mDAE : modified Denoising AutoEncoder for missing data imputation

Anonymous authors
Paper under double-blind review

Abstract

This paper introduces a method based on Denoising AutoEncoder (DAE) for missing data imputation. The specificities of the proposed mDAE method result from a modification of the loss function and a straightforward procedure for choosing the hyper-parameters. An ablation study of these specificities demonstrates their relevance on several UCI Machine Learning Repository datasets for several types and proportions of missing values. This numerical study is completed by comparing eight other methods (four standard and four more recent), demonstrating the good behaviour of the mDAE method. A criterion called Mean Distance to Best (MDB) is proposed to globally compare the results of the methods on all datasets. According to this criterion, the mDAE method was consistently ranked among the top three methods (along with SoftImput and missForest), while the four more recent methods were systematically ranked last. The Python code of the numerical study will be available on GitHub so that results can be reproduced or generalized with other datasets and methods.

1 Introduction

With the rapid increase in data collection, missing values are a ubiquitous challenge across various domains. Data may be missing for several reasons. For instance, it was never collected, records were lost or merging several datasets failed. It is then generally necessary to deal with this problem before performing machine learning methods on these data. Several options are available to address this issue, including removing or recreating the missing values. Removing rows or columns containing missing data results in a considerable loss of information when missing data is distributed across multiple locations in the dataset. One usually prefers missing-data imputation, which consists of filling missing entries with estimated values using the observed data. Missing data imputation is a very active research area (Van Buuren, 2018; Little & Rubin, 2019) with more than 150 implementations available according to Mayer et al. (2021). This paper focuses on state-of-the-art imputation methods categorized as standard machine learning, deep learning or optimal transport. Methods based on standard machine learning include, among others, k-nearest neighbours (Troyanskaya et al., 2001), matrix completion via iterative soft-thresholded SVD (Mazumder et al., 2010), Multivariate Imputation by Chained Equations (Van Buuren & Groothuis-Oudshoorn, 2011) or MissForest (Stekhoven & Bühlmann, 2012). Methods based on deep learning include, among others, Generative Adversarial Networks (Goodfellow et al., 2014; Yoon et al., 2018), Variational AutoEncoders (Kingma & Welling, 2013; Ivanov et al., 2018; Mattei & Frellsen, 2019; Peis et al., 2022) and methods based on Denoising AutoEncoders (see e.g. the review of Pereira et al., 2020). One can also mention the recent works of Muzellec et al. (2020); Zhao et al. (2023) based on optimal transport.

This paper proposes a modified Denoising AutoEncoder (mDAE) dedicated to imputing missing data. AutoEncoders (AE) (Bengio et al., 2009) are artificial neural networks used to learn efficient representation of unlabeled data (encodings) and a decoding function that recreates the input data from the encoded representation. Denoising AutoEncoders (DAEs) were first proposed by Vincent et al. (2008) to recover, from noisy data, the original data without noise by corrupting the inputs of a standard AE. For example, inputs can be corrupted by masking noise where a fixed proportion of the inputs are randomly set to 0. DAEs, initially proposed for image processing, have also been used for missing data imputation (Duan et al., 2014; Gondara

& Wang, 2018; Ryu et al., 2020). In these papers, a DAE is trained on complete data to impute missing data during the test phase. However, when the missing values are spread over all the inputs, obtaining a subset of complete data large enough to train the network is impossible. To address this issue, we modified the loss function of the DAE to train it on incomplete data (mDAE).

The mDAE method is first evaluated via an ablation study to check the relevance of some of its components (modification of the loss function, choice of the hyperparameter by cross-validation, overcomplete structure). After this ablation study, the mDAE method is compared with eight imputation methods (4 based on standard machine learning and the others based on deep learning and optimal transport) along with the reference method of average imputation. The comparison is made on several datasets of the UCI Machine Learning Repository Dua & Graff (2017) and three missing data structures (Missing Completely at Random, Missing At Random, Missing Not At Random). Mean Distance to the Best (MDB) criterion is proposed to assess how well a method works overall on the considered datasets. This numerical study shows the good behaviour of the mDAE method (still in the top 3), the good behaviour of the SofImpute and missForest methods based on classical machine learning approaches and the poor behaviour of recent methods based on deep learning and optimal transport (still ranked in the bottom 4).

2 The mDAE method

AutoEncoders (AE) (Bengio et al., 2009) are well-known artificial neural networks used to learn efficient representation of unlabeled data via an encoding function and to recreate the input data via a decoding function. For tabular data, the input is a set of n observations $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\mathbb{R}^p$ which forms the rows of a data matrix $\mathbf{X} = (x_{ij})$ of dimension $n \times p$, where p is the number of features. The encoding function f_θ of a basic autoencoder (see Figure 1) transforms an input $\mathbf{x}_i \in \mathbb{R}^p$ into a latent vector $\mathbf{y}_i \in \mathbb{R}^q$:

$$\mathbf{y}_i = f_\theta(\mathbf{x}_i) = s(\mathbf{W}\mathbf{x}_i + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{q \times p}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^p$ is a bias vector and s is an activation function (e.g., ReLU or sigmoid). The decoding function $g_{\theta'}$ then transforms the latent vector $\mathbf{y}_i \in \mathbb{R}^q$ into an output $\mathbf{z}_i \in \mathbb{R}^p$:

$$\mathbf{z}_i = g_{\theta'}(\mathbf{y}_i) = s(\mathbf{W}'\mathbf{y}_i + \mathbf{b}'),$$

where $\mathbf{W}' \in \mathbb{R}^{p \times q}$ and $\mathbf{b}' \in \mathbb{R}^q$.

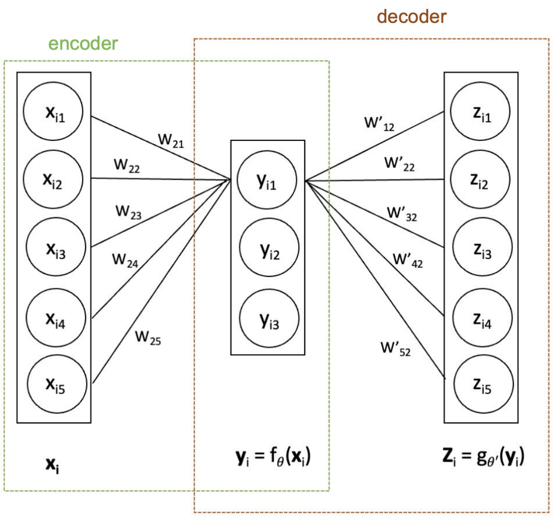


Figure 1: Scheme of a basic AutoEncoder (AE).

In general, autoencoders have more than one hidden layer and the parameters $\theta = (\mathbf{W}_1, ..., \mathbf{W}_K, \mathbf{b}_1, ..., \mathbf{b}_K)$ of the encoder and $\theta' = (\mathbf{W}'_1, ..., \mathbf{W}'_K, \mathbf{b}'_1, ..., \mathbf{b}'_K)$ of the decoder, are learned by minimization of the so called reconstruction loss. The reconstruction loss used to learn weights and biases of an autoencoder is often the L2 loss defined by:

$$\mathcal{L}_{AE} = \sum_{i=1}^{n} \underbrace{\|\mathbf{x}_i - (g_{\theta'} \circ f_\theta)(\mathbf{x}_i)\|^2}_{L(\mathbf{x}_i, \mathbf{z}_i)} = \|\mathbf{X} - \mathbf{Z}\|_F^2, \tag{1}$$

where L is the loss function defined here as the squared Euclidean distance between the input $\mathbf{x}_i$ and its reconstruction $\mathbf{z}_i = (g_{\theta'} \circ f_\theta)(\mathbf{x}_i)$, and $\|\mathbf{X} - \mathbf{Z}\|_F$ is the Frobenius norm between the data matrix $\mathbf{X}$ and its reconstructed matrix $\mathbf{Z}$. Note that this criterion will favor the reconstruction of features (columns of $\mathbf{X}$) with high variance. Therefore, the data matrix $\mathbf{X}$ is typically standardized.

Denoising AutoEncoders (DAE) (Vincent et al., 2008) are autoencoders defined to remove noise from a given input. To do this, an autoencoder is trained to output the original data using corrupted data in the input. The masking noise, for instance, is a corrupting process where each observation $\mathbf{x}_i$ is corrupted by randomly setting a proportion μ of its components to zero. Let $N(\mathbf{x}_i)$ denotes this corrupted version of $\mathbf{x}_i$. The loss $L(\mathbf{x}_i, \mathbf{z}_i)$ is here slightly different from the one in (1) as it compares the input $\mathbf{x}_i$ with the output $\mathbf{z}_i = (g_{\theta'} \circ f_\theta)(N(\mathbf{x}_i))$ obtained with corrupted observations $N(\mathbf{x}_i)$ (see Figure 2). The reconstruction loss minimized by a DAE is then:

$$\mathcal{L}_{DAE} = \sum_{i=1}^{n} \underbrace{\|\mathbf{x}_i - (g_{\theta'} \circ f_\theta)(N(\mathbf{x}_i))\|^2}_{L(\mathbf{x}_i, \mathbf{z}_i)} = \|\mathbf{X} - \mathbf{Z}\|_F^2 \tag{2}$$

Note that the proportion μ of the masking noise is a hyper-parameter that may need to be calibrated.

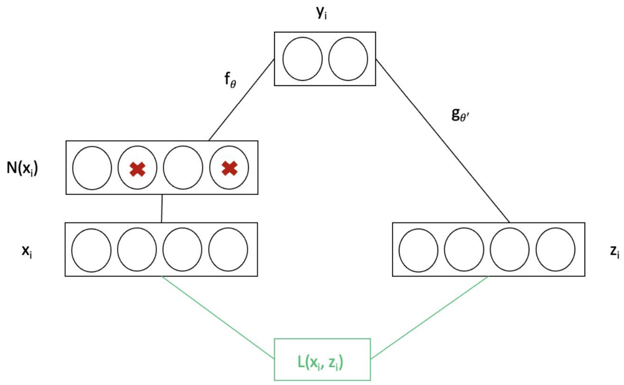


Figure 2: Scheme of a denoising AutoEncoder (DAE). Red crosses represent the values in $N(\mathbf{x}_i)$ randomly set to 0.

While DAE methods were first defined to remove noise from a given input, they can be used for missing values imputation. However, using a DAE requires a complete dataset for parameters learning. With missing data in the dataset, it is not possible. The following section shows how this problem can be managed by pre-imputing missing values and how a change in the loss function makes it possible to predict missing values more accurately.

2.1 Imputing missing values using the mDAE method

Let $\mathbf{X}$ be now a centred (or centred-reduced) incomplete data matrix and Ω be the set of indices $(i, j) \in \{1, ..., n\} \times \{1, ..., p\}$ where the values $\mathbf{x}_{ij}$ are not missing. A first idea for imputing missing values in $\mathbf{X}$ is to learn a DAE to reconstruct the missing values pre-imputed. The data matrix with pre-imputed values is noted $\tilde{\mathbf{X}}$. This pre-imputed matrix allows to have a complete data set for the training of the DAE. Because the data are centred, the mean value of each column is equal to zero, and the data matrix $\tilde{\mathbf{X}}$, pre-imputed by the column means, is the projection of $\mathbf{X}$ onto the observed entries:

$$\tilde{\mathbf{X}} = P_{\Omega}(\mathbf{X}) = \begin{cases} x_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{if } (i,j) \notin \Omega. \end{cases}$$

The reconstruction loss (2) minimized by a DAE to reconstruct $\tilde{\mathbf{X}}$ is then:

$$\mathcal{L}_{DAE} = \sum_{i=1}^{n} \|\tilde{\mathbf{x}}_i - (g_{\theta'} \circ f_{\theta})(N(\mathbf{x}_i))\|^2 = \|P_{\Omega}(\mathbf{X}) - \mathbf{Z}\|_F^2, \tag{3}$$

where $\mathbf{Z}$ is now the reconstruction of the pre-imputed matrix $\tilde{\mathbf{X}}$. However, using (3), the DAE learns to reconstruct zeros at the locations of the missing values, which is irrelevant (see Figure 3).

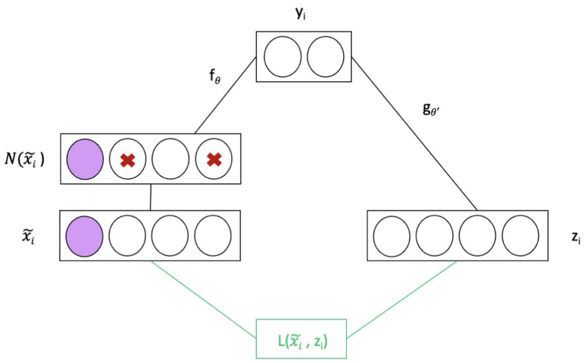


Figure 3: Scheme of a DAE directly applied on pre-imputed data. Violet dots in $\tilde{\mathbf{x}}_i$ represent the missing values set to 0. Red crosses in $N(\tilde{\mathbf{x}}_i)$ represent the values randomly set to 0.

Our proposal is then to modify the reconstruction error (3) to skip these locations (see Figure 4). The mDAE hereafter minimizes the modified reconstruction loss:

$$\mathcal{L}_{mDAE} = \|P_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{Z})\|_F^2, \tag{4}$$

and the complete data matrix imputed with this mDAE method is:

$$P_{\Omega}(\mathbf{X}) + P_{\Omega^{\perp}}(\mathbf{Z}), \tag{5}$$

where $\Omega^{\perp}$ is the set of indices $(i,j) \in \{1,...,n\} \times \{1,...,p\}$ where $\mathbf{x}_{ij}$ is missing.

Thanks to the corruption, the mDAE method learns the structure of the data with missing values to reconstruct the missing values of the data.

2.2 Choice of the hyper-parameter μ

The hyper-parameter μ of the mDAE method is the proportion of zeros used to corrupt the data with the masking noise (red crosses in $N(\tilde{\mathbf{x}}_i)$ in Figure 4). This hyper-parameter can be chosen randomly in a grid of values μ in $[0,1]$. Alternatively, it can be chosen through an optimized procedure to minimize an error of reconstruction of the missing values. For that purpose, the non-missing values of the original data are split into two sets: a training set to learn the parameters and a validation set to estimate the error of reconstruction of missing values. Let $V \subset \Omega$ be the subset of indices (i,j) of the validation set, drawn randomly from the set of observed entries Ω. For each value of μ in the grid, the error of reconstruction of the missing values is estimated using the following procedure:

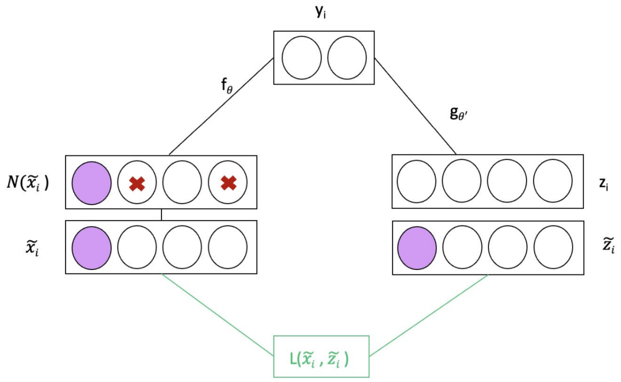


Figure 4: Scheme of a mDAE. Violet dots in $\tilde{\mathbf{x}}_i$ represent the missing values set to 0. Violet dots in $\tilde{\mathbf{z}}_i$ represent the predicted values set to 0. Red crosses in $N(\tilde{\mathbf{x}}_i)$ represent the values randomly set to 0

1. The parameters of the mDAE are learned on the training set $\Omega \setminus V$ by minimization of the reconstruction loss:
$$\mathcal{L}_{mDAE} = \|P_{\Omega\setminus V}(\mathbf{X}) - P_{\Omega\setminus V}(\mathbf{Z})\|_F^2, \tag{6}$$
where $\Omega \setminus V$ is the set of observed entries minus those drawn at random for the validation.
2. The mean squared error (MSE) of reconstruction of the missing values is estimated on the validation set by:
$$MSE_{val} = \frac{1}{|V|}\|P_V(\mathbf{X}) - P_V(\mathbf{Z})\|_F^2. \tag{7}$$
where $\mathbf{Z}$ is the matrix reconstructed with the mDAE learned on the training set $\Omega \setminus V$ and $|V|$ is the cardinal of the validation set.

The previous two steps are repeated B times (for the B draws of missing values) and the mean of the errors of reconstruction of the missing values is performed to get a more robust estimation.

2.3 Choice of the structure

Two families of structures are known for autoencoders. The undercomplete case where the hidden layer is smaller than the input layer and the overcomplete case where it is bigger. If overcomplete structure is not relevant with autoencoders, it is well-known that denoising autoencoders work well with overcomplete structures. Here, a grid of 6 simple structures (2 undercomplete and four overcomplete) is suggested to choose the "best" structure when using the mDAE method (see Figure 5). For each structure in this grid, the error of reconstruction of the missing values is estimated on validation data, using the same procedure as for the selection of the hyper-parameter μ (see section 2.2). Ideally, the hyper-parameter μ and the structure should be chosen simultaneously by exhaustively considering all possible combinations. However, alternative grid search exists, for instance, by sampling a given number of candidates from the parameter space.

3 Numerical study

The first part of this numerical study concerns the properties of the mDAE method. More specifically, an ablation study is conducted to verify the relevance of the choices made to construct this method. The second part compares the mDAE method with other well-known or more recent methods for imputing missing data.

All comparisons are made using seven complete (without missing values) numerical datasets from the UCI Machine Learning Repository Dua & Graff (2017) (see Table 1).To evaluate an imputation method, a certain proportion of each dataset is first artificially replaced by missing values. The artificial missing values are

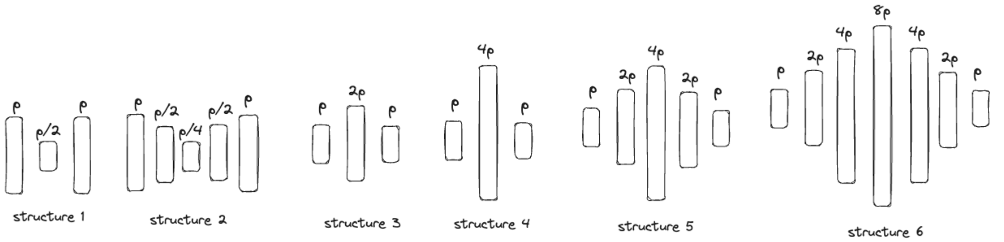


Figure 5: A grid of 6 simple structures where p is the number of units of the input layer.

drawn using either the MAR (Missing At Random), the MCAR (Missing Completely At Random) or the MNAR (Missing Not At Random) mechanism (see e.g. Rubin, 1976). Note that the MCAR and MAR missing values were generated using a logistic masking model as implemented in the GitHub repository of Muzellec. Then, for a given mask $\Omega^{\perp}$ of artificial missing values, the performance of the method is evaluated using the Root Mean Squared Error (RMSE) between the initial data matrix $\mathbf{X}$ and the reconstructed data matrix $\mathbf{Z}$ on $\Omega^{\perp}$:

$$RMSE = \sqrt{\frac{1}{|\Omega^{\perp}|}\|P_{\Omega^{\perp}}(\mathbf{X}) - P_{\Omega^{\perp}}(\mathbf{Z})\|_F^2}, \tag{8}$$

where $|\Omega^{\perp}|$ is the number of artificial missing values. To get more robust results, the process is repeated B times with B sets of artificial missing values drawn randomly using one of the three generation mechanisms. Finally, a method is evaluated by the mean and standard deviation of the B values of RMSE obtained with a certain proportion of artificial missing data and a certain mechanism of missing values (MAR, MCAR or MNAR). Note that all datasets are standardized (i.e. all features are centered and scaled to unit variance) prior to running the experiments, to give the same weights to all features in the analyses. All the results presented in this section are reproducible using Python code, which will be available on GitHub.

Table 1: The seven datasets used in the numerical study

Names	**Abreviations**	**Rows**	**Columns**
Breast cancer diagnostic	breast	509	30
Connectionist bench sonar	sonar	208	60
Ionosphere	iono	351	34
Blood transfusion	blood	748	4
Seeds	seeds	210	7
Climate model crashes	climate	540	18
Wine quality red	wine	1599	10

3.1 Ablation study of the mDAE method

An ablation study is a methodology used to evaluate the importance of different components of an algorithm, by comparing the results obtained with and without this component. Here, the following components of the mDAE method are studied:

- the use of the modified reconstruction loss (4) rather than the standard L_2 loss defined in (3),
- the use of an optimized value of the hyper-parameter μ (as described section 2.2) rather than a value chosen randomly in $[0, 1]$,

- the use of an overcomplete structure (the 5th structure in Figure 5) rather than an undercomplete structure (the 2nd structure in Figure 5).

Table 2 shows the results of the ablation study for the seven datasets and 20% of MCAR artificial missing values. The mean value over the B sets of artificial missing values (± the standard deviation) of the RMSE of reconstruction of the artificial missing values is calculated for each dataset with the mDAE method, with the method deprived of its loss function (i.e. with a standard L_2 loss function), with the method deprived of its optimized choice of μ (i.e. with a random choice), with the method deprived of its overcomplete structure (i.e. with an under complete structure). Each time, the loss of imputation quality (i.e. the increase of the mean RMSE) is measured between the mDAE without one of the three components (the modified loss, an optimized choice of μ or an overcomplete structure) and the complete mDAE. For instance, for the breast cancer dataset, using the standard L_2 loss increases the mean RMSE of $46.99\% = \frac{0.685-0.466}{0.466}$.

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	**0.466 ± 0.016**	1.007 ± 0.007	**0.656 ± 0.007**	**0.776 ± 0.018**	**0.496 ± 0.022**	**0.790 ± 0.030**	**0.701 ± 0.059**
mDAE w/o modified loss	0.685 ± 0.036 (46.996%)	**1.005 ± 0.008** (-0.199%)	0.988 ± 0.013 (50.610%)	0.808 ± 0.020 (4.124%)	0.587 ± 0.028 (18.347%)	0.828 ± 0.034 (4.810%)	0.755 ± 0.058 (7.703%)
mDAE w/o optimal μ	0.501 ± 0.043 (7.511%)	1.030 ± 0.013 (2.284%)	0.682 ± 0.049 (3.963%)	0.802 ± 0.039 (3.351%)	0.514 ± 0.054 (3.629%)	0.853 ± 0.033 (7.975%)	0.710 ± 0.055 (1.284%)
mDAE w/o overcomplete	0.500 ± 0.011 (7.296%)	1.147 ± 0.013 (13.903%)	0.699 ± 0.008 (6.555%)	0.808 ± 0.025 (4.124%)	0.671 ± 0.209 (35.282%)	0.932 ± 0.045 (17.975%)	0.960 ± 0.140 (36.947%)

Table 2: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 20% of MCAR artificial missing values. First row : results of the mDAE method (with the modified loss, the optimal choice of the hyper-parameter μ and with an overcomplete structure). Second row : results without (w/o) the modified loss (with the standard L_2 loss instead). Third row : results without (w/o) the optimal choice of μ (with random choice of μ instead). Fourth row : results without (w/o) overcomplete structure (with an undercomplete structure instead). The results in brackets are the growth rate of the average RMSE when the component under consideration is removed.

The results in Table 2 show that the mDAE method with its three components (first row) constantly reconstructs missing data better, except for climate data, where modifying the loss function does not improve the results. Not using the modified loss function (second row) increases the RMSE for the breast and seeds datasets by up to 50%. Using a random value of the hyper-parameter μ rather than an optimized one (third row) deteriorates the imputation quality for all datasets, but to a lesser extent (between 1 and 8% increase in RMSE). Using an undercomplete structure rather than an overcomplete one (fourth row) clearly increases the RMSE to around 35%

If the three components (modification of the loss, optimization of μ and overcomplete structure) seem relevant, the gain obtained by choosing the best μ in a grid rather than randomly in $[0, 1]$ does not seem very significant. This is an important result, as it allows the user to choose the hyper-parameter μ randomly to save computation time.

The results of ablation studies for other types of artificial missing values (MAR and MNAR) and other proportions of artificial missing values (20% and 40%) are given in Appendix A. These results confirm the importance of the modification of the loss function, the importance of choosing an overcomplete structure, and the more relative importance of choosing μ in a grid search rather than a random one.

3.2 Comparison with other methods

This section compares the mDAE method with four relatively classic and four more recent methods (see Table 3). The four first methods are KNN ((Troyanskaya et al., 2001)) where missing values are replaced by a weighted average of the k-nearest neighbours, SoftImput (Mazumder et al., 2010) based on iterative soft-thresholded SVD, and two iterative chained equation methods (Van Buuren & Groothuis-Oudshoorn, 2011) which model features with missing values as a function of the others: the missForest method (Stekhoven & Bühlmann, 2012) is based on Random Forests and the BayesianRidge method is based on ridge regressions, to estimate at each step the regression functions. The four others (more recent) methods in Table 3 are GAIN (Yoon et al., 2018) which is an adaptation of Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) to impute missing data, MIWAE Mattei & Frellsen (2019) which is an adaptation of Variational AutoEncoders (VAE) (Kingma & Welling, 2013), and two methods using optimal transport: the algorithm called Batch Sinkhorn Imputation proposed by Muzellec et al. (2020), and the method TDM proposed by Zhao et al. (2023).

For KNN and SoftImpute, the hyperparameters are selected through cross-validation. According to the implementations used for the two chained equation methods, the hyperparameters of the Bayesian ridge regressions are estimated during the fits of the model. The hyperparameters of the Random Forests are 100 trees, and all features are considered when looking for the best split (i.e., bagged trees). The hyperparameter settings recommended in the corresponding papers and implementations are used for the four last methods. For the mDAE method, the settings studied section 3.1 (choice of μ by crossvalidation and the overcomplete structure 5 of Figure 5) are used. More favourable settings for the mDAE method would have been to select μ and the structure by cross-validation on all possible parameter combinations. This approach was not adopted for computation time reasons in this numerical study.

Table 3: The methods used in the numerical study

Names	**Abreviations**
k-nearest neighbors[1]	knn
SoftImput[2]	si
missForest[3]	rf
BayesianRidge[3]	br
Generative Adversarial Imputation Network[4]	gain
Missing Data Importance Weighted Autoencoders[5]	miwae
Batch Sinkhorn Imputation[2]	skh
Transformed Distribution Matching for missing value imputation[6]	tdm

With these settings of hyperparameters, the eight methods of Table 3, as well as the mDAE method and the basic mean imputation method, are compared in Figure 6 on the 7 datasets and 20% of MCAR artificial missing values. The mean value (± the standard deviation) of the RMSE of reconstruction of the artificial missing values is plotted for each dataset and each method.

We note in Figure 6 that certain methods like SoftImpute (SI), missForest (RF) or mDAE work reasonably well on all datasets (no dataset where the RMSE value is much worse than others). It can also be noted that the mDAE method gives better or equivalent results on the 7 datasets than the four methods based on neural networks and optimal transport (gain, miwae, skh and tdm).

But no method always wins. In order to measure how a method performs globally well on several datasets, we propose to use a new metric called Mean Distance to the Best (MDB) hereafter. If I denotes the number of datasets and J the number of methods, the MDB of a method j is defined by:

$$MDB(j) = \frac{1}{I} \sum_{i=1}^{I} \left(R_{ij} - \min_{\ell=1\ldots J} R_{i\ell} \right) \tag{9}$$

---

[1] Available in the class KNNImputer, https://scikit-learn.org/stable/api/sklearn.impute.html

[2] https://github.com/BorisMuzellec/MissingDataOT

[3] Available in the class IterativeImputer, https://scikit-learn.org/stable/api/sklearn.impute.html

[4] https://github.com/jsyoon0823/GAIN

[5] https://github.com/pamattei/miwae

[6] https://github.com/hezgit/TDM

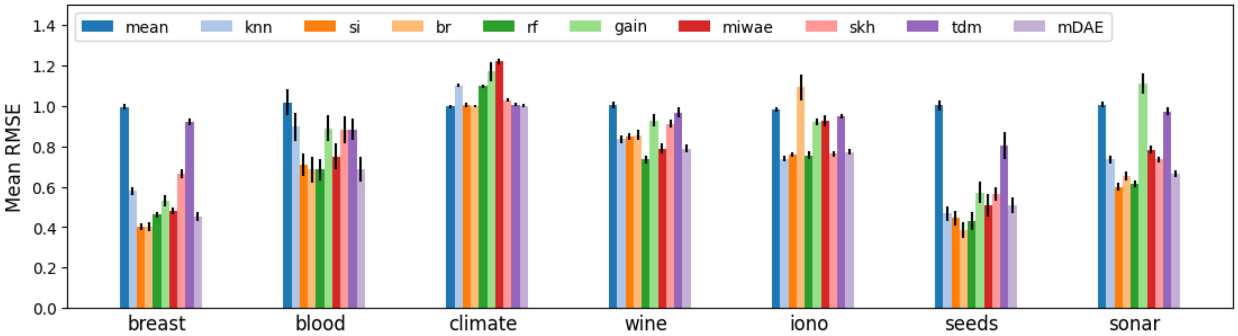


Figure 6: Mean RMSE of reconstruction (± the standard deviation) for $B = 12$ random draws of 20% of MCAR artificial missing values.

where R_{ij} is the RMSE obtained with the method j on the dataset i. $MDB(j)$ interprets as the mean (over the datasets) of the distances between the RMSE of the method j and the RMSE of the best method. It is equal to 0 if the method j is the best for all datasets. It increases if the quality of the method j is far from the quality of the best method, on average over the datasets.

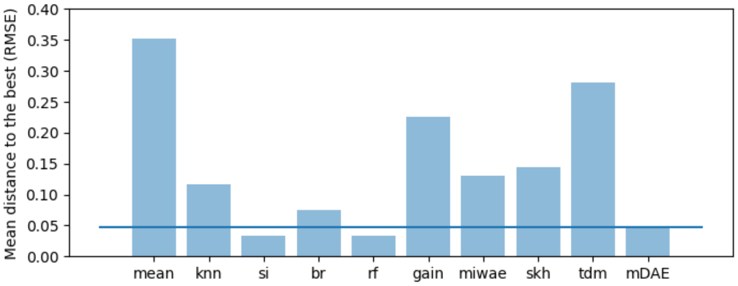


Figure 7: Mean Distance to the Best (MDB) obtained with 20% of MCAR artificial missing values.

Figure 7 shows the MDB obtained with 20% of artificial MCAR missing values and the quality (the RMSE) of the methods plotted Figure 6 . This figure shows that the two best methods according to this criterion are SoftImput (si) and missForest (rf). The mDAE method is the 3rd best method. The Figures 8, 9 and 10 in Appendix B show the results with 40% of artificial MCAR missing values, and with 20% or 40% of MAR and MNAR missing values. With these different proportions and types of missing data, the top three remains SoftImput, missForest and mDAE. SofImpute is always in first place, tied once (40% MAR) with mDAE. The mDAE and missForest methods are generally one or the other in second and third position.

These results confirm the good performance of the classic imputation methods (SoftImput and missForest) compared to more recent methods based on neural networks and optimal transport (gain, miwae, skh and tdm), and the good performance (comparable to missForest) of the mDAE method based on Denoising AutoEncoders.

4 Conclusion

This article proposes a method for missing data imputation, based on DAE, as well as a procedure for choosing the hyper-parameters (the proportion of noise μ and the structure of the network). An ablation study of this method was performed with different datasets, different types and proportions of missing data. It showed the relatively small improvement of the results when the hyper-parameter μ is chosen by cross-validation rather than randomly. On the contrary, using an overcomplete rather than an undercomplete network seems appropriate. A specific study is still required to confirm this result, which would enable to recommend the use of a random μ and an overcomplete structure.

Then, a numerical study compared the proposed mDAE method with eight other standard or recent missing values imputation methods. The results showed the good behavior of SofImput, mDAE and missForest. A new criterion called Mean Distance to the Best (MDB) was used to compare the methods globally over all the considered datasets and to rank them. The four most recent methods based on deep learning and optimal transport were systematically found in the last four positions for all types and proportions of artificial missing values. One might think these methods give better results with image or natural language processing data. This should be tested more thoroughly. The Python code for this numerical comparison will be made available on GitHub so that it can be reproduced with other datasets or completed with other methods.

Finally, the specific features of the mDAE method should make it possible to consider block-wise missing values by imposing a block-wise structuring of the masking noise. This type of missing data is frequent, for instance, with Electronic health records, longitudinal studies or time series data, where failures in sensors and communication can result in a loss of multiple consecutive data points.

References

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2 (1):1–127, 2009.

Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Yanjie Duan, Yisheng Lv, Wenwen Kang, and Yifei Zhao. A deep learning based approach for traffic data imputation. In *17th International IEEE conference on intelligent transportation systems (ITSC)*, pp. 912–917. IEEE, 2014.

Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*, pp. 260–272. Springer, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

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

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

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

Imke Mayer, Aude Sportisse, Julie Josse, Nicholas Tierney, and Nathalie Vialaneix. R-miss-tastic: a unified platform for missing values methods and workflows, 2021.

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.

Boris Muzellec. MissingDataOT. URL `https://github.com/BorisMuzellec/MissingDataOT`.

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, pp. 7130–7140. PMLR, 2020.

Ignacio Peis, Chao Ma, and José Miguel Hernández-Lobato. Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo. *Advances in Neural Information Processing Systems*, 35:35839–35851, 2022.

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.

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

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

Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17 (6):520–525, 2001.

Stef Van Buuren. *Flexible imputation of missing data.* CRC press, 2018.

Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pp. 5689–5698. PMLR, 2018.

He Zhao, Ke Sun, Amir Dezfouli, and Edwin V Bonilla. Transformed distribution matching for missing value imputation. In *International Conference on Machine Learning*, pp. 42159–42186. PMLR, 2023.

A Appendix

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	**0.535 ± 0.012**	**1.005 ± 0.009**	**0.735 ± 0.011**	**0.793 ± 0.012**	**0.537 ± 0.026**	**0.862 ± 0.029**	0.761 ± 0.053
mDAE w/o modified loss	0.829 ± 0.021 (54.953%)	1.006 ± 0.008 (0.100%)	1.007 ± 0.007 (37.007%)	0.880 ± 0.013 (10.971%)	0.741 ± 0.029 (37.989%)	0.927 ± 0.027 (7.541%)	0.844 ± 0.052 (10.907%)
mDAE w/o optimal μ	0.538 ± 0.028 (0.561%)	1.023 ± 0.018 (1.791%)	0.764 ± 0.041 (3.946%)	0.832 ± 0.035 (4.918%)	0.563 ± 0.052 (4.842%)	0.885 ± 0.055 (2.668%)	**0.746 ± 0.054** (-1.971%)
mDAE w/o overcomplete	0.548 ± 0.014 (2.430%)	1.159 ± 0.014 (15.323%)	0.774 ± 0.013 (5.306%)	0.845 ± 0.016 (6.557%)	0.756 ± 0.203 (40.782%)	0.959 ± 0.028 (11.253%)	0.849 ± 0.106 (11.564%)

Table 4: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 40% of MCAR artificial missing values.

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	0.484 ± 0.039	1.009 ± 0.012	**0.657 ± 0.033**	**0.834 ± 0.029**	**0.468 ± 0.057**	**0.829 ± 0.042**	**0.613 ± 0.187**
mDAE w/o modified loss	0.812 ± 0.066 (67.769%)	**1.005 ± 0.011** (-0.396%)	0.978 ± 0.026 (48.858%)	0.898 ± 0.028 (7.674%)	0.682 ± 0.088 (45.726%)	0.892 ± 0.061 (7.600%)	0.839 ± 0.299 (36.868%)
mDAE w/o optimal μ	**0.482 ± 0.038** (-0.413%)	1.033 ± 0.015 (2.379%)	0.686 ± 0.042 (4.414%)	0.880 ± 0.055 (5.516%)	0.485 ± 0.071 (3.632%)	0.888 ± 0.090 (7.117%)	0.637 ± 0.225 (3.915%)
mDAE w/o overcomplete	0.521 ± 0.035 (7.645%)	1.161 ± 0.013 (15.064%)	0.716 ± 0.030 (8.980%)	0.899 ± 0.046 (7.794%)	0.830 ± 0.294 (77.350%)	0.974 ± 0.065 (17.491%)	0.967 ± 0.345 (57.749%)

Table 5: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 20% of MAR artificial missing values.

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	**0.510 ± 0.032**	**1.005 ± 0.006**	**0.718 ± 0.017**	**0.804 ± 0.018**	**0.511 ± 0.040**	**0.830 ± 0.021**	**0.658 ± 0.090**
mDAE w/o modified loss	0.854 ± 0.045 (67.451%)	1.005 ± 0.008 (0.000%)	1.000 ± 0.024 (39.276%)	0.891 ± 0.025 (10.821%)	0.821 ± 0.052 (60.665%)	0.916 ± 0.027 (10.361%)	0.893 ± 0.155 (35.714%)
mDAE w/o optimal μ	0.546 ± 0.059 (7.059%)	1.033 ± 0.021 (2.786%)	0.766 ± 0.033 (6.685%)	0.846 ± 0.046 (5.224%)	0.537 ± 0.052 (5.088%)	0.925 ± 0.047 (11.446%)	0.668 ± 0.099 (1.520%)
mDAE w/o overcomplete	0.526 ± 0.023 (3.137%)	1.149 ± 0.015 (14.328%)	0.780 ± 0.024 (8.635%)	0.868 ± 0.028 (7.960%)	0.743 ± 0.230 (45.401%)	1.018 ± 0.140 (22.651%)	0.989 ± 0.232 (50.304%)

Table 6: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 40% of MAR artificial missing values.

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	**0.486 ± 0.029**	1.001 ± 0.006	**0.684 ± 0.013**	**0.829 ± 0.027**	**0.503 ± 0.032**	**0.805 ± 0.033**	**0.738 ± 0.176**
mDAE w/o modified loss	0.795 ± 0.048 (63.580%)	**1.000 ± 0.008** (-0.100%)	1.005 ± 0.019 (46.930%)	0.890 ± 0.033 (7.358%)	0.682 ± 0.084 (35.586%)	0.864 ± 0.043 (7.329%)	0.943 ± 0.248 (27.778%)
mDAE w/o optimal μ	0.518 ± 0.050 (6.584%)	1.024 ± 0.011 (2.298%)	0.698 ± 0.030 (2.047%)	0.836 ± 0.021 (0.844%)	0.531 ± 0.025 (5.567%)	0.839 ± 0.076 (4.224%)	0.772 ± 0.213 (4.607%)
mDAE w/o overcomplete	0.521 ± 0.025 (7.202%)	1.156 ± 0.016 (15.485%)	0.742 ± 0.028 (8.480%)	0.893 ± 0.055 (7.720%)	0.686 ± 0.229 (36.382%)	0.965 ± 0.091 (19.876%)	0.950 ± 0.288 (28.726%)

Table 7: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 20% of MNAR artificial missing values.

Method	**breast**	**climate**	**sonar**	**iono**	**seeds**	**wine**	**blood**
mDAE	**0.543 ± 0.030**	**1.005 ± 0.006**	**0.738 ± 0.018**	**0.798 ± 0.018**	**0.524 ± 0.031**	**0.873 ± 0.042**	**0.737 ± 0.102**
mDAE w/o modified loss	0.866 ± 0.043 (59.484%)	1.007 ± 0.007 (0.199%)	0.998 ± 0.016 (35.230%)	0.893 ± 0.011 (11.905%)	0.800 ± 0.039 (52.672%)	0.950 ± 0.047 (8.820%)	0.885 ± 0.109 (20.081%)
mDAE w/o optimal μ	0.574 ± 0.048 (5.709%)	1.016 ± 0.012 (1.095%)	0.750 ± 0.039 (1.626%)	0.830 ± 0.035 (4.010%)	0.565 ± 0.051 (7.824%)	0.922 ± 0.039 (5.613%)	0.750 ± 0.115 (1.764%)
mDAE w/o overcomplete	0.559 ± 0.024 (2.947%)	1.158 ± 0.013 (15.224%)	0.796 ± 0.025 (7.859%)	0.859 ± 0.020 (7.644%)	0.827 ± 0.236 (57.824%)	1.000 ± 0.034 (14.548%)	0.927 ± 0.082 (25.780%)

Table 8: Mean RMSE of reconstruction (± the standard deviation) for $B = 8$ random draws of 40% of MNAR artificial missing values.

B Appendix

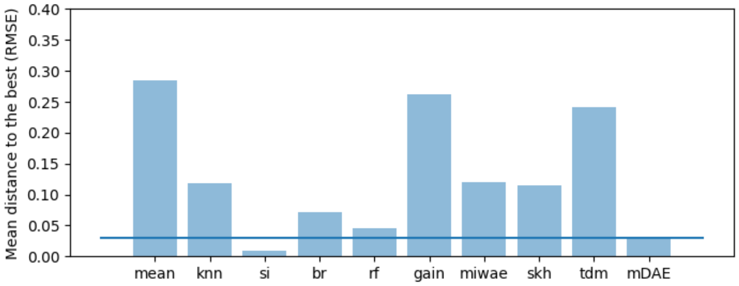
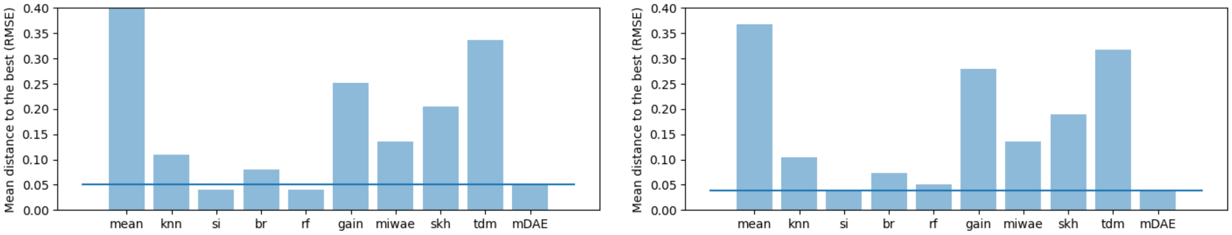


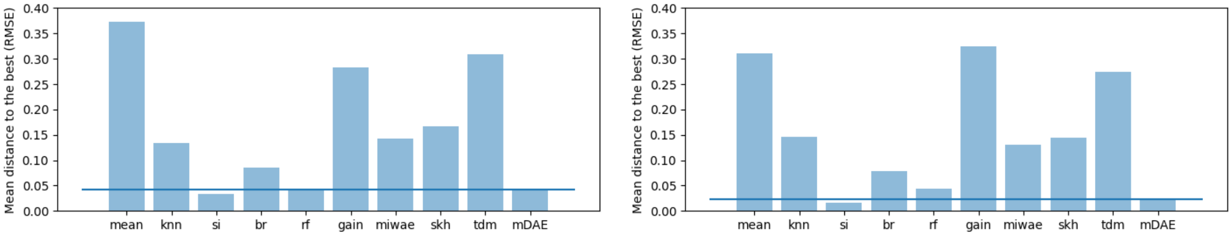
Figure 8: Mean Distance to the Best (MDB) obtained with 40% of MCAR artificial missing values.



(a) MAR 20%

(b) MAR 40%

Figure 9: Mean Distance to the Best (MDB) obtained with MAR artificial missing values.



(a) MNAR 20%

(b) MNAR 40%

Figure 10: Mean Distance to the Best (MDB) obtained with MNAR artificial missing values.