimental results; finally, Section 9.4 discusses the results and conclusions extracted from them.

9.1 Proposed Strategies

Regarding the artificial generation of MNAR values, the existing approaches are rather limited [97]. The most common approach is to remove the lowest values of a selected feature up to a predefined missing rate [112]. Alternatively, the highest values of the feature can be removed instead [124]. Since this approach is univariate (i.e., the missing values are generated in a single feature), the feature most correlated with the class labels is often chosen [112]. This strategy poses a major limitation: since it requires ordering the feature to find its lowest or highest values, only continuous and ordinal data types are supported, not nominal features. For multivariate scenarios, Garciarena et al. [32] proposed an extension of the described univariate approach to work on several features, called Missingness depending on its Value Itself (MIV). The strategy is essentially the same: the lowest values are removed, but now for several previously selected features. The missingness rate is global for all features, being therefore divided by the selected ones. The same authors also proposed another mechanism called Missing depending on unobserved Variables (MuOv), where the generated missing values are related to a feature not included in the dataset. The approach randomly selects N patterns to be missing up to a certain rate (since the causative features are unknown), and the values from these patterns are removed. Twala et al. [112] used the same approach, but the features are aggregated in pairs or triplets and only one feature from each set is injected with missing values. Ali et al. [3] proposed a different multivariate strategy where each feature is split

into two groups: one with all values no larger than the feature's median, and another with the remaining values. One of the groups is then randomly chosen to have values removed in an amount equal to twice the missing rate, therefore respecting the quotas needed to achieve the global missing rate defined for the dataset. Zhu et al. [130] and Pan et al. [73] followed the same strategy for continuous and ordinal features, but extended it to nominal data by dividing the categories into two equally-sized groups. Since these approaches work only with half of the observations (considering the split in two groups), the global missing rate is limited to 50%.

In conclusion, the existing artificial generation strategies for MNAR are rather limited and do not cover realistic settings for this mechanism. Therefore, four new strategies are introduced in this work: three are entirely novel (Sections 9.1.2, 9.1.3 and 9.1.4), and the remaining one comprises several new variants of a common strategy (Section 9.1.1).

9.1.1 Missingness Based on Own Values

The Missingness Based on Own Values (MBOV) is the most common strategy to generate missing values under the MNAR mechanism. It makes the assumption that the missingness at a value depends on itself. Therefore, the lower values of a selected feature are removed upon a certain missing rate [112]. Alternatively, the higher values can be chosen instead [124]. For reasons previously stated, MBOV can not be applied to nominal features.

Besides considering the default behavior of this strategy, two variants are introduced in this work to better mimic real-world scenarios. A first variant adds a stochastic component to the generation, which is controlled by a scalar p that represents the fraction of the missing values to be randomly chosen, therefore following the MCAR mechanism. The remaining missing values are generated under MNAR assumptions, with the lower values being removed. The key idea is to add a certain amount of randomness to an otherwise deterministic MNAR mechanism. The scalar p controls the level of randomness: for $p = 1$, the generation is entirely random, which means the missing values would all be under the MCAR mechanism; on the other hand, for $p = 0$, the generation would follow the default deterministic MNAR behavior. Therefore, to keep most of the missing values under MNAR assumptions, this value should satisfy $0 < p < 0.5$. The second variant is an alternative to the default behavior where the values of a selected feature closest to middle are removed instead of the lowest or highest values. The goal is to simulate a scenario where extreme values are more often reported, while the more common values (i.e., closer to the mean/median) are more easily kept out. As an example, in an healthcare context, physicians tend to report values considered to be out of the expected domain, but often disregard values within the normal range. An example of these variants is available in Figure 9.1.

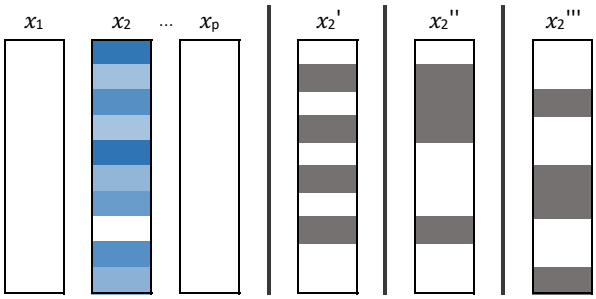

Figure 9.1: Example of the MBOV variants with a 40% missing rate. Gray instances of x_2 are missing values based on lower-values removal with $p = 0$ (x_2') and $p = 0.25$ (x_2'') values under MCAR, and middlemost-values removal (x_2'''). The magnitude of x_2 values is represented by a gradient of the blue color.

9.1.2 Missingness Based on Unobserved Values

The Missingness Based on Unobserved Values (MBUV) is based on a common MAR generation strategy, where the missing values are related with an observed feature of the dataset. To transpose this strategy for a MNAR context, this relation must be established with an unobserved feature (i.e., a feature not available in the dataset). Therefore, a new random continuous feature is generated, following a normal distribution, and its extreme instances (i.e., those with the lowest and highest values) are set to have missing values on a selected feature from the dataset. The approach is similar to the Missing depending on unobserved Variables (MuOv) from [32], with two main differences: MuOv provides multivariate generation by performing the described logic for several features, while MBUV is univariate (although it can be extended for multivariate scenarios easily, as seen in Section 9.1.4); and MuOv has a random process to choose the instances injected with the missing values, while MBUV divides the desired missing rate in half and selects the edge instances upon this threshold. This strategy is applicable to any data type (including nominal features), since it only requires the unobserved feature to be ordered. An example of this strategy is available in Figure 9.2.

9.1.3 Missingness Based on Intra-Relation

The Missingness Based on Intra-Relation (MBIR) is a novel strategy that tries to identify a likely MAR relation and transform it to MNAR. The approach relies on two features of the dataset: the one that will have the missing values (f_{miss}) and another one used to establish the MAR relation (f_{obs}). The f_{miss} feature is selected a priori and can be of any type, but the f_{obs} feature is selected in an iterative process. The instances containing the lowest values of each continuous and ordinal feature in the dataset (with the exception of f_{miss}) are set to be missing for the f_{miss} feature, upon a certain missing rate. An auxiliary missing indicator feature (f_{ind}) is created for f_{miss}, being set to 1

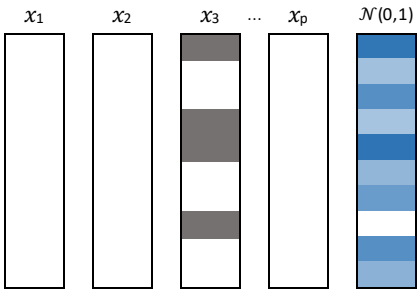

Figure 9.2: Example of the MBUV strategy with a 40% missing rate. Gray instances of x_3 are missing values, which are based on the edge values of the new random continuous feature (rightmost one). The magnitude of its values is represented by a gradient of the blue color.

for the instances where the values are missing, and set to 0 for the remaining ones. The differences between the values of f_{obs} for both groups ($f_{\text{ind}} = 1$ and $f_{\text{ind}} = 0$) are then assessed through statistical tests, and if, the differences are found to be significant, the generated missingness is likely to be MAR. From all the iterations which meet the previous criteria, the one with the most significant differences is chosen and the respective f_{obs} feature is deleted. As stated before, the goal is to transform a likely MAR scenario into MNAR, which is achieved by deleting f_{obs} since the relation found with an existent feature of the dataset becomes a relation with an unobserved feature. Pseudo code of the described MBIR procedure is presented in Algorithm 4.

Algorithm 4 Pseudocode for the MBIR procedure.

Input: *dataset*, f_{miss}, *mr*
Output: *amputed_dataset*

1: **for** each f_{obs} in continuous and ordinal features of *dataset* $\setminus \{f_{\text{miss}}\}$ **do**
2: Find the instances $inst_{\text{mr}}$ with the lower mr% values of f_{obs}
3: Set the f_{miss} values to be missing on the $inst_{\text{mr}}$ instances
4: Create the missing indicator f_{ind} for f_{miss} ($f_{\text{ind}} = 1$ when missing, $f_{\text{ind}} = 0$ otherwise)
5: Find if the differences between f_{obs} values for $f_{\text{ind}} = 1$ and $f_{\text{ind}} = 0$ are statistically significant
6: **if** differences are statistically significant **then**
7: Save f_{obs} to $MAR_{f_{\text{obs}}}$
8: **end if**
9: **end for**
10: Select the f_{obs} from $MAR_{f_{\text{obs}}}$ associated with the iteration that presented the most statistically significant differences
11: Create *amputed_dataset* by applying lines 2 and 3 to the *dataset*
12: Delete the selected f_{obs} feature from *amputed_dataset*
13: **return** *amputed_dataset*

The MBIR strategy relies on statistical tests to identify scenarios that are likely to be MAR. To make that identification more resilient, two different statistical approaches are

proposed here: a classic method, based on frequentist statistics, and a method based on Bayesian statistics. Both approaches are intended for two-sample scenarios, and evaluate if both samples are significantly different. For the frequentist approach, the Mann-Whitney rank test [61] is used to evaluate if the f_{obs} values for $f_{\text{ind}} = 1$ and $f_{\text{ind}} = 0$ are different with a statistical significance level of $\alpha = 0.05$. This test is non-parametric and does not require a normality assumption, which is an advantage when comparing to the well-known t-test. For the Bayesian approach, a Bayes factor computed for a two-sample design [68] is used to evaluate the same two groups of f_{obs} values. Following the significance scale proposed in [49], Bayes factors ≥ 10 show, at least, strong evidence of differences between the samples.

9.1.4 Missingness Based on Own and Unobserved Values

The Missingness Based on Own and Unobserved Values (MBOUV) is a combination of the MBOV and MBUV approaches to perform multivariate generation of missing values. The desired missingness rate is considered global for the entire dataset and is randomly distributed by all its features, leading to different levels of missingness in those features. Afterwards, the generation follows two rules: the MBUV is applied to all nominal features and to half of the continuous ones, while the MBOV (with lower values removal) is applied to the remaining half of the features. Both approaches are applied in an iterative way, and the split of continuous features is performed randomly.

9.2 Benchmark Setup

To properly conduct the benchmark study, a comprehensive baseline of state-of-the-art imputation methods was chosen. These methods are briefly described in Table 9.1. Furthermore, the autoencoder-based methods (DAE and VAE) require the hyper-parameterization of their neural network architectures, which are described in Table 9.2.

The experiment was conducted with 10 public medical datasets, which address different domains of clinical research based on routinely collected data. As previously stated, this type of data is ideal since it often suffers from missing data under MNAR [74]. All these datasets are public and available at the UCI[1] Machine Learning Repository. Their diverse characteristics, both in terms of the number of instances and types of features, are presented in Table 9.3.

The used datasets are complete (i.e., do not have missing data) and, during the experiment, they are injected with missing values under MNAR mechanisms using the approaches described in Section 9.1 (a process often called amputation). This approach was applied

[1] `https://archive.ics.uci.edu/ml/datasets.php`

Table 9.1: Baseline of imputation methods used for the benchmark.

Method	**Description**
Denoising Autoencoder (DAE)	An autoencoder is a neural network that learns a compressed representation of the input data, aiming to output a reconstruction of that same data [18]. A DAE is trained with a corrupted version of the input data, which is created by adding noise. In the missing data context, the "noise" is assumed to be the removal of values [22, 118].
Variational Autoencoder (VAE)	Autoencoder trained to learn a Gaussian multivariate latent representation of the input data, generating new values by sampling from this Gaussian. [44, 65].
k-Nearest Neighbors (kNN) Imputation with $k = 5$	Finds the k nearest neighbors of the record with missing values (using the Euclidean distance) and imputes with the average values (for continuous features) or the most common ones (for categorical features) from the k neighbors [95].
Mean/Mode Imputation	The mean (for continuous features) or the mode (for categorical features) is used to impute.
Multiple Imputation by Chained Equations (MICE)	Fits iteratively several regression models through a multiple imputation strategy, assuming as the dependent features the ones containing missing data [15].
SoftImpute	Matrix completion process, achieved through a nuclear-norm regularization iterative approach, where the missing data is imputed with estimated values obtained with a soft-thresholded SVD [64].

Table 9.2: Architecture details of the autoencoder-based methods.

Aspect	**Configuration**
DAE Structure	Two hidden layers with 10 units and ReLU activation function.
VAE Structure	Two hidden layers with 10 units, plus the three layers needed for the generative process: two "parallel" layers (mean and variance) with 10 units, connected to another one that samples new points from the latent Gaussian distribution.
Optimization Algorithm	Adam, with the MSE loss and a learning rate of 0.001.
Training Procedure	Batches of 64 instances, for a maximum of 2000 epochs, early stopping, and learning rate reduction of 80% when the validation loss shows no improvements in 100 consecutive epochs.
Overfitting Mitigation	Each hidden layer under L2 regularization with 0.01 weight.
Network Initialization	Network weights are randomly sampled from a truncated normal distribution centered at zero.
Output	Sigmoid activation function, since the data is normalized to the range $[0, 1]$.

iteratively to each feature, excepted for the MBOUV since it provides multivariate amputation. The imputation results were assessed through the mean absolute error (MAE) between the complete dataset (i.e., ground truth) and the imputed one. For nominal

Table 9.3: Details of the medical datasets used in the study.

Dataset	# Instances	# Features	
		Categorical	Continuous
bc-coimbra	116	1	9
cleveland	303	7	7
cmc	1473	8	2
ctg	2126	2	23
pima	768	1	8
saheart	462	2	8
thyroid	7200	16	6
transfusion	748	1	4
vertebral	310	1	5
wisconsin	569	1	30

features, this metric is applied to the one-hot encoded vectors, and the results are then transformed into error rates that are in the same scale of the continuous and ordinal features. For aggregation purposes, the overall MAE of a dataset is calculated by averaging the MAE of all its features. A high-level representation of this procedure is shown in Figure 9.3.

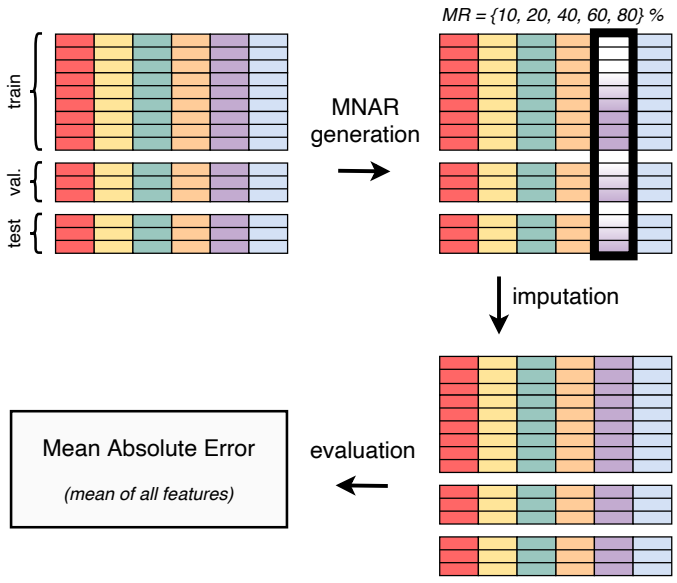

Figure 9.3: High-level representation of the benchmark setup.

All datasets were normalized to the interval $[0, 1]$ and split into train (50%), validation (20%) and test (30%) sets, which are used by the methods that require a training procedure. The data split was performed in a stratified manner to ensure equal missingness rates and MNAR assumptions for all sets. The one-hot encoding procedure was applied to the categorical features after the missing data injection. The benchmark study covered five missing rates (10%, 20%, 40%, 60% and 80%), which allows a complete analysis of the imputation results with different levels of missingness. The autoencoder-based methods

require a pre-imputation of the missing values, which was performed with the mean/mode of the features.

The experiment was independently executed 30 times. The datasets were shuffled in each execution, and the final results are the average of these 30 runs. Furthermore, the results are provided with 95% confidence intervals for significance evaluation: if in a given experiment, two imputation methods have MAE results with overlapping confidence intervals, the difference between those results is not considered statistically significant.

9.3 Experimental Results

In this section, we present and analyze the experimental results. To properly understand the imputation patterns in different types of data, the analysis is first performed separately for continuous and categorical features (Sections 9.3.1 and 9.3.2, respectively) and, afterwards, for all features together (Section 9.3.3). The results obtained for each applicable MNAR generation strategy are discussed individually in each of the Subsections, while the general conclusions are discussed in Section 9.4. The results are presented using charts where the MAE values for each missingness rate are plotted for all imputation approaches. The 95% confidence intervals of the results are displayed as translucent grey areas behind the plotted points and lines. When these intervals overlap for different imputation methods, the grey areas become more opaque, which means the differences between the results of those methods are not statistically significant. For a more detailed view of the experimental results, the raw data is attached in Section 9.5.

9.3.1 Continuous Features

The following subsections present the results for the MBOV, MBUV, and MBIR strategies applied to continuous features.

9.3.1.1 Missingness Based on Own Values (MBOV)

Figure 9.4 shows the results for the MBOV approach with the lowest values being removed. While the SoftImpute, followed by MICE, present the best results for lower rates (minimum MAEs of 0.084 and 0.1, respectively), AE-based imputation, particularly the VAE, performs better at higher rates (minimum MAE of 0.075), with the inflection point around the 40% rate.

Figure 9.5 shows the MBOV's results but with the highest values being removed. The main conclusions regarding the best and worst imputation approaches are essentially the same obtained for MBOV with lowest-value removal. The exception is the SoftImpute

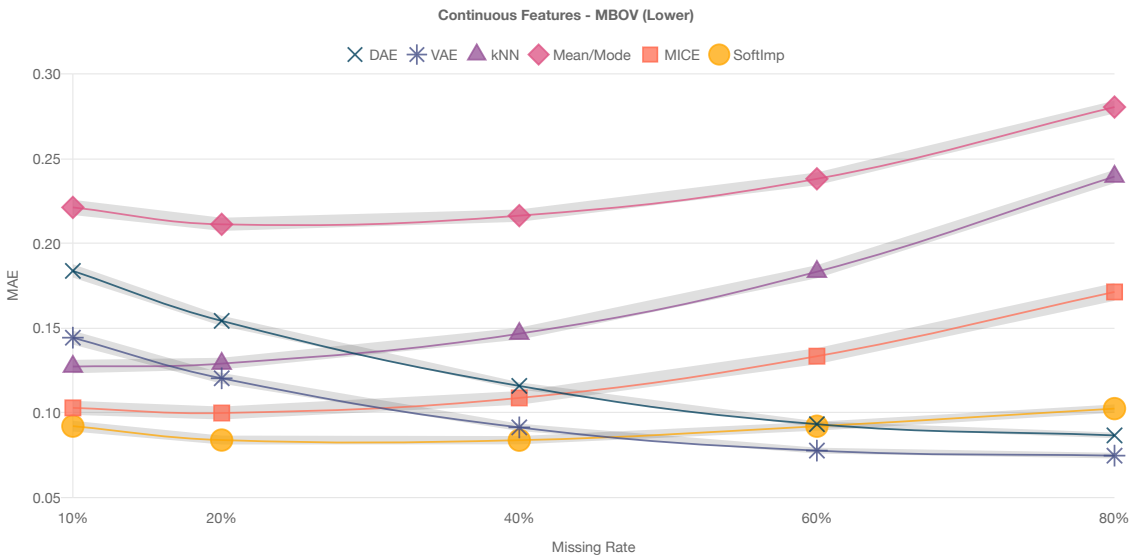

Figure 9.4: Results for MBOV with lowest values being removed, applied to continuous features.

method, which achieved significantly worse results in this setting (only the mean imputation presented worst results). Therefore, the MICE method obtained the best MAE (0.152) for lower missingness rates, while AE-based imputation achieved again the best results for the higher rates (minimum MAE of 0.087 for the VAE). The inflection point appears to be around the 30% rate in this setting. There is also an inversion of the MAE trend when the missing rate increases. For the MBOV with lowest-value removal, the MAE tends to increase gradually with the missing rate, except for the Autoencoder methods where the opposite happens. For the MBOV with higher-value removal, the MAE tends to decrease with the missing rate in all imputation methods. Although this change of behavior is probably related to the features' distribution, it is important to notice that the autoencoder methods are both consistent in terms of results and behavior. In both MBOV settings, the worst results were obtained by the mean imputation (maximum MAE of 0.332).

Figure 9.6 presents the results for the MBOV approach when a quarter of the missing values are randomly chosen ($p = 0.25$), therefore being under MCAR assumptions. Here, the SoftImpute method is consistently the best (minimum MAE of 0.08), being only slightly surpassed by the VAE for the 80% rate (MAE of 0.088). The worst results were again obtained by the mean imputation (maximum MAE of 0.201). The pattern of results is similar to the MBOV with lowest-value removal, but the 25% MCAR values lead to a significant improvement of the SoftImpute results.

Figure 9.7 shows the results for the MBOV approach with the middlemost values being removed. This is the MBOV setting where the pattern at the results is more different. Up to the 40% missingness rate, the results do not present significant differences between

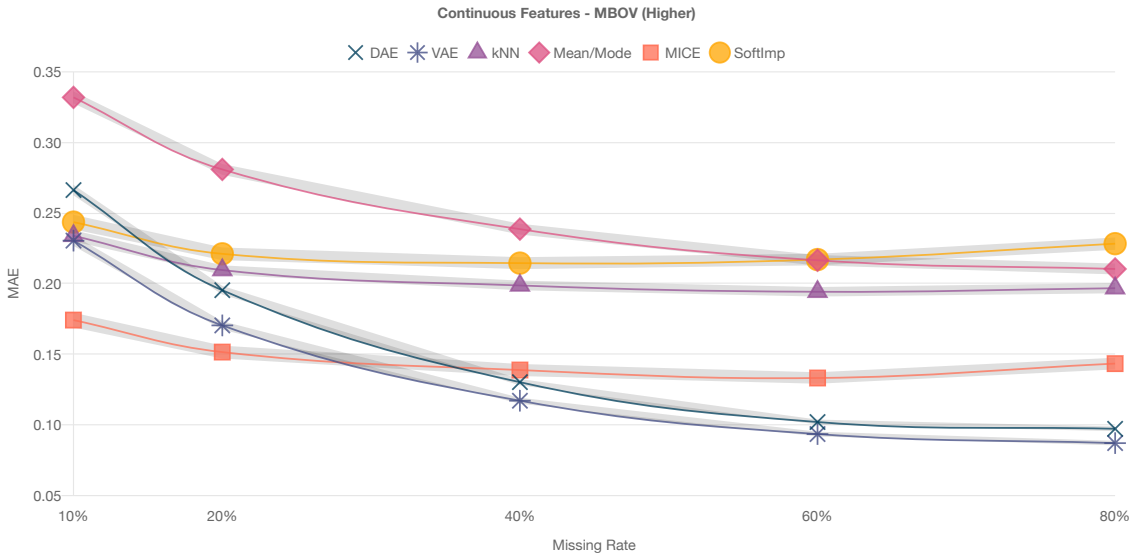

Figure 9.5: Results for MBOV with highest values being removed, applied to continuous features.

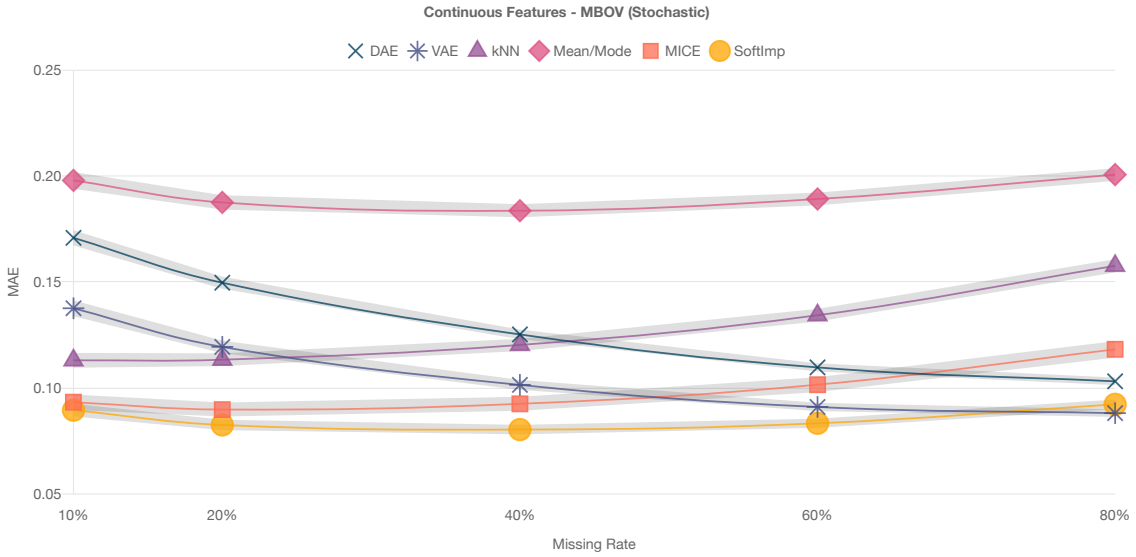

Figure 9.6: Results for MBOV with partial random missingness, applied to continuous features.

most imputation methods. The exception is the kNN imputation, which is consistently the worst for all rates (maximum MAE of 0.132). Nevertheless, for the 10% and 20% rates, the mean imputation shows the best results (minimum MAE of 0.038). Since the values being removed are the middlemost ones, there is a good chance they will be close to the features' mean, which can justify these results. However, above the 40% rate, the different imputation methods tend to produce results with more significant differences. For the 40%, 60%, and 80% rates, VAE outperforms the remaining methods (minimum MAE of 0.052).

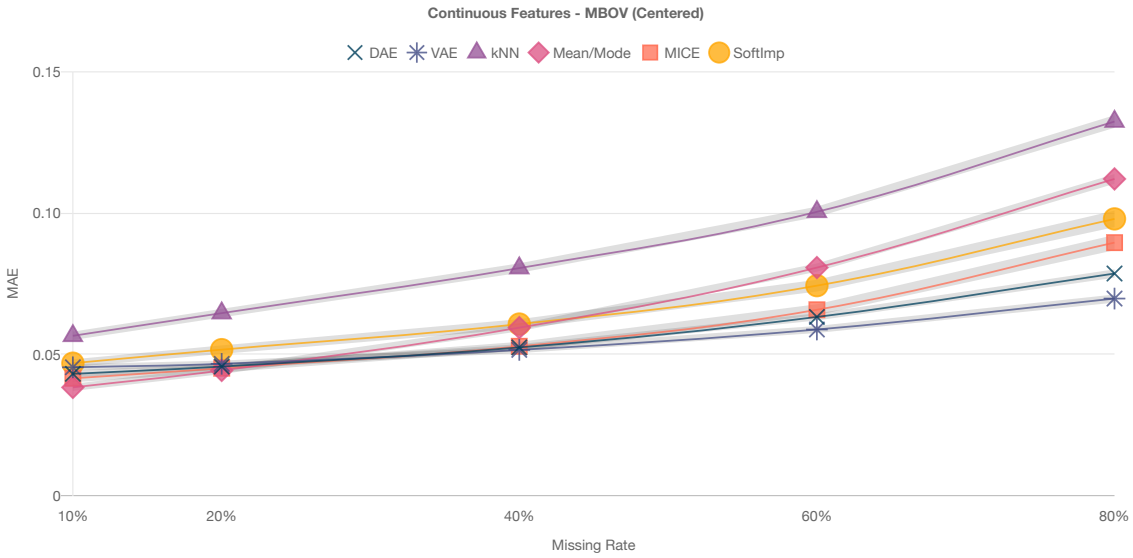

Figure 9.7: Results for MBOV with middlemost values removal, applied to continuous features.

9.3.1.2 Missingness Based on Unobserved Values (MBUV)

Figure 9.8 presents the results for the MBUV missingness mechanism. In this setting all imputation approaches present consistent MAE values for all rates, except the SoftImpute, which tends to perform worse with higher rates. The best imputation was achieved by the MICE method (minimum MAE of 0.062) and the worst by the mean imputation (maximum MAE of 0.119).

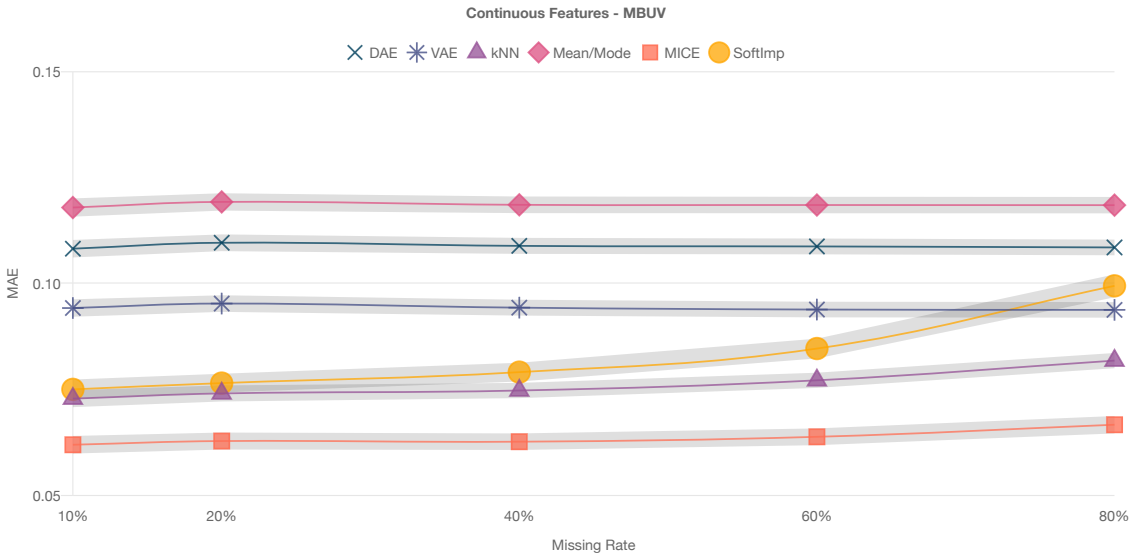

Figure 9.8: Results for MBUV applied to continuous features.

9.3.1.3 Missingness Based on Intra-Relation (MBIR)

Regarding the MBIR generation, Figures 9.9 and 9.10 show the results for the frequentist and Bayesian tests (see Subsection 9.1.3), respectively. The results obtained using both tests are essentially the same, with small MAE differences that can be accounted by the stochastic nature of some methods. The best and worst methods were respectively the MICE (minimum MAE of 0.065) and mean imputation (maximum MAE of 0.161), consistently throughout all missing rates. There is a slight increase in the MAE values when the missing rate grows, except for the AE-based methods, where the behavior is the opposite. Furthermore, the differences between the MAE values of the kNN and SoftImpute methods are not significant since their confidence intervals overlap often.

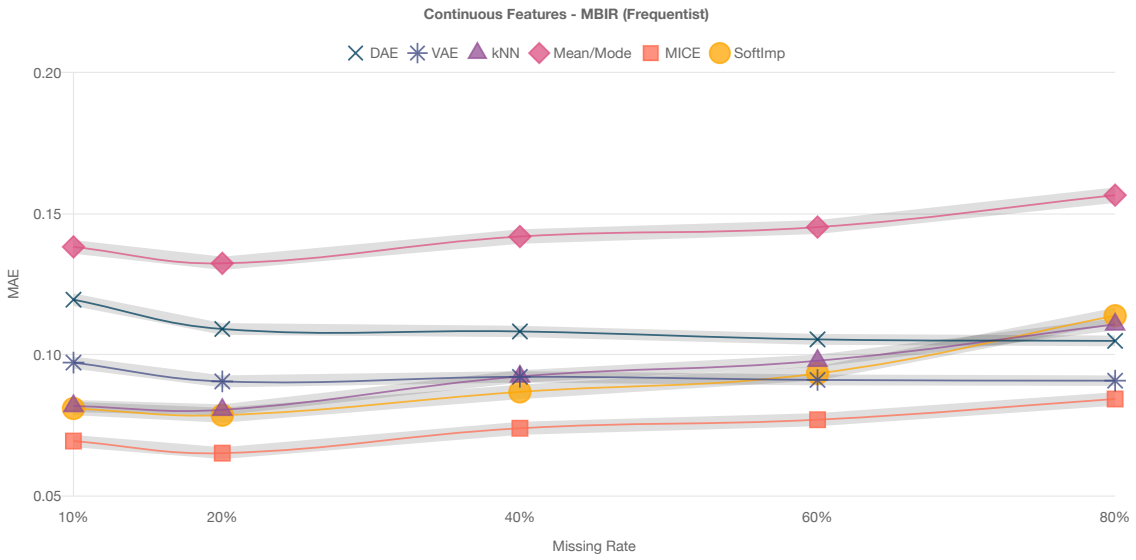

Figure 9.9: Results for MBIR with the frequentist test, applied to continuous features.

9.3.2 Categorical Features

The following subsections present the results for the MBUV and MBIR mechanisms applied to categorical features.

9.3.2.1 Missingness Based on Unobserved Values (MBUV)

Figure 9.11 presents the results for the MBUV approach. An immediately noticeable aspect is the large confidence intervals displayed by all imputation methods, which leads to an absence of significantly best and worst methods when considering the amount of overlap. The clearest differences are between the mode imputation and the SoftImpute methods, both appearing to be significantly worse than the remaining ones. Nevertheless, mode imputation shows the worst MAE up to the 60% rate, being surpassed by SoftImpute

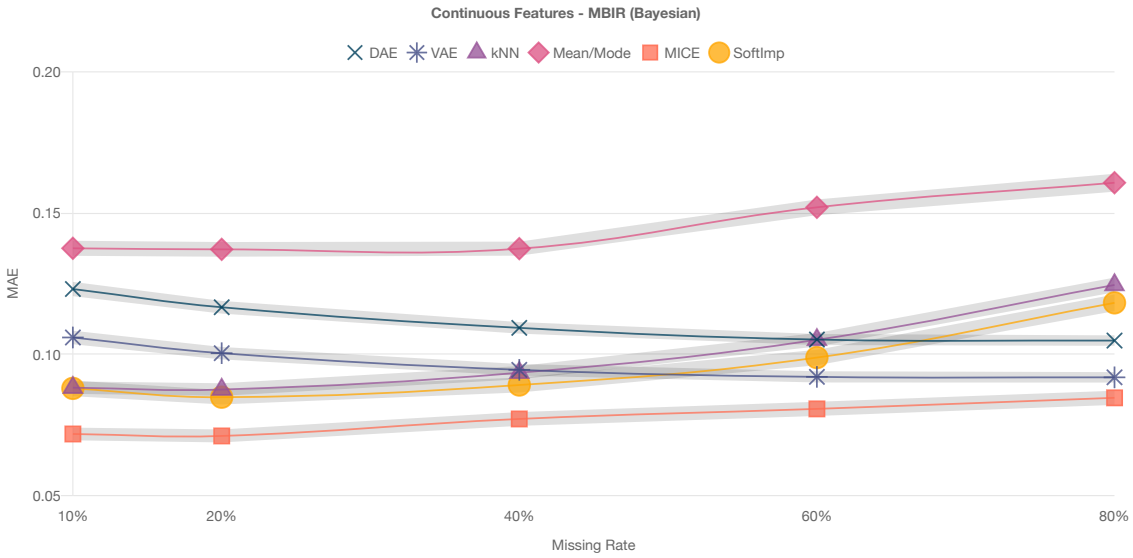

Figure 9.10: Results for MBIR with the Bayesian test, applied to continuous features.

after that rate (maximum MAEs of 0.188 and 0.205, respectively). The best MAE values are obtained with the MICE method, which is only slightly surpassed by the VAE after the 60% rate (minimum MAEs of 0.125 and 0.134, respectively).

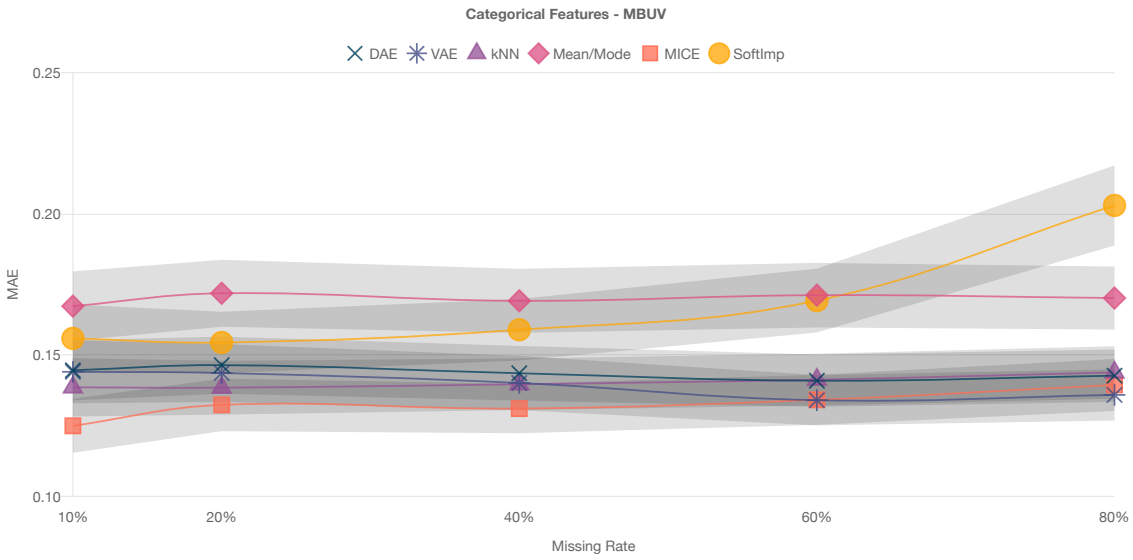

Figure 9.11: Results for MBUV applied to categorical features.

9.3.2.2 Missingness Based on Intra-Relation (MBIR)

Figures 9.12 and 9.13 present the MBIR results for the frequentist and Bayesian tests, respectively. Once again there are no significant differences between the MAE values obtained with both statistical tests. Moreover, the exact same patterns of results found for the MBUV mechanism are visible here: large confidence intervals that often overlap,

although this overlap tends to fade away above the 60% rate; the mode imputation and the SoftImpute methods still provide the worst MAE values, although the mode achieved worse results in this setting (maximum MAE of 0.221); MICE is generally the best method (minimum MAE of 0.111), although now it is surpassed by both AE-based methods beyond the 60% missingness rate (minimum MAE of 0.131 for the VAE).

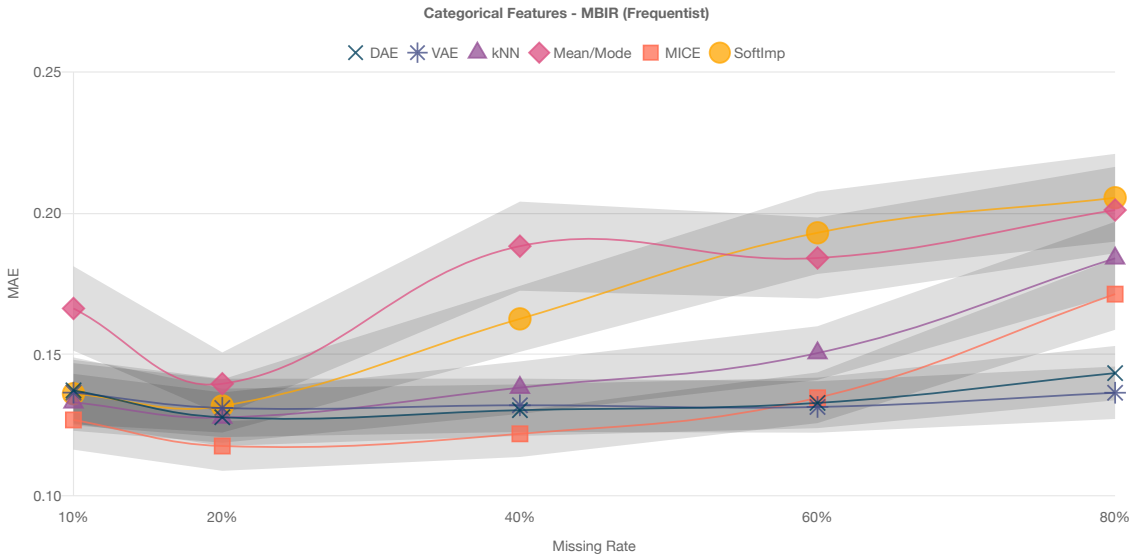

Figure 9.12: Results for MBIR with the frequentist test, applied to categorical features.

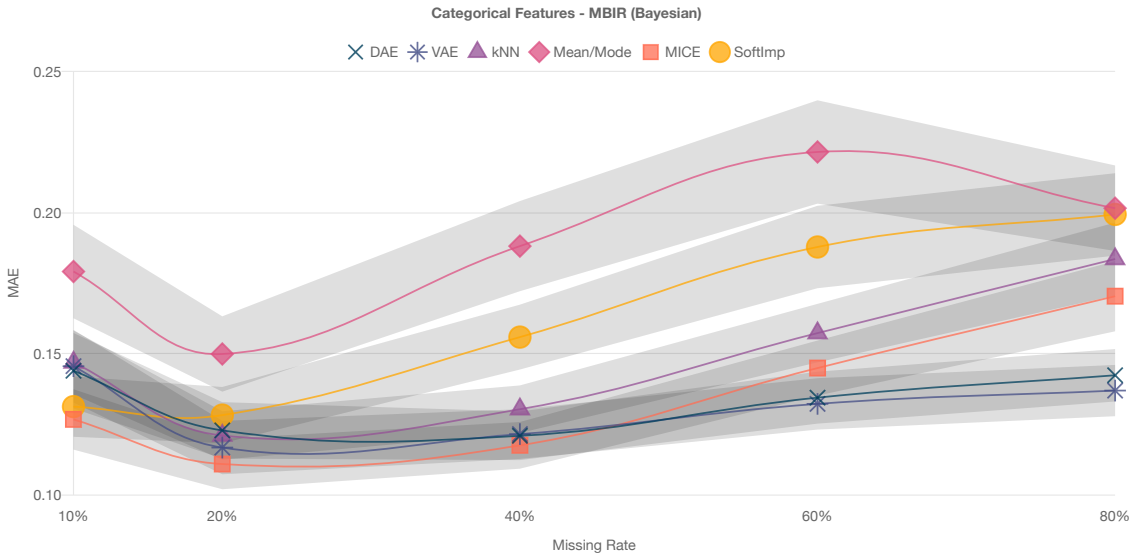

Figure 9.13: Results for MBIR with the Bayesian test, applied to categorical features.

9.3.3 Mixed Features

The following subsections present the results for the MBUV, MBIR, and MBOUV mechanisms applied to all features.

9.3.3.1 Missingness Based on Unobserved Values (MBUV)

Figure 9.14 shows the results for MBUV. The conclusions are the same obtained for this generation approach when only continuous features were considered: the MAE values are consistent for all missing rates, with the single exception of SoftImpute that performs worse with rates above 60%. The best imputation results were achieved by the MICE method (minimum MAE of 0.076) and the worst by the mean/mode imputation (maximum MAE of 0.131). In fact, when comparing with continuous features only, the MAE values in this mixed features setting are shifted towards higher errors. This is a clear indicator that the categorical features have a negative impact on the results, suggesting that all these methods tend to perform better with continuous features.

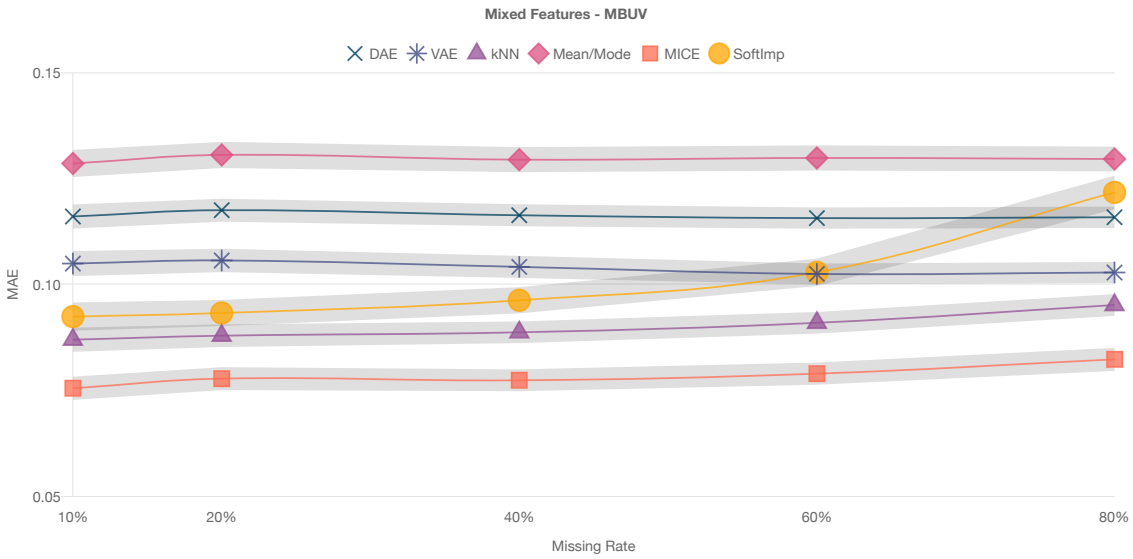

Figure 9.14: Results for MBUV applied to all (mixed) features.

9.3.3.2 Missingness Based on Intra-Relation (MBIR)

The MBIR results for the frequentist and Bayesian tests are, respectively, available on Figures 9.15 and 9.16. Following the pattern found with the continuous and categorical cases, there are no significant differences between both statistical tests. In fact, similarly to what happened with the MBUV results, the conclusions found here for MBIR are the same found when only the continuous features were considered: the best and worst methods continue to be the MICE (minimum MAE of 0.076) and mean/mode imputation (maximum MAE of 0.17) for all missing rates, expect for the 80% rate where the VAE surpasses MICE (minimum MAE of 0.101). The kNN and SoftImpute methods continue to not show significantly different results. Moreover, as previously found for the MBUV mechanism, the MAE results are shifted towards worse values in this mixed setting when comparing to the continuous features setting.

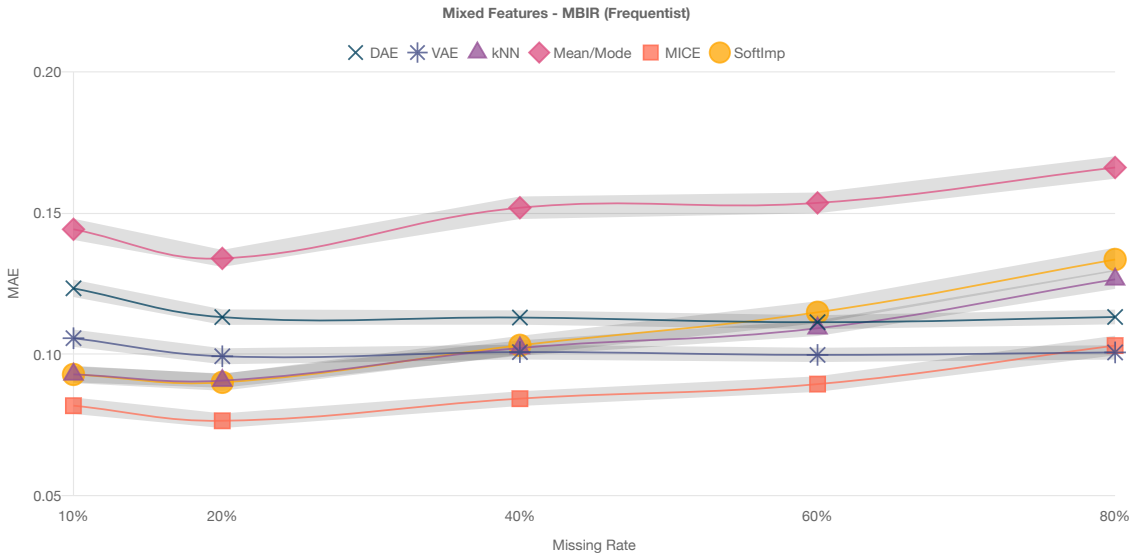

Figure 9.15: Results for MBIR with the frequentist test, applied to all (mixed) features.

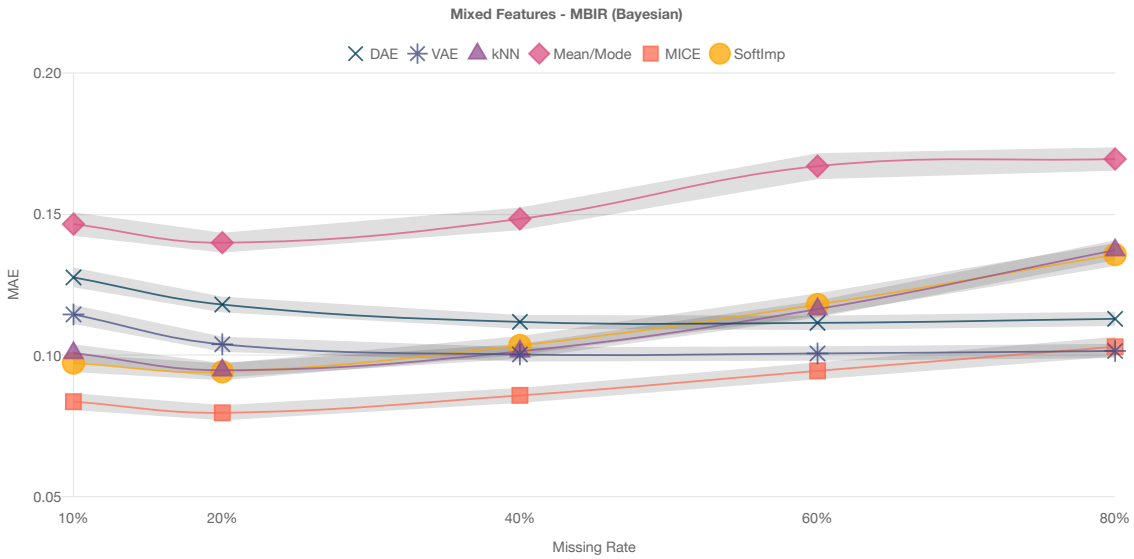

Figure 9.16: Results for MBIR with the Bayesian test, applied to all (mixed) features.

9.3.3.3 Missingness Based on Own and Unobserved Values (MBOUV)

Figure 9.17 presents the results for the MBOUV mechanism. Up to the 40% rate, there are no significant differences among the MAE values of the imputation methods, with the exception of mean/mode imputation, which clearly yields worse results (maximum MAE of 0.212). The MICE and kNN methods appear to provide slightly better results for the 10% and 20% rates, but the improvements are not significant (minimum MAEs of 0.113 and 0.114, respectively). However, for rates above 40%, there is a disparity between the imputation methods, with the AEs gradually showing significant improvements relatively to the remaining approaches (minimum MAE of 0.139 for the DAE). An interesting aspect

is that for higher rates, MICE presents the worst results by a considerable margin (the MAE value for the 80% rate was clipped). This behavior differs from previous experiments where MICE presented consistently good results and small confidence intervals for the univariate scenarios.

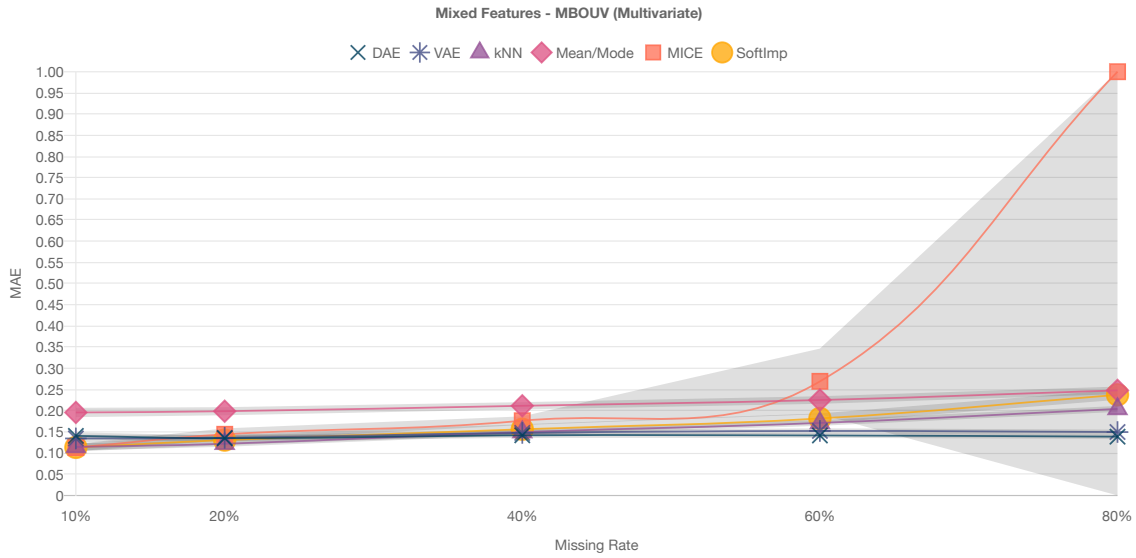

Figure 9.17: Results for MBOUV applied to all (mixed) features.

9.4 Discussion and Future Directions

After compiling all the results previously analyzed, there are several main conclusions that can be argued. To help this discussion, Table 9.4 presents a summary of the results by showing the best imputation method for each missingness rate, MNAR mechanism and data type. The methods marked with * presented statistically significant improvements compared to all the remaining ones.

When the rate is low or medium (up to 60%), the best imputation is very often achieved by the MICE method. However, for half of the MBOV settings, the SoftImpute method showed superior results, although MICE immediately follows with very close errors. The single exception is for MBOV with the middlemost values being removed, where the mean imputation surpassed these methods. Nevertheless, the superior consistency of the MICE results is undeniable with the low and medium levels of missingness. However, for larger rates (above 60%), MICE loses accuracy and is often surpassed by the AE-based methods (VAE and DAE), which are the best ones for these high levels of missingness with the MBOV, MBOUV, and all categorical features settings. A plausible justification is that, with lower missing rates, the amount of information available is still enough for methods usually applied in MAR contexts (such as MICE and SoftImpute) to still achieve the best results. In such settings, the MNAR assumptions are weaker, which leads to a lower level of

Table 9.4: Best imputation method for each missing rate, MNAR generation and data type. The methods which presented statistically significant improvements are marked with an asterisk.

Data Type	**MNAR Generation**	**Missing Rate**				
		10%	**20%**	**40%**	**60%**	**80%**
Continuous	MBOV (Lower)	SoftImp*	SoftImp*	SoftImp*	VAE*	VAE*
	MBOV (Higher)	MICE*	MICE*	VAE*	VAE*	VAE*
	MBOV (Stochastic)	SoftImp	SoftImp*	SoftImp*	SoftImp*	VAE*
	MBOV (Centered)	Mean/Mode*	Mean/Mode	VAE	VAE*	VAE*
	MBUV	MICE*	MICE*	MICE*	MICE*	MICE*
	MBIR (Frequentist)	MICE*	MICE*	MICE*	MICE*	MICE*
	MBIR (Bayesian)	MICE*	MICE*	MICE*	MICE*	MICE*
Categorical	MBUV	MICE	MICE	MICE	VAE	VAE
	MBIR (Frequentist)	MICE	MICE	MICE	VAE	VAE
	MBIR (Bayesian)	MICE	MICE	MICE	VAE	VAE
Mixed	MBUV	MICE*	MICE*	MICE*	MICE*	MICE*
	MBIR (Frequentist)	MICE*	MICE*	MICE*	MICE*	VAE
	MBIR (Bayesian)	MICE*	MICE*	MICE*	MICE*	VAE
	MBOUV (Multivariate)	MICE	KNN	DAE	DAE	DAE

uncertainty while performing the imputation, since this process is eased by the available information. This is visible in the MBOV setting with a stochastic component, where 25% of the missing values are under MCAR (which weakens the MNAR assumptions) and the results for SoftImpute and MICE improve in all missing rates. Furthermore, the multiple imputation strategy followed by the MICE method helps to deal with the more noticeable uncertainty in the medium levels of missingness, which is another justification for its superiority.

For higher rates, the AE-based methods show higher resilience to noisy (i.e., corrupted) data and more generalization capabilities, which leads to better results. For example, in several MNAR scenarios applied to continuous data, their results tend to improve when the missing rate increases. This is a clear indicator of how well AEs deal with corrupted data, to the point where they improve their results with more noise mostly because it leads to less overfitting and improved generalization [120]. Besides, AEs can also capture more complex patterns since they rely on non-linear activation functions, while methods such as MICE use linear models. This is particularly relevant with high missing rates since the mechanism describing the missing values is expected to be more complex and uncertain. Moreover, AEs do not assume that the data follow a specific distribution, which is also helpful since the MNAR mechanism introduces deviations in the data distributions, and

these deviations tend to increase when the missing rates are high. However, AEs also present a major limitation: they need to be trained with complete data in order to be applied to incomplete data. In a real-world scenario, this may be impossible to satisfy, since the datasets may not contain enough complete data to properly train the network. In this case, using AEs becomes unfeasible.

Analyzing the significance of the differences between imputation methods, while the results for the continuous features present low variance and are significantly different, the opposite behavior is found for categorical features: the confidence intervals are large for all imputation methods and consistently overlap. This pattern of results, associated with higher MAE values, show that all the imputation methods are more suitable for continuous features and most provide similar and lower imputation quality for the categorical ones. Regarding the consistency of the results throughout different missingness rates, there is a general tendency where the imputation methods gradually perform worse when the level of missingness increases. This is an expected behavior since higher missing rates lead to more uncertainty in the imputation process. However, as previously explained, the AE-based methods show the opposite trend for most scenarios with continuous features, which makes them a natural candidate for these high levels of missingness.

Finally, the approaches here proposed for artificial generation of MNAR provide an ample experimental setup that mimics different characteristics of this mechanism. Although there is some consistency among the best and worst imputation approaches within different generation approaches, the results tend to be different both in terms of trend and error magnitude. The single exception would be the use of frequentist and Bayesian tests in the MBIR approach, since both lead to similar results. Apart from this exception, the experimental results show the importance of properly simulating the missing data scenarios, with real-world-based and diverse strategies, in order to obtain meaningful and trustworthy results.

In conclusion, these benchmark results can be summarized in the following guidelines:

- For low and medium levels of missingness in univariate settings, the MICE and SoftImpute methods should be the go-to solution. For multivariate scenarios, MICE should be avoided;
- For higher missing rates, AE-based methods should be considered, particularly when there is enough complete data to properly train the network. Otherwise, MICE and SoftImpute continue to be good solutions;
- These three methods (MICE, SoftImpute, and AE) appear to deal well with departures from the MNAR assumptions, which makes them suitable for scenarios where missing data mechanisms are mixed;
- The kNN and mean/mode imputation are still among the most widely used methods

[32]. However, they should be avoided in MNAR settings. This conclusion comes without surprise, since they are often referred to as methods for MAR and MCAR;

- All these methods can be applied to mixed data types, but they are more suitable for continuous data. New approaches more tailored for categorical features are needed. Although the imputation of categorical features can be transformed into a classification problem, this solution does not provide a global imputation strategy for the dataset;
- It is not possible to state that a specific generation strategy is better at mimicking the MNAR nature since they all cover different characteristics of this mechanism. Instead, a mix of these strategies should be used to obtain more representative and reliable results.

In the future several directions can be tackled to further explore and improve the imputation of MNAR values. The benchmark studies should be further extended to accommodate settings where the missing mechanism is a composition of MNAR with MCAR and/or MAR, which is also a setting often found in real-world data. Furthermore, the impact of post-processing approaches sometimes applied to the MNAR imputation can also be studied and considered in the benchmark (e.g., sensitivity analysis and the application of the delta-adjustment method). Regarding the artificial generation of MNAR, there are specific settings of this mechanism that can be incorporated into new strategies. For example, for data domains which contain sensitive attributes, it is reasonable that the mechanism may only be applied to those types of attributes, while other mechanisms can be used for the remaining features. This would create a concept of sensitive-aware MNAR generation.

9.5 Supplementary Tables

For a more detailed view of the experimental results, the raw data is attached in Tables 9.5, 9.6 and 9.7 for the continuous, categorical, and mixed analysis, respectively. Each table presents the mean absolute error (MAE) values and their 95% confidence intervals for each imputation method, missing rate and Missing Not At Random (MNAR) generation strategy.

Table 9.5: Results for continuous features. The MAE values and their 95% confidence intervals are presented for each imputation method.

MNAR Generation	Missing Rate	DAE		VAE		kNN		Mean/Mode		MICE		SoftImp	
		μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)
MBOV (Lower)	10%	0.184	(0.18;0.19)	0.144	(0.14;0.15)	0.127	(0.12;0.13)	0.221	(0.22;0.23)	0.103	(0.10;0.11)	0.092	(0.09;0.10)
	20%	0.154	(0.15;0.16)	0.120	(0.12;0.12)	0.129	(0.13;0.13)	0.211	(0.21;0.22)	0.100	(0.10;0.10)	0.084	(0.08;0.09)
	40%	0.116	(0.11;0.12)	0.091	(0.09;0.09)	0.147	(0.14;0.15)	0.216	(0.21;0.22)	0.109	(0.10;0.11)	0.084	(0.08;0.09)
	60%	0.093	(0.09;0.09)	0.078	(0.08;0.08)	0.183	(0.18;0.19)	0.238	(0.23;0.24)	0.133	(0.13;0.14)	0.092	(0.09;0.09)
	80%	0.087	(0.08;0.09)	0.075	(0.07;0.08)	0.239	(0.24;0.24)	0.280	(0.28;0.28)	0.171	(0.17;0.18)	0.102	(0.10;0.10)
MBOV (Higher)	10%	0.266	(0.26;0.27)	0.230	(0.23;0.23)	0.234	(0.23;0.24)	0.332	(0.33;0.34)	0.174	(0.17;0.18)	0.244	(0.24;0.25)
	20%	0.195	(0.19;0.20)	0.170	(0.17;0.17)	0.210	(0.21;0.21)	0.281	(0.28;0.28)	0.152	(0.15;0.16)	0.221	(0.22;0.23)
	40%	0.130	(0.13;0.13)	0.117	(0.12;0.12)	0.199	(0.20;0.20)	0.239	(0.23;0.24)	0.139	(0.13;0.14)	0.215	(0.21;0.22)
	60%	0.102	(0.10;0.10)	0.094	(0.09;0.10)	0.194	(0.19;0.20)	0.217	(0.21;0.22)	0.133	(0.13;0.14)	0.217	(0.21;0.22)
	80%	0.097	(0.10;0.10)	0.087	(0.09;0.09)	0.197	(0.19;0.20)	0.210	(0.21;0.21)	0.143	(0.14;0.15)	0.228	(0.22;0.23)
MBOV (Stochastic)	10%	0.171	(0.17;0.17)	0.138	(0.13;0.14)	0.113	(0.11;0.12)	0.198	(0.19;0.20)	0.093	(0.09;0.10)	0.090	(0.09;0.09)
	20%	0.150	(0.15;0.15)	0.119	(0.12;0.12)	0.113	(0.11;0.12)	0.188	(0.18;0.19)	0.090	(0.09;0.09)	0.083	(0.08;0.09)
	40%	0.125	(0.12;0.13)	0.101	(0.10;0.10)	0.120	(0.12;0.12)	0.184	(0.18;0.19)	0.093	(0.09;0.10)	0.080	(0.08;0.08)
	60%	0.110	(0.11;0.11)	0.091	(0.09;0.09)	0.134	(0.13;0.14)	0.189	(0.19;0.19)	0.102	(0.10;0.11)	0.083	(0.08;0.09)
	80%	0.103	(0.10;0.10)	0.088	(0.09;0.09)	0.158	(0.15;0.16)	0.201	(0.20;0.20)	0.118	(0.11;0.12)	0.092	(0.09;0.09)
MBOV (Centered)	10%	0.043	(0.04;0.04)	0.045	(0.04;0.05)	0.057	(0.06;0.06)	0.038	(0.04;0.04)	0.041	(0.04;0.04)	0.047	(0.05;0.05)
	20%	0.046	(0.04;0.05)	0.047	(0.05;0.05)	0.065	(0.06;0.07)	0.044	(0.04;0.05)	0.045	(0.04;0.05)	0.052	(0.05;0.05)
	40%	0.052	(0.05;0.05)	0.052	(0.05;0.05)	0.080	(0.08;0.08)	0.059	(0.06;0.06)	0.053	(0.05;0.05)	0.061	(0.06;0.06)
	60%	0.063	(0.06;0.06)	0.059	(0.06;0.06)	0.100	(0.10;0.10)	0.081	(0.08;0.08)	0.066	(0.06;0.07)	0.074	(0.07;0.08)
	80%	0.079	(0.08;0.08)	0.070	(0.07;0.07)	0.132	(0.13;0.13)	0.112	(0.11;0.11)	0.090	(0.09;0.09)	0.098	(0.09;0.10)
MBUV	10%	0.108	(0.11;0.11)	0.094	(0.09;0.10)	0.073	(0.07;0.07)	0.118	(0.12;0.12)	0.062	(0.06;0.06)	0.075	(0.07;0.08)
	20%	0.110	(0.11;0.11)	0.095	(0.09;0.10)	0.074	(0.07;0.08)	0.119	(0.12;0.12)	0.063	(0.06;0.06)	0.077	(0.07;0.08)
	40%	0.109	(0.11;0.11)	0.094	(0.09;0.10)	0.075	(0.07;0.08)	0.119	(0.12;0.12)	0.063	(0.06;0.06)	0.079	(0.08;0.08)
	60%	0.109	(0.11;0.11)	0.094	(0.09;0.10)	0.077	(0.08;0.08)	0.119	(0.12;0.12)	0.064	(0.06;0.07)	0.085	(0.08;0.09)
	80%	0.109	(0.11;0.11)	0.094	(0.09;0.10)	0.082	(0.08;0.08)	0.118	(0.12;0.12)	0.067	(0.06;0.07)	0.099	(0.10;0.10)
MBIR (Frequentist)	10%	0.120	(0.12;0.12)	0.097	(0.10;0.10)	0.082	(0.08;0.08)	0.138	(0.14;0.14)	0.069	(0.07;0.07)	0.081	(0.08;0.08)
	20%	0.109	(0.11;0.11)	0.091	(0.09;0.09)	0.081	(0.08;0.08)	0.132	(0.13;0.13)	0.065	(0.06;0.07)	0.079	(0.08;0.08)
	40%	0.108	(0.11;0.11)	0.092	(0.09;0.09)	0.092	(0.09;0.09)	0.142	(0.14;0.14)	0.074	(0.07;0.08)	0.087	(0.08;0.09)
	60%	0.105	(0.10;0.11)	0.091	(0.09;0.09)	0.098	(0.10;0.10)	0.145	(0.14;0.15)	0.077	(0.07;0.08)	0.093	(0.09;0.10)
	80%	0.105	(0.10;0.11)	0.091	(0.09;0.09)	0.111	(0.11;0.11)	0.156	(0.15;0.16)	0.084	(0.08;0.09)	0.114	(0.11;0.12)
MBIR (Bayesian)	10%	0.123	(0.12;0.13)	0.106	(0.10;0.11)	0.088	(0.09;0.09)	0.138	(0.13;0.14)	0.072	(0.07;0.07)	0.088	(0.09;0.09)
	20%	0.117	(0.11;0.12)	0.100	(0.10;0.10)	0.088	(0.09;0.09)	0.137	(0.13;0.14)	0.071	(0.07;0.07)	0.085	(0.08;0.09)
	40%	0.109	(0.11;0.11)	0.094	(0.09;0.10)	0.093	(0.09;0.10)	0.137	(0.13;0.14)	0.077	(0.07;0.08)	0.089	(0.09;0.09)
	60%	0.105	(0.10;0.11)	0.092	(0.09;0.09)	0.105	(0.10;0.11)	0.152	(0.15;0.15)	0.081	(0.08;0.08)	0.099	(0.10;0.10)
	80%	0.105	(0.10;0.11)	0.092	(0.09;0.09)	0.125	(0.12;0.13)	0.161	(0.16;0.16)	0.085	(0.08;0.09)	0.118	(0.12;0.12)

Table 9.6: Results for categorical features. The MAE values and their 95% confidence intervals are presented for each imputation method.

MNAR Generation	Missing Rate	DAE		VAE		kNN		Mean/Mode		MICE		SoftImp	
		μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)
MBUV	10%	0.145	(0.13;0.16)	0.144	(0.13;0.16)	0.139	(0.13;0.15)	0.167	(0.16;0.18)	0.125	(0.12;0.13)	0.156	(0.14;0.17)
	20%	0.146	(0.14;0.16)	0.144	(0.13;0.15)	0.138	(0.13;0.15)	0.172	(0.16;0.18)	0.132	(0.12;0.14)	0.154	(0.14;0.17)
	40%	0.144	(0.13;0.15)	0.140	(0.13;0.15)	0.140	(0.13;0.15)	0.169	(0.16;0.18)	0.131	(0.12;0.14)	0.159	(0.15;0.17)
	60%	0.141	(0.13;0.15)	0.134	(0.13;0.14)	0.141	(0.13;0.15)	0.171	(0.16;0.18)	0.134	(0.13;0.14)	0.169	(0.16;0.18)
	80%	0.143	(0.13;0.15)	0.136	(0.13;0.14)	0.144	(0.13;0.15)	0.170	(0.16;0.18)	0.139	(0.13;0.15)	0.203	(0.19;0.22)
MBIR (Frequentist)	10%	0.137	(0.13;0.15)	0.136	(0.12;0.15)	0.133	(0.12;0.14)	0.166	(0.15;0.18)	0.127	(0.12;0.14)	0.136	(0.13;0.15)
	20%	0.128	(0.12;0.14)	0.131	(0.12;0.14)	0.128	(0.12;0.14)	0.140	(0.13;0.15)	0.118	(0.11;0.13)	0.132	(0.12;0.14)
	40%	0.130	(0.12;0.14)	0.132	(0.12;0.14)	0.138	(0.13;0.15)	0.188	(0.17;0.20)	0.122	(0.11;0.13)	0.163	(0.15;0.17)
	60%	0.133	(0.12;0.14)	0.131	(0.12;0.14)	0.150	(0.14;0.16)	0.184	(0.17;0.20)	0.135	(0.13;0.14)	0.193	(0.18;0.21)
	80%	0.143	(0.13;0.15)	0.136	(0.13;0.15)	0.184	(0.17;0.20)	0.201	(0.19;0.22)	0.171	(0.16;0.18)	0.205	(0.19;0.22)
MBIR (Bayesian)	10%	0.144	(0.13;0.16)	0.146	(0.13;0.16)	0.147	(0.14;0.16)	0.179	(0.16;0.20)	0.127	(0.12;0.14)	0.131	(0.12;0.14)
	20%	0.123	(0.11;0.13)	0.117	(0.11;0.13)	0.121	(0.11;0.13)	0.150	(0.14;0.16)	0.111	(0.10;0.12)	0.128	(0.12;0.14)
	40%	0.121	(0.11;0.13)	0.122	(0.11;0.13)	0.130	(0.12;0.14)	0.188	(0.17;0.20)	0.118	(0.11;0.13)	0.156	(0.14;0.17)
	60%	0.134	(0.13;0.14)	0.132	(0.12;0.14)	0.157	(0.15;0.17)	0.221	(0.20;0.24)	0.145	(0.14;0.15)	0.188	(0.17;0.20)
	80%	0.142	(0.13;0.15)	0.137	(0.13;0.15)	0.184	(0.17;0.20)	0.202	(0.19;0.22)	0.170	(0.16;0.18)	0.199	(0.18;0.21)

Table 9.7: Results for all (mixed) features. The MAE values and their 95% confidence intervals are presented for each imputation method.

MNAR Generation	Missing Rate	DAE		VAE		kNN		Mean/Mode		MICE		SoftImp	
		μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)	μ	(95% CI)
MBUV	10%	0.116	(0.11;0.12)	0.105	(0.10;0.11)	0.087	(0.08;0.09)	0.129	(0.13;0.13)	0.076	(0.07;0.08)	0.092	(0.09;0.10)
	20%	0.118	(0.11;0.12)	0.106	(0.10;0.11)	0.088	(0.09;0.09)	0.131	(0.13;0.13)	0.078	(0.08;0.08)	0.093	(0.09;0.10)
	40%	0.116	(0.11;0.12)	0.104	(0.10;0.11)	0.089	(0.09;0.09)	0.129	(0.13;0.13)	0.077	(0.07;0.08)	0.096	(0.09;0.10)
	60%	0.116	(0.11;0.12)	0.103	(0.10;0.10)	0.091	(0.09;0.09)	0.130	(0.13;0.13)	0.079	(0.08;0.08)	0.103	(0.10;0.11)
	80%	0.116	(0.11;0.12)	0.103	(0.10;0.11)	0.095	(0.09;0.10)	0.130	(0.13;0.13)	0.082	(0.08;0.09)	0.122	(0.12;0.13)
MBIR (Frequentist)	10%	0.123	(0.12;0.13)	0.106	(0.10;0.11)	0.093	(0.09;0.10)	0.144	(0.14;0.15)	0.082	(0.08;0.08)	0.093	(0.09;0.10)
	20%	0.113	(0.11;0.12)	0.099	(0.10;0.10)	0.091	(0.09;0.09)	0.134	(0.13;0.14)	0.076	(0.07;0.08)	0.090	(0.09;0.09)
	40%	0.113	(0.11;0.12)	0.101	(0.10;0.10)	0.102	(0.10;0.10)	0.152	(0.15;0.16)	0.084	(0.08;0.09)	0.103	(0.10;0.11)
	60%	0.111	(0.11;0.11)	0.100	(0.10;0.10)	0.109	(0.11;0.11)	0.154	(0.15;0.16)	0.089	(0.09;0.09)	0.115	(0.11;0.12)
	80%	0.113	(0.11;0.12)	0.101	(0.10;0.10)	0.127	(0.12;0.13)	0.166	(0.16;0.17)	0.103	(0.10;0.11)	0.134	(0.13;0.14)
MBIR (Bayesian)	10%	0.128	(0.12;0.13)	0.114	(0.11;0.12)	0.101	(0.10;0.10)	0.146	(0.14;0.15)	0.084	(0.08;0.09)	0.097	(0.09;0.10)
	20%	0.118	(0.12;0.12)	0.104	(0.10;0.11)	0.095	(0.09;0.10)	0.140	(0.14;0.14)	0.080	(0.08;0.08)	0.094	(0.09;0.10)
	40%	0.112	(0.11;0.11)	0.100	(0.10;0.10)	0.101	(0.10;0.10)	0.148	(0.14;0.15)	0.086	(0.08;0.09)	0.104	(0.10;0.11)
	60%	0.112	(0.11;0.11)	0.101	(0.10;0.10)	0.116	(0.11;0.12)	0.167	(0.16;0.17)	0.095	(0.09;0.10)	0.118	(0.11;0.12)
	80%	0.113	(0.11;0.12)	0.102	(0.10;0.10)	0.137	(0.13;0.14)	0.170	(0.17;0.17)	0.103	(0.10;0.11)	0.136	(0.13;0.14)
MBOUV (Multivariate)	10%	0.141	(0.13;0.15)	0.134	(0.13;0.14)	0.114	(0.11;0.12)	0.196	(0.18;0.21)	0.113	(0.10;0.12)	0.113	(0.10;0.12)
	20%	0.135	(0.13;0.14)	0.136	(0.13;0.14)	0.122	(0.11;0.13)	0.199	(0.19;0.21)	0.144	(0.13;0.16)	0.131	(0.12;0.14)
	40%	0.142	(0.14;0.15)	0.148	(0.14;0.15)	0.149	(0.14;0.16)	0.212	(0.20;0.22)	0.176	(0.17;0.19)	0.156	(0.14;0.17)
	60%	0.142	(0.14;0.15)	0.152	(0.15;0.16)	0.171	(0.16;0.18)	0.225	(0.22;0.23)	0.269	(0.19;0.35)	0.182	(0.17;0.19)
	80%	0.139	(0.13;0.14)	0.150	(0.14;0.16)	0.204	(0.20;0.21)	0.248	(0.24;0.26)	1.017	(-0.25;2.29)	0.237	(0.23;0.25)

Part III

Final Remarks

This page is intentionally left blank.

Chapter 10

Conclusions

In this chapter, we summarize the conclusions from all the developed works while answering the two research questions that were the focus of this thesis. Furthermore, we also propose some future directions for this field.

RQ-1: Can we extend and improve autoencoder-based models to impute MNAR values?

From our comprehensive survey on the use of autoencoders for missing data imputation, we could conclude that autoencoders are among the deep learning models most often used for this task since they were able to provide, in general, better results than other statistical and machine learning state-of-the-art models. Furthermore, two additional findings were also important to understand the open challenges within these models: although not being targeted for the MNAR values, most works that used autoencoders to address this mechanism concluded that they provide better results than the competition; and the VAE variant was still starting to be explored and used for this purpose, although the existing works presented it as equally (or even more) promising for the imputation task.

We decided to use the VAE as a filter mechanism for the data used by other imputation techniques. This approach proved to be a good use of the VAE for MNAR since it was able to only retain the more relevant information for the imputation, which helped achieve better results when compared to the other imputation techniques. We also decided to extend VAEs to include a multiple imputation procedure, which is a technique that has proven to provide good results for MNAR by mitigating the level of uncertainty in the estimated values. This strategy proved to achieve great results for MNAR, outperforming other statistical and deep learning-based techniques. It also had a positive impact on the performance of the imputed data used in classification tasks. Finally, we decided to explore a different variant, namely the Siamese autoencoders. This variant had never been used for imputation purposes, and we extended it by adapting the triplet mining and loss

function for this task. We were able to significantly outperform the existing state-of-the-art strategies while targeting the MNAR nature since the model relies on a small amount of complete data to be trained.

In sum, we were able to extend and improve autoencoder-based models that provide better imputation quality for the MNAR mechanism. The proposed strategies can tackle directly and indirectly the MNAR characteristics, achieving significantly better results than the existing state-of-the-art statistical and machine learning methods.

RQ-2: Can we automate human-based strategies used to improve the estimation of MNAR values?

Since the existing imputation solutions tend to produce biased results for MNAR values, a common strategy to mitigate that bias is to pre-process or post-process the data with techniques that can tackle such issues. These techniques are often manually executed by humans, which can lead to them not being suitable for scenarios where the data is considerably large or the access to domain experts is limited. When it comes to pre-processing, a common strategy to overcome the MNAR limitations is to try to gather the data from the original source. For example, if we have a dataset with information from a medical study, we can try to contact the patients and redo the necessary steps to obtain the missing MNAR data. However, this highly depends on human effort and resources. We tried to automate this type of procedure by using for the imputation multiple data sources that share a certain level of information. In the example given before, it could be multiple datasets of different medical studies that share part of the observations. The idea of basing the imputation on more sources than the one being imputed directly targets the MNAR nature, namely its relationship with unobserved data. We were able to improve the imputation quality with this strategy, while also improving the performance of classification tasks.

For post-processing, a common technique applied is the delta-adjustment method. It consists of imputing the values assuming the MAR mechanism, and then manually adjusting the estimates to reduce the gap between the MAR assumptions and the real MNAR mechanism. Although the procedure is quite simple, it depends on domain experts that must analyze the imputation results and propose the necessary adjustments. We tried to automate this procedure by generating approximate delta-adjustment factors for each feature. We were able to improve the results of all the imputation methods tested, without needing inputs from any domain experts.

The typical experimental setup for missing data requires generating the missing values on complete datasets, in order to have a ground truth to compare the estimates. In that regard, the existing strategies to perform the generation of MNAR missing values are quite limited since they rely on removing the lower or higher values of continuous features. Since

this procedure is crucial to have realistic experimental setups, we also proposed a set of new artificial generation strategies for MNAR, which were then validated in a benchmark study. Although these strategies do not solve the MNAR limitations from the imputation perspective, they are trying to mimic human behaviors when it comes to how the MNAR values are generated.

In sum, we were able to automate human-based strategies used to improve estimates of MNAR values, both through pre-processing and post-processing procedures that provided significant enhancements to the imputation quality. Furthermore, we also contributed to new artificial generation strategies for MNAR, which are very relevant for the missing data community.

Future Work

When it comes to future work, there are a considerable number of advancements needed for MNAR. We believe that this thesis has promoted significant improvements in how to better deal with this mechanism. However, it still is understudied and requires more tailored solutions. Based on the work here described, we believe most of the strategies we proposed can be extended and applied for non-tabular data, particularly images, which is one of the future directions that we want to explore. Furthermore, we believe other deep learning models can also be leveraged to handle MNAR (e.g., generative adversarial networks), and that is another direction that we want to further investigate.

This page is intentionally left blank.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] Pedro Henriques Abreu, Miriam Seoane Santos, Miguel Henriques Abreu, Bruno Andrade, and Daniel Castro Silva. Predicting breast cancer recurrence using machine learning techniques: a systematic review. *ACM Computing Surveys (CSUR)*, 49:52, 2016.

[3] Nzar A Ali and Zhyan M Omer. Improving accuracy of missing data imputation in data mining. *Kurdistan Journal of Applied Research*, 2(3):66–73, 2017.

[4] Shun-ichi Amari and Si Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[5] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[6] GEAPA Batista and MC Monard. Experimental comparison of k-nearest neighbor and mean or mode imputation methods with the internal strategies used by c4. 5 and cn2 to treat missing data. *University of São Paulo*, 34, 2003.

[7] Gustavo EAPA Batista, Maria Carolina Monard, et al. A study of k-nearest neighbour as an imputation method. *HIS*, 87(251-260):48, 2002.

[8] Brett K Beaulieu-Jones and Jason H Moore. Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific Symposium on Biocomputing 2017*, pages 207–218, 2017.

[9] Brett K Beaulieu-Jones, Daniel R Lavage, John W Snyder, Jason H Moore, Sarah A Pendergrass, and Christopher R Bauer. Characterizing and managing missing structured data in electronic health records: data analysis. *JMIR Medical Informatics*, 6 (1):e11, 2018.

[10] Mark Belger, Josep Maria Haro, Catherine Reed, Michael Happich, Kristin Kahle-Wrobleski, Josep Maria Argimon, Giuseppe Bruno, Richard Dodel, Roy W Jones,

Bruno Vellas, et al. How to deal with missing longitudinal data in cost of illness analysis in alzheimer's disease—suggestions from the geras observational study. *BMC medical research methodology*, 16:1–11, 2016.

[11] Giuseppe Bonaccorso. *Machine Learning Algorithms*. Packt Publishing Ltd, 2017.

[12] Guillem Boquet, Jose Lopez Vicario, Antoni Morell, and Javier Serrano. Missing data in traffic estimation: A variational autoencoder imputation method. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2882–2886, 2019.

[13] Guillem Boquet, Antoni Morell, Javier Serrano, and Jose Lopez Vicario. A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection. *Transportation Research Part C: Emerging Technologies*, 115:102622, 2020.

[14] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating Sentences from a Continuous Space. In *Proceedings of the 20th Conference on Computational Natural Language Learning*, pages 10–21, 2016.

[15] S van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, pages 1–68, 2010.

[16] Giulia Carreras, Guido Miccinesi, Andrew Wilcock, Nancy Preston, Daan Nieboer, Luc Deliens, Mogensm Groenvold, Urska Lunder, Agnes van der Heide, and Michela Baccini. Missing not at random in end of life care studies: multiple imputation and sensitivity analysis on data from the action study. *BMC Medical Research Methodology*, 21(1):1–12, 2021.

[17] David Charte, Francisco Charte, Salvador García, María J del Jesus, and Francisco Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96, 2018.

[18] David Charte, Francisco Charte, Salvador García, María J del Jesus, and Francisco Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96, 2018.

[19] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[20] Nutan Chen, Justin Bayer, Sebastian Urban, and Patrick Van Der Smagt. Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In *IEEE-RAS 15th International Conference on Humanoid Robots*, pages 434–440, 2015.

[21] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021.

[22] Suvra Jyoti Choudhury and Nikhil R Pal. Imputation of missing data with neural networks for classification. *Knowledge-Based Systems*, 182:104838, 2019.

[23] Adriana Fonseca Costa, Miriam Seoane Santos, Jastin Pompeu Soares, and Pedro Henriques Abreu. Missing data imputation via denoising autoencoders: The untold story. In *17th International Symposium on Intelligent Data Analysis*, pages 87–98, 2018.

[24] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[25] Yanjie Duan, Yisheng Lv, Wenwen Kang, and Yifei Zhao. A deep learning based approach for traffic data imputation. In *IEEE 17th International Conference on Intelligent Transportation Systems*, pages 912–917, 2014.

[26] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72:168–181, 2016.

[27] Mohamed El Esawey, Ahmed Ibrahem Mosa, and Khaled Nasr. Estimation of daily bicycle traffic volumes using sparse data. *Computers, Environment and Urban Systems*, 54:195–203, 2015.

[28] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661, 2020.

[29] Celestino Ordóñez Galán, Fernando Sánchez Lasheras, Francisco Javier de Cos Juez, and Antonio Bernardo Sánchez. Missing data imputation of questionnaires by means of genetic algorithms with different fitness functions. *Journal of Computational and Applied Mathematics*, 311:704–717, 2017.

[30] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[31] Pedro J García-Laencina, Pedro Henriques Abreu, Miguel Henriques Abreu, and Noémia Afonoso. Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. *Computers in biology and medicine*, 59:125–133, 2015.

[32] Unai Garciarena and Roberto Santana. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, 89:52–65, 2017.

[33] Zoubin Ghahramani and Michael I Jordan. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems*, pages 120–127, 1994.

[34] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):215–220, 2000.

[35] Lovedeep Gondara and Ke Wang. Recovering loss to followup information using denoising autoencoders. In *2017 IEEE International Conference on Big Data*, pages 1936–1945, 2017.

[36] Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 260–272, 2018.

[37] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[38] Nguyen Xuan Hau, Truong Dinh Tu, et al. User-based autoencoder for qos prediction. In *7th IEEE International Conference on Software Engineering and Service Science*, pages 308–311, 2016.

[39] Rebecca A Hubbard, Jing Huang, Joanna Harton, Arman Oganisian, Grace Choi, Levon Utidjian, Ihuoma Eneli, L Charles Bailey, and Yong Chen. A bayesian latent class approach for ehr-based phenotyping. *Statistics in Medicine*, 38(1):74–87, 2019.

[40] Anil K Jain, Jianchang Mao, and KM Mohiuddin. Artificial neural networks: A tutorial. *Computer*, (3):31–44, 1996.

[41] Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology*, 17(1):1–10, 2017.

[42] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *Seventh International Conference on Affective Computing and Intelligent Interaction*, pages 202–208, 2017.

[43] Yao Jia, Chongyu Zhou, and Mehul Motani. Spatio-temporal autoencoder for feature learning in patient data with missing observations. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 886–890, 2017.

[44] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[45] Michael H Kutner, Christopher J Nachtsheim, John Neter, William Li, et al. *Applied linear statistical models*, volume 5. McGraw-Hill Irwin New York, 2005.

[46] Xiaochen Lai, Xia Wu, Liyong Zhang, Wei Lu, and Chongquan Zhong. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing*, 366:54–65, 2019.

[47] Finbarr P Leacy, Sian Floyd, Tom A Yates, and Ian R White. Analyses of sensitivity to the missing-at-random assumption using multiple imputation with delta adjustment: application to a tuberculosis/hiv prevalence survey with incomplete hiv-status data. *American Journal of Epidemiology*, 185(4):304–315, 2017.

[48] Jae-woong Lee and Jongwuk Lee. Idae: Imputation-boosted denoising autoencoder for collaborative filtering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2143–2146, 2017.

[49] Michael D Lee and Eric-Jan Wagenmakers. *Bayesian cognitive modeling: A practical course*. Cambridge university press, 2014.

[50] Baptiste Leurent, Manuel Gomes, Rita Faria, Stephen Morris, Richard Grieve, and James R Carpenter. Sensitivity analysis for not-at-random missing data in trial-based cost-effectiveness analysis: a tutorial. *PharmacoEconomics*, 36(8):889–901, 2018.

[51] Wei Li and Xiao-Hua Zhou. Identifiability and estimation of causal mediation effects with missing data. *Statistics in Medicine*, 36(25):3948–3965, 2017.

[52] Lifeng Lin and Haitao Chu. Bayesian multivariate meta-analysis of multiple factors. *Research Synthesis Methods*, 9(2):261–272, 2018.

[53] Roderick JA Little. A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association*, 83(404):1198–1202, 1988.

[54] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[55] Shigang Liu, Honghua Dai, and Min Gan. Information-decomposition-model-based missing value estimation for not missing at random dataset. *International Journal of Machine Learning and Cybernetics*, 9(1):85–95, 2018.

[56] Fabio Lobato, Claudomiro Sales, Igor Araujo, Vincent Tadaiesky, Lilian Dias, Leonardo Ramos, and Adamo Santana. Multi-objective genetic algorithm for missing data imputation. *Pattern Recognition Letters*, 68:126–131, 2015.

[57] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243, 2019.

[58] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[59] Salim Malek, Farid Melgani, Yakoub Bazi, and Naif Alajlan. Reconstructing cloud-contaminated multispectral images with contextualized autoencoder neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 56:2270–2282, 2018.

[60] Lucas Malla, Rafael Perera-Salazar, Emily McFadden, Morris Ogero, Kasia Stepniewska, and Mike English. Handling missing data in propensity score estimation in comparative effectiveness evaluations: a systematic review. *Journal of Comparative Effectiveness Research*, 7(3):271–279, 2018.

[61] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[62] Alexina J Mason, Manuel Gomes, Richard Grieve, and James R Carpenter. A bayesian framework for health economic evaluation in studies with missing data. *Health Economics*, 27(11):1670–1683, 2018.

[63] Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pages 4413–4423. PMLR, 2019.

[64] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[65] John T McCoy, Steve Kroon, and Lidia Auret. Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51:141–146, 2018.

[66] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[67] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[68] Richard D Morey, Jeffrey N Rouder, Michael S Pratte, and Paul L Speckman. Using MCMC chain outputs to efficiently estimate Bayes factors. *Journal of Mathematical Psychology*, 55(5):368–378, 2011.

[69] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[70] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[71] Fulufhelo V Nelwamondo, Shakir Mohamed, and Tshilidzi Marwala. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pages 1514–1521, 2007.

[72] Xiuli Ning, Yingcheng Xu, Xiaohong Gao, and Ying Li. Missing data of quality inspection imputation algorithm base on stacked denoising auto-encoder. In *IEEE 2nd International Conference on Big Data Analysis*, pages 84–88, 2017.

[73] Ruilin Pan, Tingsheng Yang, Jianhua Cao, Ke Lu, and Zhanchao Zhang. Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. *Applied Intelligence*, 43(3):614–632, 2015.

[74] Niels Peek and Pedro Pereira Rodrigues. Three controversies in health data science. *International Journal of Data Science and Analytics*, 6(3):261–269, 2018.

[75] Ricardo Cardoso Pereira, Pedro Henriques Abreu, Pedro Pereira Rodrigues, and Mário A. T. Figueiredo. Imputation of Data Missing Not At Random: Artificial Generation and Benchmark Analysis. *Expert Systems With Applications*.

[76] Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. MNAR Imputation with Distributed Healthcare Data. In *19th EPIA Conference on Artificial Intelligence (EPIA 2019)*, pages 184–195, 2019.

[77] Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. VAE-BRIDGE: Variational Autoencoder Filter for Bayesian Ridge Imputation of Missing Data. In *2020 International Joint Conference on Neural Networks (IJCNN 2020)*, pages 1–7, 2020.

[78] Ricardo Cardoso Pereira, Joana Cristo Santos, José Pereira Amorim, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. Missing Image Data Imputation using Variational Autoencoders with Weighted Loss. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2020)*, pages 475–480, 2020.

[79] Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. Reviewing Autoencoders for Missing Data Imputation: Technical Trends, Applications and Outcomes. *Journal of Artificial Intelligence Research*, 69:1255–1285, 2020.

[80] Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Partial Multiple Imputation with Variational Autoencoders: Tackling Not at Randomness in Healthcare Data. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4218–4227, 2022.

[81] Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Siamese Autoencoder-based Approach for Missing Data Imputation. In *International Conference on Computational Science 2023 (ICCS 2023)*, 2023.

[82] Ricardo Cardoso Pereira, Pedro Pereira Rodrigues, Mário A. T. Figueiredo, and Pedro Henriques Abreu. Automatic Delta-Adjustment Method applied to Missing Not At Random Imputation. In *International Conference on Computational Science 2023 (ICCS 2023)*, 2023.

[83] Ryan A Peterson and Joseph E Cavanaugh. Ordered quantile normalization: a semiparametric transformation built for the cross-validation era. *Journal of Applied Statistics*, pages 1–16, 2019.

[84] Tra My Pham, James R Carpenter, Tim P Morris, Angela M Wood, and Irene Petersen. Population-calibrated multiple imputation for a binary/categorical covariate in categorical regression models. *Statistics in Medicine*, 38(5):792–808, 2019.

[85] Yeping Lina Qiu, Hong Zheng, and Olivier Gevaert. Genomic data imputation with variational auto-encoders. *GigaScience*, 9(8), 2020.

[86] Panteha Hayati Rezvan, Katherine J Lee, and Julie A Simpson. Sensitivity analysis within multiple imputation framework using delta-adjustment: Application to longitudinal study of australian children. *Longitudinal and Life Course Studies*, 9(3): 259–278, 2018.

[87] Donald B Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

[88] Donald B Rubin. *Multiple Imputation for Nonresponse in Surveys*, volume 81. John Wiley & Sons, 2004.

[89] Seunghyoung Ryu, Minsoo Kim, and Hongseok Kim. Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access*, 8:40656–40666, 2020.

[90] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Synthesizing and reconstructing missing sensory modalities in behavioral context recognition. *Sensors*, 18:29–67, 2018.

[91] Daiji Sakurai, Yoshikazu Fukuyama, Adamo Santana, Kenya Murakami, and Tetsuro Matsui. Estimation of missing data of showcase using autoencoder. In *56th Annual Conference of the Society of Instrument and Control Engineers of Japan*, pages 1303–1307, 2017.

[92] Adrián Sánchez-Morales, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Values deletion to improve deep imputation processes. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 240–246, 2017.

[93] Adrián Sánchez-Morales, José-Luis Sancho-Gómez, Juan-Antonio Martínez-García, and Aníbal R Figueiras-Vidal. Improving deep learning performance with missing values via deletion and compensation. *Neural Computing and Applications*, 32: 13233–13244, 2019.

[94] Adrián Sánchez-Morales, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Complete autoencoders for classification with missing values. *Neural Computing and Applications*, pages 1–7, 2020.

[95] Miriam Seoane Santos, Pedro Henriques Abreu, Pedro J García-Laencina, Adélia Simão, and Armando Carvalho. A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *Journal of Biomedical Informatics*, 58:49–59, 2015.

[96] Miriam Seoane Santos, Ricardo Cardoso Pereira, Adriana Fonseca Costa, Jastin Pompeu Soares, João Santos, and Pedro Henriques Abreu. Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, 7:11651–11667, 2019.

[97] Miriam Seoane Santos, Ricardo Cardoso Pereira, Adriana Fonseca Costa, Jastin Pompeu Soares, João Santos, and Pedro Henriques Abreu. Generating Synthetic Missing Data: A Review by Missing Mechanism. *IEEE Access*, 7:11651–11667, 2019.

[98] Joseph L Schafer. *Analysis of incomplete multivariate data*. Chapman and Hall/CRC, 1997.

[99] Joseph L Schafer and John W Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147, 2002.

[100] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[101] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[102] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. Vigan: Missing view imputation with generative adversarial networks. In *IEEE International Conference on Big Data*, pages 766–775, 2017.

[103] Ming Shao, Zhengming Ding, and Yun Fu. Sparse low-rank fusion based deep features for missing modality face recognition. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–6, 2015.

[104] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[105] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[106] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[107] Bao Luo Sun, Lan Liu, Wang Miao, Kathleen Wirth, James Robins, and Eric J Tchetgen. Semiparametric estimation with data missing not at random using an instrumental variable. *Statistica Sinica*, 28(4):1965–1983, 2018.

[108] Ping-Tee Tan, Suzie Cro, Eleanor Van Vogt, Matyas Szigeti, and Victoria R Cornelius. A review of the use of controlled multiple imputation in randomised controlled trials with missing outcome data. *BMC Medical Research Methodology*, 21(1):1–17, 2021.

[109] Fei Tang and Hemant Ishwaran. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6):363–377, 2017.

[110] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1(Jun):211–244, 2001.

[111] Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1405–1414, 2017.

[112] Bhekisipho Twala. An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, 23(5):373–405, 2009.

[113] Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

[114] H Cevallos Valdiviezo and Stefan Van Aelst. Tree-based prediction on incomplete data using imputation or surrogate decisions. *Information Sciences*, 311:163–181, 2015.

[115] Stef Van Buuren. *Flexible imputation of missing data*. Chapman and Hall/CRC, 2018.

[116] Sander MJ van Kuijk, Wolfgang Viechtbauer, Louis L Peeters, and Luc Smits. Bias in regression coefficient estimates when assumptions for handling missing data are violated: a simulation study. *Epidemiology, Biostatistics and Public Health*, 13(1), 2016.

[117] Stijn Vansteelandt, James Carpenter, and Michael G Kenward. Analysis of incomplete data using inverse probability weighting and doubly robust estimators. *Methodology*, 6(1):37–48, 2010.

[118] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine learning*, pages 1096–1103, 2008.

[119] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[120] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12):3371–3408, 2010.

[121] Runmin Wei, Jingye Wang, Mingming Su, Erik Jia, Shaoqiu Chen, Tianlu Chen, and Yan Ni. Missing value imputation approach for mass spectrometry-based metabolomics data. *Scientific Reports*, 8(1):663, 2018.

[122] Ian R White, Patrick Royston, and Angela M Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in Medicine*, 30(4): 377–399, 2011.

[123] IR White, J Carpenter, and NJ Horton. A mean score method for sensitivity analysis to departures from the missing at random assumption in randomised trials. *Statistica Sinica*, 28(4):1985–2003, 2018.

[124] Jing Xia, Shengyu Zhang, Guolong Cai, Li Li, Qing Pan, Jing Yan, and Gangmin Ning. Adjusted weight voting algorithm for random forests in handling missing values. *Pattern Recognition*, 69:52–60, 2017.

[125] Ruimin Xie, Nabil Magbool Jan, Kuangrong Hao, Lei Chen, and Biao Huang. Supervised variational autoencoders for soft sensor modeling with missing data. *IEEE Transactions on Industrial Informatics*, 16:2820–2828, 2019.

[126] Jinsung Yoon, James Jordon, and Mihaela Schaar. GAIN: Missing Data Imputation using Generative Adversarial Nets. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5675–5684, 2018.

[127] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[128] Zhongheng Zhang, Linghong Cao, Yan Zhao, Ziyin Xu, Rangui Chen, Lukai Lv, and Ping Xu. Hospitalized patients with heart failure: integrating electronic healthcare records and external outcome data, 2020. URL `https://physionet.org/content/heart-failure-zigong/1.2/`.

[129] Liang Zhao, Zhikui Chen, Zhennan Yang, Yueming Hu, and Mohammad S Obaidat. Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. *IEEE Systems Journal*, 12:1610–1620, 2016.

[130] Bing Zhu, Changzheng He, and Panos Liatsis. A robust missing value imputation method for noisy data. *Applied Intelligence*, 36(1):61–74, 2012.