Journal of Artificial Intelligence Research 69 (2020) 1255-1285 Submitted 07/2020; published 12/2020

Reviewing Autoencoders for Missing Data Imputation: Technical Trends, Applications and Outcomes

Ricardo Cardoso Pereira RDPEREIRA@DEI.UC.PT
Centre for Informatics and Systems of the University of Coimbra (CISUC), Department of Informatics Engineering, University of Coimbra, 3030-790, Coimbra, Portugal

Miriam Seoane Santos MIRIAMS@DEI.UC.PT
IPO-Porto Research Centre, 4200-072, Porto, Portugal
Centre for Informatics and Systems of the University of Coimbra (CISUC), Department of Informatics Engineering, University of Coimbra, 3030-790, Coimbra, Portugal

Pedro Pereira Rodrigues PPRODRIGUES@MED.UP.PT
Center for Health Technology and Services Research (CINTESIS), Faculty of Medicine (MEDCIDS-FMUP), University of Porto, 4200-319, Porto, Portugal

Pedro Henriques Abreu PHA@DEI.UC.PT
Centre for Informatics and Systems of the University of Coimbra (CISUC), Department of Informatics Engineering, University of Coimbra, 3030-790, Coimbra, Portugal

Abstract

Missing data is a problem often found in real-world datasets and it can degrade the performance of most machine learning models. Several deep learning techniques have been used to address this issue, and one of them is the Autoencoder and its Denoising and Variational variants. These models are able to learn a representation of the data with missing values and generate plausible new ones to replace them. This study surveys the use of Autoencoders for the imputation of tabular data and considers 26 works published between 2014 and 2020. The analysis is mainly focused on discussing patterns and recommendations for the architecture, hyperparameters and training settings of the network, while providing a detailed discussion of the results obtained by Autoencoders when compared to other state-of-the-art methods, and of the data contexts where they have been applied. The conclusions include a set of recommendations for the technical settings of the network, and show that Denoising Autoencoders outperform their competitors, particularly the often used statistical methods.

1. Introduction

Missing data is a common problem that appears in real-world contexts and may compromise the performance of most learning models (Abreu et al., 2014a,b). In the research community, three missing mechanisms are recognized (Baraldi and Enders, 2010; Rubin, 1976; Little and Rubin, 2019):

- Missing Completely At Random (MCAR) - occurs when the mechanism under the missingness is unrelated to any observed or unobserved values from the dataset;

- Missing At Random (MAR) - occurs when the cause of the missing data is related with observed values from the dataset;
- Missing Not At Random (MNAR) - occurs when the probability of a value being missing is related with that same value and/or with other unknown data.

The existent literature identifies several ways to handle these mechanisms (see Figure 1): case deletion, imputation methods, Maximum Likelihood and machine learning without missing data estimation (García-Laencina et al., 2009; Graham et al., 2003).

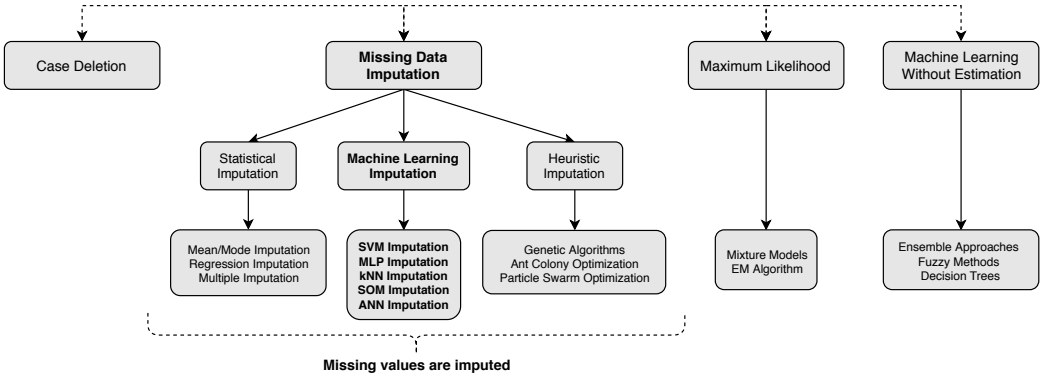

Figure 1: Methods to handle missing values (adapted from García-Laencina et al., 2009).

Each of these approaches presents advantages and limitations, but the one most frequently used is data imputation (García-Laencina et al., 2015; Santos et al., 2015, 2017). In this approach, plausible values are generated to replace the missing ones, and it is mainly divided into statistical-based and machine learning-based methods (García-Laencina et al., 2009). Statistical methods consist in replacing the missing observations with the most similar ones among the training data, without the need of constructing a predictive model to evaluate their "similarity" (e.g. mean/mode imputation). Machine learning-based techniques construct a predictive model with the available data to estimate values to replace those that are missing (e.g. Support Vector Machine imputation).

Although deep learning has received much attention and has been applied to several artificial intelligence problems over the past couple of years, its investigation 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 (Nelwamondo et al., 2007; Sánchez-Morales et al., 2019; Costa et al., 2018; Gondara and Wang, 2018). 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 survey 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 in the following section). 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. To the best of our knowledge, no other studies exist in the literature that address this topic, especially with such a wide technical and comparative analysis, which makes this survey a novelty in the missing data field. The analysis conducted in this survey 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 this survey's 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 survey 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 5.

The remainder of the paper is organized in the following way: Section 2 presents the theoretical background and technical details related to the hyperparameters and training of the AEs; Section 3 a characterization of the datasets used in the works; Section 4 the comparison with other methods regarding imputation and classification/regression; Section 5 an analysis of the works that use non-tabular data; and Section 6 the conclusions with a discussion of the results, recommendations and challenges found throughout the survey.

2. 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.

2.1 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 2).

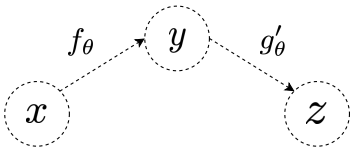

Figure 2: Simplified structure of an Autoencoder. f represents the encoder and g' the decoder. x is the input of the network and z is the output. y represents the results of the encoding process.

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 (Charte et al., 2018). 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) (Vincent et al., 2008). 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 (Charte et al., 2018) (Figure 3).

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, Srivastava et al., 2014). There are other possible corruption processes, such as adding Gaussian noise or salt-and-pepper noise to the input data (Vincent et al., 2010).

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

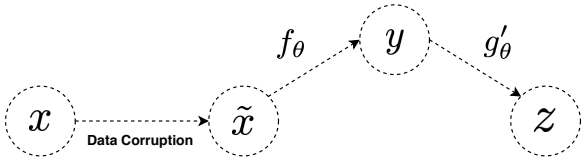

Figure 3: Simplified structure of a Denoising Autoencoder. f represents the encoder and g' the decoder. x is the uncorrupted version of the data, $\widetilde{x}$ is the corrupted input of the network and z is the output. y represents the results of the encoding process.

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 (Kingma and Welling, 2013). The VAE loss function contains two terms: the reconstruction error (as the vanilla AE) and a regularizer (see Equation 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 (Kingma and Welling, 2013).

$$L(X) = -E_{z\sim Q(z|X)}[logp(X|z)] + KL(q(z|X) \parallel p(z)) \tag{1}$$

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 (Makhzani et al., 2015).

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.

2.2 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 4 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. (2016), 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. (2015), where a hyperparameter optimization was conducted using the Hyperopt[1] library.

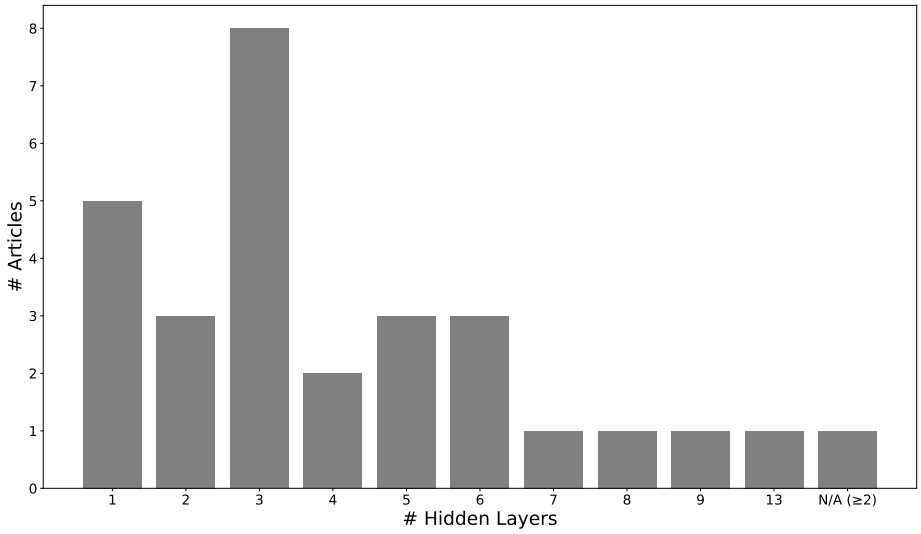

Figure 4: Number of hidden layers used in each work. N/A stands for Not Available.

Regarding the number of nodes, only 15 of the 26 works under analysis describe their type of representation, as Figure 5 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 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 (Hau et al., 2016; Sakurai et al., 2017; Jia et al., 2017), but in the work of Sakurai et al. (2017) a custom linear function is used on the output layer. For multi-layer networks the Sigmoid is also used in some works (Duan et al., 2014, 2016; Xie et al., 2019; Sánchez-Morales et al., 2020), but ReLU is the one more often applied (Gondara and Wang, 2017; Ryu et al., 2020; McCoy et al., 2018; Boquet et al., 2019, 2020; Xie et al., 2019; Saeed et al., 2018; Fortuin et al., 2020), sometimes through the Leaky ReLU variant (Ryu et al., 2020; Saeed et al.,

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

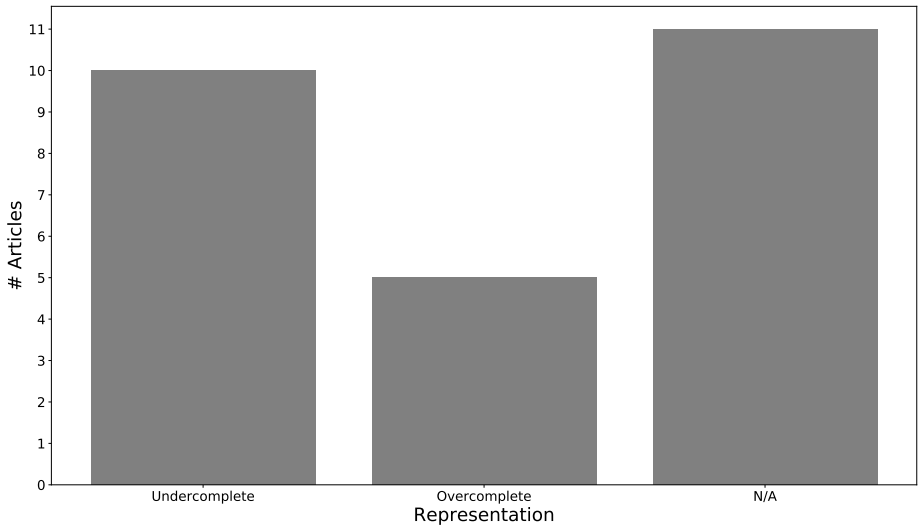

Figure 5: Representation used in each work. N/A stands for Not Available.

2018). The Hyperbolic Tangent is used less often (Gondara and Wang, 2018; Chen et al., 2015), 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. (2017).

2.3 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 (Sánchez-Morales et al., 2017; Hau et al., 2016; Sánchez-Morales et al., 2019; Xie et al., 2019; Lai et al., 2019) and 6 use one of its variants called Adam (El Esawey et al., 2015; Ryu et al., 2020; Boquet et al., 2019, 2020; Saeed et al., 2018; Fortuin et al., 2020). Other algorithms are used less often, namely the Nesterov's Accelerated Gradient (Gondara and Wang, 2018), the Scaled Conjugate Gradient Algorithm (Sakurai et al., 2017) and the RMSProp (McCoy et al., 2018).

Regarding the loss functions, the majority of the works use the standard Mean Squared Error (MSE) (Gondara and Wang, 2017, 2018; El Esawey et al., 2015; Sakurai et al., 2017; Jia et al., 2017; Boquet et al., 2019; Saeed et al., 2018) or the Squared Error (Hau et al., 2016; Chen et al., 2015; Sánchez-Morales et al., 2019; Ryu et al., 2020). The exceptions are a work that uses a custom loss function roughly based on the Squared Error (Zhao et al., 2016), 3 works that use the Cross-Entropy (Jaques et al., 2017; Lee and Lee, 2017; Sánchez-Morales et al., 2020), another one that uses the Binary Cross-Entropy with a modification to handle the missing values (Beaulieu-Jones and Moore, 2017), and finally 4 VAE-based works which use the log-likelihood (McCoy et al., 2018; Xie et al., 2019; Nazabal et al., 2020; Fortuin et al., 2020). Notice that all VAE-based works also include the Kullback–Leibler divergence as a second term of the loss function.

Table 1: Nodes used in each work, grouped by organizational approach (see Section 2.2 for more details). N/A stands for Not Available.

Approach	Article	Nodes	Representation
Symmetric	Gondara and Wang (2017)	Step of 5 over 4 layers	Overcomplete
	Gondara and Wang (2018)	Step of 7 over 5 layers	Overcomplete
	Sánchez-Morales et al. (2017)	25% to 75% growth over 3 layers	Overcomplete
	Sánchez-Morales et al. (2019)	75% growth over 3 layers	Overcomplete
	Duan et al. (2014)	Step of 72 over 3 layers	Undercomplete
	Duan et al. (2016)	Step of 72 over 3 layers	Undercomplete
	Jaques et al. (2017)	Step of 200 over 4 layers	Undercomplete
	Ryu et al. (2020)	50% growth; latent space of 90	Undercomplete
	Sánchez-Morales et al. (2020)	75% growth over 3 layers	Overcomplete
Constant	McCoy et al. (2018)	Constant 20; latent space of 10	Undercomplete
	Boquet et al. (2019)	Constant 512; latent space of 100	Undercomplete
	Beaulieu-Jones and Moore (2017)	Constant 500	Undercomplete
	Saeed et al. (2018)	Constant 128	Undercomplete
Empirical	Xie et al. (2019)	Between 3 and 13 over 7 layers	Undercomplete
	Xie et al. (2019)	Between 3 and 10 over 5 layers	Undercomplete
	Chen et al. (2015)	Between 5 and 36 over 6 layers	Undercomplete
	Fortuin et al. (2020)	Variations over 4 and 6 layers	Undercomplete
	Sakurai et al. (2017)	Between 4 and 70	Single-layer
	Hau et al. (2016)	Between 200 and 400	Single-layer
	Lee and Lee (2017)	Between 200 and 400	Single-layer
Optimization	El Esawey et al. (2015)	Between 240 and 370 over 5 layers	N/A
	Lai et al. (2019)	Between 5 and 30 over 1 or 2 layers	N/A

With the exception of the modified Binary Cross-Entropy (Beaulieu-Jones and Moore, 2017), 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 (2018) used the mean/mode method; Sánchez-Morales et al. (2017) 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. (2017) compared constant and linear imputation, the

latter performing better in general; Jaques et al. (2017) and Saeed et al. (2018) imputed all missing values with -1; Ryu et al. (2020) replaced all missing values with 0; Sánchez-Morales et al. (2020) 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. (2019) 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 (Saeed et al., 2018; Fortuin et al., 2020), 100 (Beaulieu-Jones and Moore, 2017), 300 (Hau et al., 2016), 500 (Gondara and Wang, 2018), 1000 (Gondara and Wang, 2017; Sakurai et al., 2017), 2000 (Nazabal et al., 2020) and 10000 (McCoy et al., 2018; Lai et al., 2019). Although early stopping rules are common on ANNs training, they were only applied by Gondara and Wang (2018), 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 overcomplete 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 (Schmidhuber, 2015). 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 (Gondara and Wang, 2018; Zhao et al., 2016; Jaques et al., 2017; Hau et al., 2016; Boquet et al., 2019, 2020; Saeed et al., 2018) with the coefficient values varying between 0.01 and 0.001. Gondara and Wang (2017) 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 (2017) and Saeed et al. (2018). Nazabal et al. (2020) 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 (Gondara and Wang, 2017, 2018; Jaques et al., 2017).

A common procedure applied before the training step is to normalize the input data (Abreu et al., 2016). 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 $[0, 1]$ (Gondara and Wang, 2018; Jaques et al., 2017; Saeed et al., 2018; Boquet et al., 2019, 2020; Sánchez-Morales et al., 2020) and $[-1, 1]$ with zero mean (Chen et al., 2015). Xie et al. (2019) 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% (Jaques et al., 2017), 20% (Gondara and Wang, 2017; Beaulieu-Jones and Moore, 2017), 30% (El Esawey et al., 2015) (although this work shows that using values between 10% and 60% produces equally good results) and, finally, 50% (Gondara and Wang, 2018). The work from Sánchez-Morales et al. (2020) 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 (Sánchez-Morales et al., 2017; Duan et al., 2014, 2016; Ning et al., 2017; El Esawey et al., 2015; Zhao et al., 2016; Sánchez-Morales et al., 2020). 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. (2010), and is called Stacked Autoencoder.

2.4 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 survey.

Gondara and Wang (2018) 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. (2016) 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. (2015) 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. (2017) 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. (2017) 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. (2020) 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. (2016) 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. (2017) 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. (2018) proposed a similar approach for the same purpose.

Chen et al. (2015) 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. (2019) 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 (2017) 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. (2016) introduced a similar but simpler approach, where an AE is trained and used without changes for recommendation purposes.

Nazabal et al. (2020) 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. (2020) 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. (2019) 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. Datasets Characterization

All the 26 analyzed works describe the datasets used in the experiments, as Table 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 2: Datasets used in each work. N/A stands for Not Available.

Article(s)	**Public**	**Dataset(s)**	**# Instances**	**# Features**	**Missing Rates (%)**
Gondara and Wang (2017)	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
Gondara and Wang (2018)	Yes	15 real-world	101 to 58000 (the majority are small-sized)	5 to 180	20
Beaulieu-Jones and Moore (2017)	Yes	ALS Pooled Resource Open-Access Clinical Trials (PRO-ACT)	10723 (but only 1824 are used)	23	10, 20, 30, 40, 50
Duan et al. (2014)	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 (%)
Duan et al. (2016)	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
Sánchez-Morales et al. (2017)	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
El Esawey et al. (2015)	Yes	City of Vancouver data of bicycle traffic, for 22 counter locations over 3 years	12986	2	N/A
Zhao et al. (2016)	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
Hau et al. (2016)	Yes	Real-world Web Service QoS	1974675	2	N/A
Sakurai et al. (2017)	No	Real time series showcase data of one month	N/A	N/A	N/A
Jia et al. (2017)	Yes	EEG data extracted from UCI	600	16384	30
Jaques et al. (2017)	No	Mood data from SNAPSHOT project	6180	343	30
Chen et al. (2015)	Yes	CMU Graphics Lab Motion Capture Database	5	50	N/A
Ning et al. (2017)	No	Quality inspection data collected from social networks and e-commerce websites	N/A	N/A	1, 5, 10, 15, 20, 25, 30
Sánchez-Morales et al. (2019)	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
Lee and Lee (2017)	Yes	MovieLens (ML-100K and ML-1M)	100000 and 1M	N/A	N/A
Ryu et al. (2020)	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
McCoy et al. (2018)	No	Simulated Milling Circuit	104	96	20, 90

Article(s)	Public	Dataset(s)	# Instances	# Features	Missing Rates (%)
Boquet et al. (2019)	Yes	Caltrans Performance Measurement System (PeMS)	210432	288	10, 20, 40
Boquet et al. (2020)	Yes	Caltrans Performance Measurement System (PeMS)	210432	288	10, 20, 40
Xie et al. (2019)	No	Industrial data collected from the Distributed Control System of a polyester plant in China	10000	15	10, 30, 50
Saeed et al. (2018)	Yes	ExtraSensory	$\approx$ 300000	166	5
Nazabal et al. (2020)	Yes	6 academic from UCI (Adult, Breast, Default-Credit, Letter, Spam and Wine)	699 to 32561	10 to 58	10, 50
Fortuin et al. (2020)	Yes	2012 Physionet Challenge real-world medical data	4000	37	60
Sánchez-Morales et al. (2020)	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
Lai et al. (2019)	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 (Beaulieu-Jones and Moore, 2017; Jia et al., 2017; Jaques et al., 2017; Chen et al., 2015; Fortuin et al., 2020; Saeed et al., 2018), quality of service (Ning et al., 2017; Zhao et al., 2016; Hau et al., 2016; Sakurai et al., 2017), traffic data (Duan et al., 2014, 2016; El Esawey et al., 2015; Boquet et al., 2019, 2020) and automation systems (Xie et al., 2019; McCoy et al., 2018; Ryu et al., 2020). 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 Gondara and Wang (2017, 2018); Beaulieu-Jones and Moore (2017); Boquet et al. (2019, 2020) address the MCAR and MNAR mechanisms, while the papers from Duan

et al. (2014); Sánchez-Morales et al. (2017, 2019); McCoy et al. (2018) address only MCAR. The MAR mechanism is not stated by any of the works.

4. 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 6 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 7.

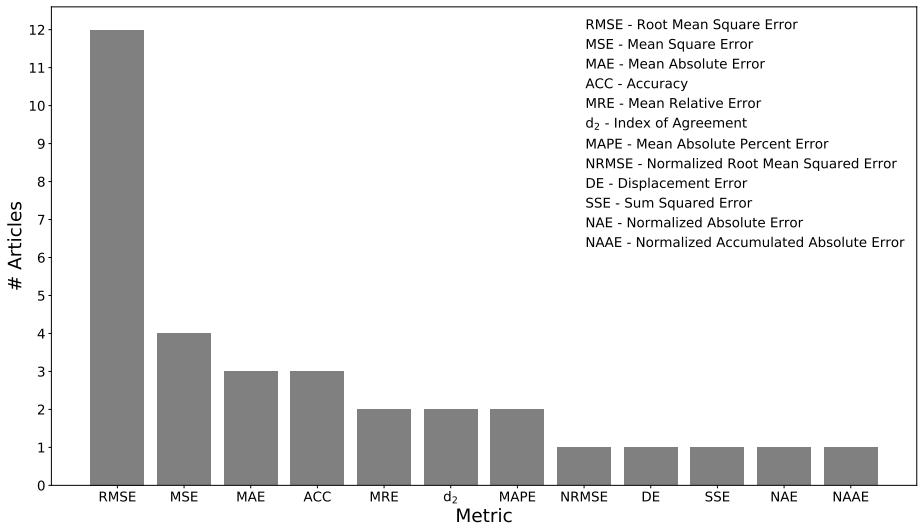

Figure 6: Metrics used for the evaluation of imputation tasks.

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% (Jaques et al., 2017), 66.666%-33.333% (Duan et al., 2014), 70%-30% (Gondara and Wang, 2018; El Esawey et al., 2015; Sánchez-Morales et al., 2020), 80%-20% (Sánchez-Morales et al., 2017, 2019; Xie et al., 2019; Lai et al., 2019) and 90%-10% (Sakurai et al., 2017), for training and test partitions, respectively. The works from Jia et al. (2017) and Saeed et al. (2018) 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 (Sakurai et al., 2017; Jia et al., 2017).

4.1 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.

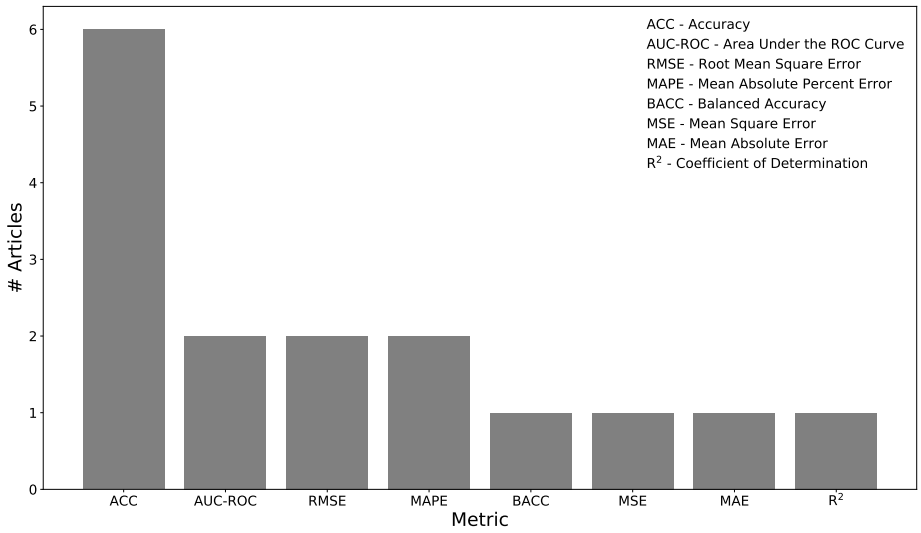

Figure 7: Metrics used for the evaluation of classification and regression tasks.

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.

4.1.1 Denoising Autoencoders

Gondara and Wang (2017) 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 (2018) 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) (Santos et al., 2019). 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 (2017) 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. (2014) 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 (Duan et al., 2016) 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. (2017) 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. (2017) 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. (2017) 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. (2019) compared the DAE and its variant with deletion and compensation 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. (2020) 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. The following families of algorithms were considered: Statistical, Matrix Completion, Distance and Connectionist.

Table 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 (Gondara and Wang, 2017)	4
	Multiple Imputation by Chained Equations (Gondara and Wang, 2018)	3
	Multiple Imputation by Chained Equations (Sánchez-Morales et al., 2019)	2
	Mean/Median imputation (Beaulieu-Jones and Moore, 2017)	4
	Autoregressive Integrated Moving-Average (Duan et al., 2016)	4
	Principal Component Analysis (Jaques et al., 2017)	4
	Linear Interpolation (Ryu et al., 2020)	3
	Historical Average (Ryu et al., 2020)	4
Matrix Completion	Iterative Singular Value Decomposition (Beaulieu-Jones and Moore, 2017)	3
	Singular Value Decomposition (Sánchez-Morales et al., 2019)	3
	SoftImpute (Beaulieu-Jones and Moore, 2017)	3
Distance	k-Nearest Neighbours (Beaulieu-Jones and Moore, 2017)	3
	History Model (Duan et al., 2016)	4
	Weighted kNNs based on Grey Correlation Analysis (Ning et al., 2017)	3
Connectionist	Artificial Neural Networks (Duan et al., 2014)	3
	Multilayer Perceptron Imputation (Sánchez-Morales et al., 2019)	3
	Artificial Neural Networks (Duan et al., 2016)	4

4.1.2 Variational Autoencoders

McCoy et al. (2018) 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. (2018) compared a Adversarial Autoencoder (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. (2020) 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 4. The families of algorithms considered for this variant were: Statistical and Connectionist.

Table 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 (McCoy et al., 2018)	4
	Mean Imputation (McCoy et al., 2018)	4
	Principal Component Analysis (Saeed et al., 2018)	4
	Multiple Imputation by Chained Equations (Nazabal et al., 2020)	3
	Mean Imputation (Nazabal et al., 2020)	4
	General Latent Feature Model (Nazabal et al., 2020)	2
Connectionist	Generative Adversarial Network (Nazabal et al., 2020)	3

4.2 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 (2018) 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. (2017) 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. (2017) 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. (2018) 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. (2019) 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. (2020) 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. (2020) 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. (2020), 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. (2019) trained a 2-layer MLP with data imputed through a VAE, a vanilla AE 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. (2020) extended the latter study and presented the same results and conclusions.

Xie et al. (2019) 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.

5. Autoencoders for Non-Tabular Data

As previously stated, this survey 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 survey. 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. (2015) 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. (2018) 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. (2017) 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. (2017) followed an approach similar to Shang et al. (2017) 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. (2019) 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 5. Moreover, a summary of the results obtained from the 3 works that evaluate the imputation quality is presented in Table 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 5: Image datasets used in each work. N/A stands for Not Available.

Article(s)	**Public**	**Dataset(s)**	**# Instances**	**# Features**	**Missing Rates (%)**
Shao et al. (2015)	Yes	BUAA and Oulu-CASIA images databases	15 and 80 (respectively)	2 image modalities (NIR and VIS)	N/A
Tran et al. (2017)	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
Malek et al. (2018)	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
Shang et al. (2017)	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
Ma et al. (2019)	Yes	MNIST	70000	784	1 to 70

Table 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 (Tran et al., 2017)	4
	SoftImpute (Tran et al., 2017)	4
	OptSpace (Tran et al., 2017)	4
Evolutionary$^{\triangle\square}$	Genetic Algorithms (GA) (Tran et al., 2017)	4
	Genetic Algorithms (Malek et al., 2018)	4
Connectionist$^{\triangle\|}$	Denoising Autoencoder (Tran et al., 2017)	4
	Stacked Denoising Autoencoder (Tran et al., 2017)	4
	Multi-modal Autoencoder (Tran et al., 2017)	4
	Deep Canonically Correlated Autoencoders (Tran et al., 2017)	3
	Variational Autoencoder (Ma et al., 2019)	4
Other$^{\square}$	Orthogonal Matching Pursuit (Malek et al., 2018)	4
	Basis Pursuit (Malek et al., 2018)	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.

6. Conclusions, Recommendations and Open Challenges

This survey 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 that should be used. To the best of the authors knowledge, this is the first work that performs this type of analysis.

6.1 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 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 survey are mostly shared with these extensions.

6.2 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.

6.3 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 Sánchez-Morales et al., 2019, 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. (2020) (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).

6.4 Open Challenges

The works under analysis in this survey 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 (Pereira et al., 2020) (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 paper mostly provides an analysis over tabular data, the results presented in Section 5 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.

Acknowledgments

This work was supported in part by the Portuguese Foundation for Science and Technology (FCT) Research Grants SFRH/BD/149018/2019 and SFRH/BD/138749/2018.

References

Abreu, P. H., Amaro, H., Silva, D. C., Machado, P., and Abreu, M. H. (2014a). Personalizing breast cancer patients with heterogeneous data. In *The International Conference on Health Informatics*, pages 39–42.

Abreu, P. H., Amaro, H., Silva, D. C., Machado, P., Abreu, M. H., Afonso, N., and Dourado, A. (2014b). Overall survival prediction for women breast cancer using ensemble methods and incomplete clinical data. In *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*, pages 1366–1369.

Abreu, P. H., Santos, M. S., Abreu, M. H., Andrade, B., and Silva, D. C. (2016). Predicting breast cancer recurrence using machine learning techniques: a systematic review. *ACM Computing Surveys (CSUR)*, 49:52.

Baraldi, A. N. and Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of School Psychology*, 48:5–37.

Beaulieu-Jones, B. K. and Moore, J. H. (2017). Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific Symposium on Biocomputing 2017*, pages 207–218.

Boquet, G., Morell, A., Serrano, J., and Vicario, J. L. (2020). 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.

Boquet, G., Vicario, J. L., Morell, A., and Serrano, J. (2019). Missing data in traffic estimation: A variational autoencoder imputation method. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2882–2886.

Charte, D., Charte, F., García, S., del Jesus, M. J., and Herrera, F. (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96.

Chen, N., Bayer, J., Urban, S., and Van Der Smagt, P. (2015). Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 434–440.

Costa, A. F., Santos, M. S., Soares, J. P., and Abreu, P. H. (2018). Missing data imputation via denoising autoencoders: The untold story. In *17th International Symposium on Intelligent Data Analysis*, pages 87–98.

Duan, Y., Lv, Y., Kang, W., and Zhao, Y. (2014). A deep learning based approach for traffic data imputation. In *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, pages 912–917.

Duan, Y., Lv, Y., Liu, Y.-L., and Wang, F.-Y. (2016). An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72:168–181.

El Esawey, M., Mosa, A. I., and Nasr, K. (2015). Estimation of daily bicycle traffic volumes using sparse data. *Computers, Environment and Urban Systems*, 54:195–203.

Fortuin, V., Baranchuk, D., Rätsch, G., and Mandt, S. (2020). Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661.

García-Laencina, P. J., Abreu, P. H., Abreu, M. H., and Afonoso, N. (2015). 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.

García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2009). Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282.

Gondara, L. and Wang, K. (2017). Recovering loss to followup information using denoising autoencoders. In *2017 IEEE International Conference on Big Data*, pages 1936–1945.

Gondara, L. and Wang, K. (2018). Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 260–272.

Graham, J. W., Cumsille, P. E., and Elek-Fisk, E. (2003). Methods for handling missing data. *Handbook of Psychology*, 2:87–114.

Hau, N. X., Tu, T. D., et al. (2016). User-based autoencoder for qos prediction. In *7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 308–311.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2017). 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 (ACII)*, pages 202–208.

Jia, Y., Zhou, C., and Motani, M. (2017). Spatio-temporal autoencoder for feature learning in patient data with missing observations. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 886–890.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lai, X., Wu, X., Zhang, L., Lu, W., and Zhong, C. (2019). Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing*, 366:54–65.

Lee, J.-w. and Lee, J. (2017). Idae: Imputation-boosted denoising autoencoder for collaborative filtering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2143–2146.

Little, R. J. and Rubin, D. B. (2019). *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons.

Ma, C., Tschiatschek, S., Palla, K., Hernandez-Lobato, J. M., Nowozin, S., and Zhang, C. (2019). Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Malek, S., Melgani, F., Bazi, Y., and Alajlan, N. (2018). Reconstructing cloud-contaminated multispectral images with contextualized autoencoder neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 56:2270–2282.

McCoy, J. T., Kroon, S., and Auret, L. (2018). Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51:141–146.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. (2020). Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501.

Nelwamondo, F. V., Mohamed, S., and Marwala, T. (2007). Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pages 1514–1521.

Ning, X., Xu, Y., Gao, X., and Li, Y. (2017). Missing data of quality inspection imputation algorithm base on stacked denoising auto-encoder. In *IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 84–88.

Pereira, R. C., Abreu, P. H., and Rodrigues, P. P. (2020). Vae-bridge: Variational autoencoder filter for bayesian ridge imputation of missing data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

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

Ryu, S., Kim, M., and Kim, H. (2020). Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access*, 8:40656–40666.

Saeed, A., Ozcelebi, T., and Lukkien, J. (2018). Synthesizing and reconstructing missing sensory modalities in behavioral context recognition. *Sensors*, 18:2967.

Sakurai, D., Fukuyama, Y., Santana, A., Murakami, K., and Matsui, T. (2017). Estimation of missing data of showcase using autoencoder. In *56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1303–1307.

Sánchez-Morales, A., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2017). Values deletion to improve deep imputation processes. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 240–246.

Sánchez-Morales, A., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2020). Complete autoencoders for classification with missing values. *Neural Computing and Applications*, pages 1–7.

Sánchez-Morales, A., Sancho-Gómez, J.-L., Martínez-García, J.-A., and Figueiras-Vidal, A. R. (2019). Improving deep learning performance with missing values via deletion and compensation. *Neural Computing and Applications*, 32:13233–13244.

Santos, M. S., Abreu, P. H., García-Laencina, P. J., Simão, A., and Carvalho, A. (2015). A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *Journal of Biomedical Informatics*, 58:49–59.

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

Santos, M. S., Soares, J. P., Abreu, P. H., Araújo, H., and Santos, J. (2017). Influence of data distribution in missing data imputation. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 285–294.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Shang, C., Palmer, A., Sun, J., Chen, K.-S., Lu, J., and Bi, J. (2017). Vigan: Missing view imputation with generative adversarial networks. In *IEEE International Conference on Big Data*, pages 766–775.

Shao, M., Ding, Z., and Fu, Y. (2015). 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 (FG)*, pages 1–6.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Tran, L., Liu, X., Zhou, J., and Jin, R. (2017). Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1405–1414.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408.

Xie, R., Jan, N. M., Hao, K., Chen, L., and Huang, B. (2019). Supervised variational autoencoders for soft sensor modeling with missing data. *IEEE Transactions on Industrial Informatics*, 16:2820–2828.

Zhao, L., Chen, Z., Yang, Z., Hu, Y., and Obaidat, M. S. (2016). Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. *IEEE Systems Journal*, 12:1610–1620.