Clustering with Missing Values: A Deep Learning Approach

Rita Tomás Caveirinha

***Abstract*—This work focuses on the usage of deep clustering models, whose research has been growing in recent years due to their usefulness of its usage across many fields. The goal of this thesis was to work around the problem of clustering missing values using deep clustering, which is a fundamental problem to tackle since in real-world data, such as the profiling of patients by their progression patterns in neurodegenerative diseases, where missing values often occur. Yet, after exploring the state-of-the-art techniques it was concluded that the existence of missing data is an obstacle to the robustness of the best clustering methodologies since they are often developed and tested with clean data. In a similar observation, it was noticeable that the deep learning architectures with the ability to work with missing data failed to solve the clustering part of the problem. With that in mind, this work explored the topic of clustering with missing data to find the most optimal solution, going through different base architectures that could potentially handle missing data, and the most important, introducing a variational autoencoder variation through the use of a binary mask to better handle missing data. With the usage of VAE-based architectures, this work performs clustering in its latent space by using the dimension reduction functionalities from UMAP, followed by automatic cluster detection provided by HDBSCAN.**

The main contributions of this work are: 1) Offer a novel approach through deep clustering with missing data, 2) A simple but effective solution for better robustness when dealing with missing data, offering similar results to complete data, 3) Automatic cluster detection, 4) Data dependant dynamic architectures.

***Index Terms*—Deep learning, Clustering, Missing Data, Variational Autoencoders, UMAP, HDBSCAN, Generative Models, Deep image clustering.**

I. Introduction

This work focuses on the topic of clustering with missing values and the creation of a deep clustering model that is robust to the occurrence of missing data values. The usage of deep clustering models has been increasing and revealing itself to be very useful for the study and analysis of the progress across fields such as medicine, with the in the study of melanoma detection [4][27], or even degenerative diseases such as Alzheimer's or Parkinson's. [42] [12] [36] Other cases can also include forest fire detection, [5], [20], or in the maintainance of automatic systems such as eolic turbines [7], [35].

However, the problem of missing values in the data occurs frequently across these fields, [37], [10], sensors and machines that provide exam results can have failures in sensors, or in some cases imaging data suffer from the occlusion (an artifact partially hides the image), or in cases where the data origin is survey-based, questions often get forgotten or the answers invalid. The most common ways to solve this problem rely on simple solutions such as the imputation of values.

Preprocessing of data is always a necessary step but the existence of missing values can create extra complexity to the problem since it implies a need for extra steps to deal with the problem.

Or in the case where the choice is to ignore the missingness, how much could this missingness affect the goal in mind. Previous work has been done with great results by approaching the problem of deep clustering with the usage of generative models. Yet, most of these advanced deep clustering architectures need complete datasets to achieve good results, which means that when missing data occurs the results are not as strong.

This lack of robustness to missing values was, therefore, the main motivation for the work done, in which it was proposed to explore the creation of a deep clustering model that is generative and robust to this problem.

The main goal was to develop a clustering methodology that can directly tackle datasets with missing values without requiring the application of listwise deletion methods.

This work starts by analyzing some of the literature on the mentioned topics, followed by an overview of the architecture achieved, as well as the ones explored, and finally, an analysis of the results obtained.

II. Related Work

A. Missing Data

The problem of missing data is present across different fields, this phenomenon can be considered Missing Completely At Random (MCAR), Missing At Random (MAR), or Missing Not At Random (MNAR). One of the approaches for this problem is through **Deletion**, where any sample that contains missing data is discarded. **Listwise Deletion**, which is when all the instances with missing data are removed, and **Pairwise Deletion**, where the instance is only removed if the needed variable to the computation is missed (if any other variable is missed but not being used, it is not removed). These techniques are used in the assumption that the data is missed completely at random (MCAR).

Another important approach is **Imputation**, which is the replacement of the missing data for a specific value, this can be applied to continuous variables (numeric) where simple techniques such as mean or median, may be used to replace the empty values, or in the instances of categorical data where it can be both string or numerical values, the replacement with the most frequent value in that category can be performed. If there is a high number of missing values in that second scenario, a special category just for these values can be created. There are also two ways in which imputation can

be performed: Single Imputation and Multiple Imputation. Single Imputation includes techniques such as single value, similarity, and regression imputation where one value at a time is imputed. Multiple Imputation consists of calculating the average of the outcomes across multiple imputed data sets to account for this. All multiple imputation methods follow three important steps: 1) imputation, 2) analysis, and 3) pooling.

Many of the commonly used techniques for the handling of missing data are usually based around performing imputation before the learning and often use the whole data for the imputation of a single value, this involves a higher cost and complexity, especially in situations with high dimensional data and big data sets. The goal was therefore to find an approach using deep learning that could be able to handle the missing data and optimally perform data clustering.

The handling of missing values in a deep learning approach has been developing at a fast pace recently, and there have been some fresh ideas quite relevant to the topic such as:

- **MIWAE**[23] is a technique for the handling of missing data with deep latent variable models. This approach is used when the training set contains missing-at-random data and is based on the importance-weighted autoencoder (IWAE) [6] yet it solves the problem of additional computational overhead due to the missing data. It works by maximizing a potentially tight lower bound of the log-likelihood of the observed data
- **not-MIWAE** [18], from the same authors a similar approach to MIWAE was created recently, this approach has a focus on cases where the missing process is dependent on the missing data since in these cases this needs to be explicitly modeled and taken into account while doing likelihood-based inference.
- **Variational Selective Autoencoder (VSAE)** [16], has a focus on the task of models for multimodal data imputation. This model learns only from partially-observed data and it works by modeling the joint distribution of observed/ unobserved modalities and the imputation mask, which results in a unified model for various downstream tasks including data generation and imputation.
- **Robust Variational Autoencoders (RVAE)** [15], has a focus on outlier detection and it is a deep generative model that learns the joint distribution of the clean data while identifying the outlier cells, and with this allows for their imputation. RVAE learns the probability of each cell being an outlier through the balancing of different likelihood models in the row outlier score, which makes this model a suitable one for detection in mixed-type datasets.
- **Variational deep embedding with recurrence (VADER)** [3] is a method that relies on a Gaussian mixture variational autoencoder framework which was extended to model multivariate time series and directly deals with missing values.

B. Clustering

The approaches above focus on the deep learning of models with missing data, often to provide either imputation or classification, however, they do not approach the clustering of this data.

The clustering of data is a common important task in machine learning, however, the focus of this task when missing data is present has been typically done with classical clustering techniques, mostly through density-based techniques, such as DBSCAN, or subspace clustering techniques, which is a base idea behind HDBSCAN, the method chosen for this work.

With the recent developments of deep learning and with that deep clustering as well, the concern for the problem of missing data is still needed. There are many relevant techniques for deep clustering, which can be divided into:

- AE-based - FFocus on the dimension reduction nature of autoencoders, which is one of the most significant algorithms in unsupervised representation learning, working by learning to efficiently compress data followed but its reconstruction. This type of approach often uses a pre-training scheme in which reconstruction loss is used to initialize parameters before applying/introducing clustering loss. Some relevant examples include DEC, DBC, DCN, DEPICT and N2D [38], [22], [39],[14], [24].
- VAE-based - Considered an Autoencoder architecture, and also with the dimension reduction advantages, but works in a continuous space and considered a generative model. Two examples of this include VaDE and GMVAE [19],[13].
- GAN-based - Generative Adversarial networks, along with VAEs are labeled as a generative type of model. GANs are composed of a system of two neural networks: a generator G, which learns a data distribution and generates samples, and a discriminator that learns to distinguish between a sample that came from the training data and a generated sample from G. The networks are simultaneously trained and compete against each other by engaging in a zero-sum game, where one agent's loss is the other agent's gain. A relevant example of this is InfoGan[9].
- Mixture of Experts-based - is a popular technique for ensemble models and the relevant idea to this work is the usage of a Mixture of Autoencoders to perform clustering, it could therefore be inserted in the autoencoder-based clustering category. This idea is based on a manager which works as a balancing agent, and a set of experts, which are independent neural networks. Some relevant contributions include DAMIC [8] MIXAE [41] and MoE-VAE [31], which provided building blocks for this work.

However, since these state-of-the-art techniques work around complete data, it is of big importance to focus the same ideas on the problem of missing data.

C. Clustering With Missing Data

The approaches above focus on the learning of models with missing data, often to provide either imputation or classification. However, they do not approach the clustering of this data. In this section, the focus is therefore on the clustering part of the problem.

Starting with **K-pod** [11] is a missing data approach that works by extending a K-means clustering algorithm to work with missing data, however since it is a shallow clustering method is more appropriate for smaller datasets.

Subspace clustering is a big topic in the literature that works well on high-dimensional data. It extends classical clustering into finding clusters within different subspaces on a dataset. By finding clusters that exist in multiple and/or overlapping subspaces, it allows the algorithms to localize the most relevant dimensions [26]. Recently there has been some work that introduces missing data into this type of algorithm. In particular, [29] offers two methods focusing on the problem of "partially observed" data and sees the problem of SCMD as a generalization of a low-rank matrix completion problem. Similar ideas can be seen in [28] and [40]

Some work has also been done around the topic of **graph clustering** with deep learning models. Due to the great potential of the usage of graphs across different branches of science, the merge of graph theory with deep learning lead to the emergence of Graph Neural Networks (GNNs) in the last years.

The usage of deep learning for multiple graph analysis has mostly been focused on tasks such as node classification and link prediction, but there has also been some work approaching the problem from a clustering point of view. The goal is to separate the nodes of the graph into different clusters, with the edge structures of the graph being taken into account, leading to a result where there are multiple edges within each cluster and a small number between different clusters.

Variational Graph Auto-Encoders are at the base of most relevant work in the graph clustering tasks, with the current state-of-the-art being based on this idea. This solution is built using a graph convolutional network (GCN) encoder and a simple inner product decoder. A key feature of this work is that some missing data was introduced in the data before the training, through the removal of a percentage of the edges of the graphs.

This could be an interesting approach to exploring the problem of missing data since graphs are flexible structures of data that could also be used to represent data types such as images or texts, which can be modeled as regularized graphs, due to their fixed number of neighbors.

The whole idea of applying deep learning to graph data structures is a whole field per se and even though this type of work aims mostly at data with more heterogeneous structures, there is great potential to use the flexible properties of using graph-structured data.

There has also been some work with **multi-view clustering** using missing data, this type of technique has a focus on multi-view data, which is very common across the field of big data. The goal of this type of task is to consider data from distinct feature sets or "views" and retrieve meaningful information in a way that considers how the data from different views complement each other and their consensus.

Examples of this include multimedia, where both a video and an audio signal can be used to represent a media segment, or when using image data obtained from different devices to film the same object. In the scope of missing data, complementary information from the different views can be used to retrieve the existent missing values.

The ideas analyzed in this section seem to show great potential in dealing with missing data problems, which also proves the relevancy of the work being done in this thesis. They look at the problem however from different lenses, focusing either on different types of data or being more developed for classification tasks, such as the case with GNNs.

III. The Architecture

The development of this thesis started with the creation of an improvement of a standard VAE through the creation of an AE-based architecture to use as a foundation that could be able to work with a variety of datasets and that was flexible to the maximum amount of changes possible. The focus of this dissertation was on image data, however, this scheme can be extended to other data such as survey or sensor obtained data, or even graph-structured data.

With this in mind, the first step was the implementation of an AE-like architecture built with dynamic blocks, the goal was that this flexible architecture could be transformed and adapted into a variety of AE-based architectures by changing the necessary key features such as its loss function and that it could also be adjusted into receiving different data, with the blocks being adapted into the data size.

For the clustering of the data in the latent space of the AE, with the usage of the UMAP dimensionality reduction approach, a manifold learning technique that is applied to the latent space of the autoencoder before a standard clustering algorithm helped improve the quality of the defined clusters, hence leading to more accurate results. The chosen shallow clustering is executed after this step of learning the representations of the data. In particular, HDBSCAN clustering is used, which provides the advantage of automatic detection of the appropriate number of clusters, and also outlier detection.

These key ideas for the clustering of data, and the dynamic convolutional blocks were then used as a base and applied to three different main architectures: 1) a masked variational autoencoder, 2) an IWAE - imputed weights autoencoder, [6] an alternative to the classical VAE that uses a strictly tighter log-likelihood lower bound derived from importance weighting; 3) MIWAE, which is based on IWAE but particularly developed for missing value imputation.

IV. Global Architecture

For the convolutional blocks structure, which can adapt to accommodate any kind of Autoencoder type of architecture, a dynamic structure is presented which is dependent on the data. The order for these convolutional blocks can be seen represented in figure 1, the reasoning behind their structure comes from the advantages of using more dynamic architectures by mixing different kernel sizes. Two distinct blocks are used for this, a general and a specific. In the general block the more general and significant features of the image are reconstructed, while in the specific blocks, more particular features are reconstructed.

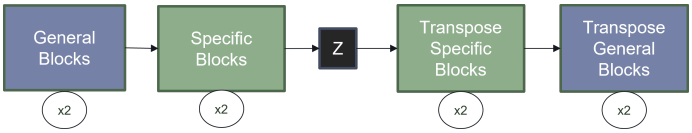


Fig. 1: Diagram Blocks structure that can be aplied to any Autoencoder architecture

Two of the most common choices for filter sizes are 3x3 or 5x5, mostly due to memory and simplicity concerns. In this implementation, the choice was two 3x3 blocks, since even though both have the same receptive field, the chosen one does not require as many mathematical operations, which leads to less training time [34] Another important aspect of the blocks is the usage of Batch Normalization [17] before the activation function, according to [33].

These blocks can be seen in figure 2.

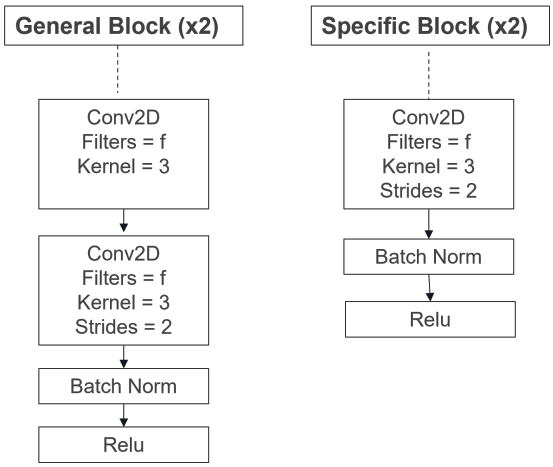


Fig. 2: Diagram of blocks

Another important factor is the depth of the architecture, this depth is dependent on the data since it is not an optimal choice to set the exact same characteristics defined for different datasets. With that in mind, the depth of the network and the number of filters are dependent on the image size of the dataset being used.

Assuming that an image has dimensions DxD, the number of blocks in the encoder can be calculated by:

$$N_{Blocks} = log_2(D) - 1 \tag{1}$$

This amount of blocks in the encoder assures that no matter how big the image is, before the latent space of the encoder the data is 2x2xF in size, where F is the number of filters in the last layer of the encoder. The number of general and specific blocks is given by:

$$N_{general} = \lceil N_{Blocks}/2 \rceil \tag{2}$$

$$N_{specific} = N_{Blocks} - N_{general} \tag{3}$$

The number of filters is given by multiplying D by two for each block of the encoder. This provides an approximately equal number of general and specific blocks in the architecture. The autoencoder-like architecture can be seen summarized in figure 3

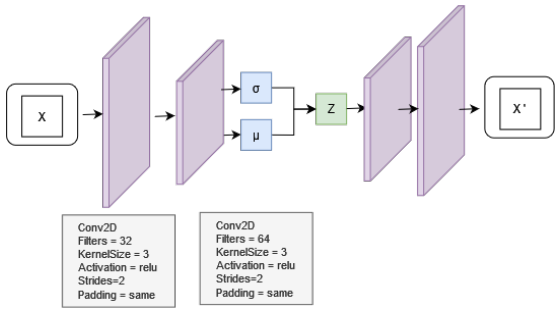


Fig. 3: Vanilla Autoencoder

This AE-like architecture can accommodate any type of autoencoder by changing the latent space. Two variations that were made in this dissertation were transforming it into a MIWAE and IWAE, two architectures not initially designed for images or convolutional layers as the ones seen in the blocks. The main contribution is, however, the adaption of the VAE loss function to receive a binary mask and with that overcome the missing data of received images.

Uniform Manifold Approximation and Projection (UMAP) is a technique for manifold learning used for dimension reduction that can be used in a similar way to t-SNE for visualization while being strong at preserving the structure of the data in smaller dimensions in a fast and efficient way. Due to the importance of dimension reduction in the field of data science, this technique is considered a viable choice for its usage in machine learning. [25] Therefore, for the clustering of the latent space, the first step was to use UMAP for dimensionality reduction. In this step, two variations were tested, one where the parameters were the default ones from the library, and in an alternative, a dynamic number of neighbors was defined. This dynamic number of neighbors influences how locally the data is viewed. The following expression was followed:

$$n\_neigh = max\{int\Big(\frac{dataset\_size}{300}\Big), 100\} \tag{4}$$

After dimensionality reduction with UMAP, a shallow clustering algorithm is applied. Different solutions could be applied for this, such as Gaussian Mixture Models (as in [24]), K-Means (as in [39]), or Hierarchical Clustering. However, in this thesis the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [2] is chosen. HDBSCAN is an extension of the DBSCAN algorithm which transforms it into a hierarchical clustering algorithm. This algorithm offers some advantages compared to classic shallow clustering techniques such as K-means and GMM, namely its ability to automatically detect outliers by creating a different cluster destined for this type of data point. Another advantage is the fact that the amount of clusters is detected automatically according to the data, which is a substantial improvement versus most deep learning-based clustering solutions where the number of clusters must be known a-priori.

With these base characteristics defined in the system, the architecture could work as a normal VAE by defining its loss function to do so.

A diagram of the base VAE described in this section can be seen in figure 3.

A. Masked VAE

One of the main contributions of this dissertation comes from introducing a binary mask when facing missing values on the data. In the proposed solution, a variant of a normal VAE is implemented by changing the loss function of the model. In this variation, a binary mask from the missing points of the data is created. With the mask obtained, it is then used as input along with the images, and in the moment of the loss calculations, multiplied by the reconstructed image.

The goal of this approach was to modify the reconstruction loss function to use a binary mask $m_i \in \{0,1\}^L$ which indicates if a certain value is missing. Thus, the loss function, which corresponds to an MSE, is computed over the observed values in the following way:

$$L_{rec} = \frac{1}{N}\sum_{i=1}^{n}(x_i' - \tilde{x_i}')^2, \tag{5}$$

where $x' = x \circ m$ represents an element-wise product between the original input data x and the mask m, and $\tilde{x}' = \tilde{x} \circ m$ an element-wise product between the reconstructed data $\tilde{x}$ and the mask m.

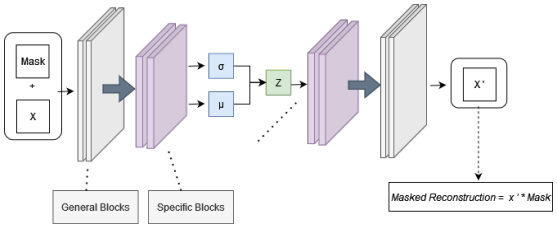


Fig. 4: Masked VAE

This technique aims for the bias introduced by missing data to be ignored by assigning an error value of zero to the areas of the images where missing values are observed.

A visual representation of the VAE model adapted to this idea is represented in figure 4.

The idea of a binary mask was in here introduced in 3 different architectures, but it is a solution scalable to multiple types of structures that use a compatible loss function.

B. IWAE

Importance Weighted Autoencoders [6], is a similar generative model to VAE, offering to learn richer representations with more latent representations. The key difference is that through importance weighting, the generative model is trained with a tighter log-likelihood lower bound. This is done through the generation of multiple "approximate posterior" samples in the recognition network, with averaged weights being used.

This model does not focus on missing data or clustering, however, it is the main base for MIWAE, a model explored in detail in the next section that adopts this model into solving the overhead created by missing data. To explore the fundamental approach to importance weights, some models were implemented and tested. In all of them, the clustering above described was applied in the latent space. Two main implementations are being considered for this analysis: 1) an implementation of the original IWAE, and *2) An adaptation of the base VAE from this dissertation's work with its latent space transformed and the encoder output changed into returning multiple samples.* Since the results from the second implementation were not satisfactory, the results obtained were dropped and only the ones from the first implementation were considered for the final analysis.

The original architecture was tested using the MNIST and the OMNIGLOT dataset and it was not meant for its usage with clustering, however, for this work, the same clustering methodology was used by picking its latent space and performing clustering. Since OMNIGLOT is considered to be more adequate for one-shot classification tasks, its testing was discarded since it's not as relevant for this work.

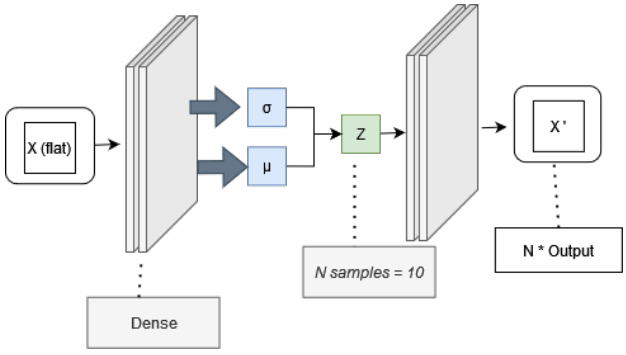


Fig. 5: Original IWAE

The main reasoning for the study of this specific architecture comes from another work in the area, in [23], MIWAE was proposed, used some of the ideas of this solution with a focus on the problem of missing data, where it is also introduced a binary mask to the solution.

C. MIWAE

The MIWAE [23] model focuses on the handling of missing at random data through deep learning. It is based on the IWAE architecture above, and similar to the idea offered in the masked VAE, a binary mask is used during the training. In the original implementation, however, the focus is on continuous datasets and the model is built with a few dense layers.

To work with this extension of the IWAE architecture, some alterations were made. This work was focused around using very simple synthetic numeric datasets, with its main focus being the imputation of the data. To adapt to the problem of this work, an AE-like architecture with convolutional blocks adapted from the one here proposed was implemented. This allowed for the MIWAE architecture to work with convolutional layers and support image datasets. Since this architecture also used the idea of a mask for its training, the same logic was kept.

V. Experimental Results

1) Datasets used:

- MNIST: A dataset of handwritten digits images with 70000 examples separated into ten classes. Each sample is a 28x28 grayscale image.
- FMNIST: A dataset of fashion items images with 70000 examples separated into ten classes. Each sample is a 28x28 grayscale image.
- USPS: A dataset of handwritten digits images with 9298 examples separated into ten classes. Each sample is a 16x16 grayscale image.
- Coil: The Columbia Object Image Library (COIL-20) dataset contains images of 20 objects, and for each of them there are 72 images captured every 5 degrees along a viewing circle. Each sample is a 128x128 grayscale image.
- Cifar10: The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.

To better fit the architecture, the FMNIST and MNIST samples were padded from 28x28 to 32x32, while The COIL20 dataset was downsampled from 128x128 to 64x54 using an antialiasing filter.

2) Creation of Patches: For the generation of an alternative dataset with random missing data, in the initial stages of this work, this was achieved simply by adding empty patches of 10x10 pixels in random central areas of the image, but it later evolved into adding a variance in the height and width of the patches. The reasoning for the focus on the central parts of the image is that since the most important information occurs in the center sections of the images, therefore inserting patches on the edges would not cause a significant impact on the results, since the inserted patches would be considered background. Hence, for the selection of the location of the patch, a margin of 4 pixels in the center of the image is considered and the starting pixel is obtained from a random uniform distribution with a range from *i=4* to *i=width-4* and *j=4* to *j=height-4*.

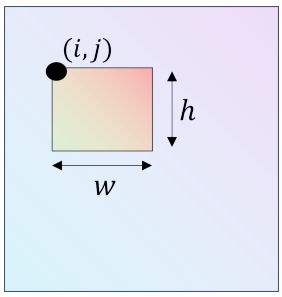


Fig. 6: Samples from MNIST dataset with missing data (substituir por diagrama com os patches

Until the end of this document, the altered datasets will be referenced as:

- MNIST-Patches: The result of adding random patches to all instances from the MNIST dataset.
- FMNIST-Patches: The result of adding random patches to all instances from the FMNIST dataset.
- USPS-Patches: The result of adding random patches to all instances from the USPS dataset.
- COIL20-Patches: The result of adding random patches to all instances from the COIL20 dataset.

The datasets with missing values also include the changes mentioned in the section above, with a resizing of the MNIST and FMNIST samples from 28x28 to 32x32, and the COIL20 dataset being downsampled from 128x128 to 64x64.

3) Evaluation: Clustering accuracy was used to evaluate the performance of HDBSCAN and GMM, which measures the proportion of data points for which the obtained clusters can be correctly mapped to the correct classes. This mapping can be obtained using the Hungarian algorithm [21], and the accuracy is given by:

$$ACC(y_{true}, y_{pred}) = max_T(\frac{\sum_{i=1}^{N} 1(y_{true}(i) = T(y_{pred}(i)))}{N}), \quad (6)$$

where y_{true} represents the ground truth labels, y_{pred} the predicted labels, N is the total number of samples, and finally, T is the best one-to-one mapping that matches the clustering indexes to the ground truth labels.

For the HDBSCAN clustering, two different accuracies were generated, one involving the full data set, and the other one only taking into account the points that were correctly labeled, ignoring the outliers.

For a richer analysis of the results, a GMM clustering accuracy was also retrieved from the same embeddings obtained in the tested models.

Throughout the results shown in this chapter, UMAP was a constant factor in every clustering calculation, where it was used on the embedding before running the clustering algorithm. It is therefore of big importance to first see its impact when used in different situations. Table I, provides an analysis of the impact of adding dimensionality reduction with UMAP when applied to shallow clustering algorithms.

One of the metrics used for this were ARS - Adjusted Rand index [30], which is a way to measure the similarity of the two label sets, ignoring permutations and AMI, Adjusted Mutual Information, which is a way to measure the agreement between two sets, in this case, the obtained labels and the original labels. AMI was more recently proposed and works similarly to Normalized Mutual Information (NMI), which is also often used in the literature.

In table I, it is possible to see the comparison of the usage of UMAP in three cases where shallow clustering is being used: 1) Direct application of the K-means algorithm on a flattened input data, 2) K-means on the embedding obtained from a VAE, 3) GMM with the embedding obtained from VAE. The dataset used for this was a complete version of the MNIST dataset, on the built VAE. For all cases UMAP significantly improved the results, which provided confirmation to the advantages of UMAP offered in the literature [24].

With the improvements from dimensionality reduction obtained, the next step was to observe the difference when performing a density-based clustering algorithm. In the table II the results from the HDBSCAN algorithm when using full labels were added, and since the accuracies obtained were

	No UMAP			UMAP		
	ARS	AMI	ACC	ARS	AMI	ACC
K-means	0.37	0.50	0.53	**0.79**	**0.86**	**0.82**
K-Means + VAE	0.52	0.62	0.65	**0.95**	**0.94**	**0.98**
GMM + VAE	0.83	0.86	0.92	**0.96**	**0.95**	**0.98**

TABLE I: Impact of the usage of UMAP as a dimensionality reduction method when used along with shallow clustering methods

similar, throughout the experiments GMM clustering was also considered.

	ARS	AMI	ACC
VAE + UMAP + GMM	0.96	0.95	0.98
VAE + UMAP + HDBSCAN (Full Labels)	0.95	0.94	0.98

TABLE II: Results from HDBSCAN and GMM clustering.

A. Results of convolutional blocks architecture

The first implementation step was to build a stable architecture that was able to provide good clustering results in a complete dataset. Table III presents the clustering results on the MNIST dataset of the implementation of the VAE with the dynamic convolutional blocks as described in the section above.

The first three lines represent the clustering through the usage of UMAP + HDBSCAN, where the first line shows that the standard VAE is able to assign 67% of the data points to a cluster, with a 43% accuracy, while when considering the whole dataset this accuracy drops to 32%. In the second column, we can see the first indicator of success by the noticeable difference in accuracy, which increases to 98% on both methods of clustering (and with all datapoints being assigned a cluster).

	Basic VAE	VAE with Dynamic Convolutional Blocks
Percentage of labeled points	0.67	1
Accuracy of labeled points	0.43	0.98
Accuracy on full dataset	0.32	0.98
Accuracy of gmm	0.38	0.98

TABLE III: Influence of the dynamic blocks architecture with MNIST dataset

B. Influence of Patches in Clustering Results

After obtaining a stable architecture providing good results on complete datasets, the next step was the analysis of the impact of introducing missing values in the data used for model training. The first glance into this impact is represented in table IV, where the clustering results when considering the full dataset (i.e. including the data points not labeled in the HDBSCAN algorithm) shows that for all the datasets where missing data was introduced, a decrease of the accuracy occurred.

Also considering the same case but considering the labeled data by the HDBSCAN algorithm, table V show the influence on the accuracy of the successfully labeled points, as well as

	Full image		With Patches	
	Med	σ	Med	σ
MNIST	0.978	0.007	0.914	0.008
FASHION MNIST	0.584	0.009	0.537	0.017
USPS	0.967	0.002	0.873	0.057
COIL20	0.803	0.016	0.759	0.021

TABLE IV: Impact of missing values in the accuracy from UMAP + HDBSCAN considering the full dataset. For this experiment the median and standard deviation of 10 runs is considered.

		Full image		With Patches	
		Med	σ	Med	σ
MNIST	Percentage of labeled points	0.980	0.004	0.90	0.009
	Accuracy of labeled points	0.983	0.0005	0.974	0.001
FASHION MNIST	Percentage of labeled points	0.73	0.062	0.84	0.017
	Accuracy of labeled points	0.716	0.033	0.609	0.011
USPS	Percentage of labeled points	0.99	0.009	0.94	0.005
	Accuracy of labeled points	0.973	0.001	0.968	0.001
COIL20	Percentage of labeled points	0.95	0.029	0.89	0.077
	Accuracy of labeled points	0.834	0.022	0.819	0.066

TABLE V: Impact of missing values in the percentage and accuracy of HDBSCAN labeled data. For this experiment the median and standard deviation of 10 runs is considered.

the difference in the percentage of data that the algorithm was able to assign labels.

For the full dataset, there seems to be a correlation between the size of the images and the amount of information contained, for the USPS dataset, where the image sizes were smaller and the images more simple, the biggest difference occurred, with a loss of 10% of accuracy, while the smallest difference can be seen in the COIL20 dataset, the one with the biggest images size.

For HDBSCAN labeled data the accuracies also dropped, as well as the percentage of labeled, which can be seen in V

C. Influence of Mask in the results

After the analysis of the impact of adding missing data analyzed, the next step was to adapt the model to overcome the problem. This is where the usage of a binary mask in the model inputs was introduced with its results being shown in this section. To ensure a more rigorous quality of results, for each trained instance the results were obtained 10 times, with a median and a standard deviation being calculated.

The results showed that through the implementation of a mask on the dynamic VAE, the sensibility to missing data was softened in most cases.

For the FMNIST-Patches dataset, represented in table VI the difference can be seen in the accuracy results for GMM with a difference of 6% and HDBSCAN on the full dataset with 4%. For the subset of labeled data, there seems to be a similar value, however, the percentage of labeled points also increased,

which shows that certain data points were now clustered with the introduction of the mask.

	FMNIST-Patches + Mask		FMNIST-Patches + No Mask VAE	
	μ	σ	μ	σ
Percentage of labeled points	**0.83**	0.029	0.69	0.083
Accuracy of labeled points	0.675	0.009	0.688	0.025
Accuracy on full dataset	**0.578**	0.008	0.537	0.017
Accuracy of GMM	**0.661**	0.021	0.607	0.018

TABLE VI: Influence of mask on FMNIST-Patches dataset

On the MNIST-Patches dataset, represented in table VII, the results align with the observations on the FMNIST-Patches dataset, the HDBSCAN algorithm improved 2% on the full dataset, and the percentage of labeled points also increased. In this case, however, the GMM algorithm showed similar results for both cases.

	MNIST-Patches + Masked VAE		MNIST-Patches + Standard VAE	
	μ	σ	μ	σ
Percentage of labeled points	**0.97**	0.008	0.9	0.022
Accuracy of labeled points	0.972	0.001	0.974	0.002
Accuracy on full dataset	**0.944**	0.004	0.922	0.008
Accuracy of GMM	0.968	0.005	0.964	0.001

TABLE VII: Influence of mask on MNIST-Patches dataset

On the USPS-Patches dataset, the impact is shown in table VIII, the biggest difference can be observed, when applying the HDBSCAN algorithm on the full dataset. Also, the amount of labeled points increases with the proposed approach compared to the standard VAE.

	USPS-Patches + Masked VAE		USPS-Patches + Standard VAE	
	μ	σ	μ	σ
Percentage of labeled points	**0.95**	0.006	0.84	0.049
Accuracy of labeled points	**0.973**	0.001	0.928	0.018
Accuracy on full dataset	**0.94**	0.005	0.873	0.057
Accuracy of GMM	**0.965**	0.001	0.955	0.001

TABLE VIII: Influence of mask on USPS-Patches dataset

Finally, for the COIL20-Patches, represented in table IX, the results do not show improvement of the proposed approach, with only a slight variation in the results of HDBSCAN.

	COIL20-Patches + Masked VAE		COIL20-Patches + Standard VAE	
UMAP - dynamic parameters	μ	σ	μ	σ
Percentage of labeled points	0.94	0.051	0.91	0.037
Accuracy of labeled points	0.879	0.034	0.888	0.026
Accuracy on full dataset	0.847	0.008	0.841	0.017
Accuracy of gmm	0.5	0.0	0.5	0.0

TABLE IX: Influence of mask on COIL20-Patches dataset

D. Alternative misssing values imputation

The VAE needs to have a complete dataset to be able to train, consequently, the missing pixels were substituted by zero after obtaining a binary mask, however, the option to simply replace by other values was still a possibility. Therefore, some experiments were done with this in mind, focusing on the alternatives of using 1)an initialization with random values, 2) the average of the closest pixel imputation and 3) the average of two closest pixels. In this scenario, the dataset used as an example is the FMNIST-Patches.

Table X shows the results of this experiment, it can be concluded that replacing these values with zero is the best option, since not only were the results better when the techniques looking at neighbors were tested it created a relevant increase in computational complexity.

	HDBSCAN			GMM
	Percentage of labeled points	Accuracy of labeled points	Full dataset acurracy	
Random	0.71	0.657	0.526	0.574
Zero	**0.83**	**0.675**	**0.578**	**0.661**
KNN = 1	0.73	0.656	0.527	0.576
KNN = 2	0.7	0.658	0.520	0.565

TABLE X: Clustering resuls for FMNIST-Patches dataset, when considering alternative missing values imputation techniques

E. MIWAE and IWAE architectures

For IWAE and MIWAE, two of the implemented architectures described in the previous section, similar experiments were attempted, where the MNIST and fashion MNIST datasets were used with and without patches of missing data added.

The clustering results for the best cases with the VAE and MIWAE models can be observed in table XI.

	VAE - No Mask		VAE - MASK		MIWAE	
	med	stnd dev	med	stnd dev	med	stnd dev
Percentage of labeled points	0.69	0.083	0.83	0.029	0.59	0.1
Accuracy of labeled points	0.688	0.025	0.675	0.009	0.6	0.05
Accuracy on full dataset	0.537	0.017	0.578	0.008	0.49	0.03
Accuracy of GMM	0.607	0.018	0.661	0.021	0.54	0

TABLE XI: Accuracies of FMNIST-Patches on VAE and MIWAE architectures

For the IWAE architecture, the baseline used was an implementation available at [1], it showed very satisfactory results in the complete MNIST dataset, even though it did not surpass the VAE model proposed in this work. This model also showed one of the biggest impactful differences in clustering results when used with missing values, with the accuracy on the full dataset when using HDBSCAN dropping from 92% to 76%. The introduction of a binary mask showed a slight improvement as well. Similar conclusions can be drawn for the FMNIST dataset, see table XIII.

	MNIST-Patches		MNIST
	No mask	Mask	
Percentage of labeled points	0.75	0.79	0.94
Accuracy of labeled points	0.927	0.924	0.968
Accuracy on full dataset	0.764	0.799	0.924
Accuracy of gmm	0.85	0.85	0.95

TABLE XII: IWAE results on MNIST dataset

Overall the MIWAE and IWAE models did not surpass the proposed VAE architecture, however, some interesting

	FMNIST-Patches		FMNIST
	No mask	Mask	
Percentage of labeled points	0.41	0.48	0.62
Accuracy of labeled points	0.67	0.60	0.62
Accuracy on full dataset	0.35	0.36	0.4268
Accuracy of gmm	0.48	0.47	0.5884

TABLE XIII: IWAE results on FMNIST dataset

	IWAE		MIWAE	VAE	
	No mask	Mask		No Mask	Mask
Percentage of labeled points	0.41	0.48	0.59	0.69	**0.83**
Accuracy of labeled points	0.67	0.60	0.6	0.69	**0.68**
Accuracy on full dataset	0.35	0.36	0.49	0.54	**0.58**
Accuracy of gmm	0.48	0.47	0.544	0.60	**0.66**

TABLE XIV: Comparison between clustering results on FMNIST-Patches betweeen IWAE, MIWAE and VAE

conclusions can be obtained. The overall idea of using multiple samples seems to show that it is not an optimal path to approach the problem, but it does show that the usage of a mechanism to alter the loss function depending on the existing missing values can significantly improve the results.

F. Image Reconstruction

For the analysis of the reconstruction of images from the VAE, IWAE, and MIWAE architectures, the visual outputs obtained were observed and some reconstruction metrics were retrieved, namely MSE and SSIM, which is often used to quantify image quality degradation on data compression or data transmission processes. [32].

However, the biggest observation from the reconstructed data comes from the actual images obtained.

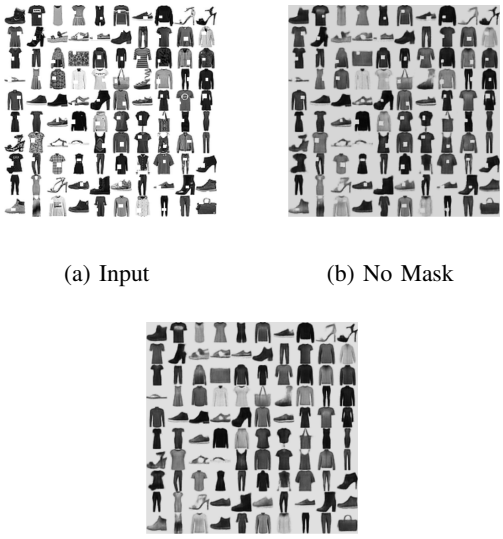
In the following images, a representation of a) the patched data that the models received, b) the reconstruction of the same data when using no mask during training and c) the same reconstruction when using a binary mask.

The biggest difference can be observed when using the FMNIST-Patches dataset, figure 7. When using no mask an interesting phenomenon occurs where the color of the overall images also changes, this is a sign of how a slight removal of information can cause a bias in the final results and reconstructions. When observing the patches in more detail, which is easier to observe through the FMNIST-Patches due to the bigger amount of information in each image, it is possible to see that the patches disappeared almost entirely in the reconstructed image, although with some information lost and the color of the whole image is affected.

G. Final Clustering

The final accuracy results for HDBSCAN on full data, for complete and incomplete datasets, and on the multiple architectures explored in this dissertation is shown in table XV. This allows a full observation of the impact of patches and how the VAE with the usage of a binary mask provides the best results.

Overall the developed VAE architecture in this work, especially after introducing the binary mask achieved the goals and is able to compete with the many different techniques existent in the literature.



(a) Input (b) No Mask

(c) Mask

Fig. 7: Fashion MNIST reconstructions

Clustering on full data set	FMNIST	FMNIST - Patches	MNIST	MNIST - Patches	USPS - Patches	COIL20 - Patches
MIWAE	0.42	0.47		0.44	-	-
IWAE	0.42	0.35	0.925	0.764	-	-
VAE Masked	0.590	0.578	0.98	0.94	0.944	0.811
VAE	0.584	0.537	0.98	0.922	0.873	0.759
MIWAE + VAE		0.52				

TABLE XV: Accuracy on full dataset from the implemented architectures and other architectures in the literature for comparison

VI. Conclusion

With the increasing growth of data-related fields, missing data problems is a recurring problem that is fundamental to approach. In this dissertation, the impact of this obstacle when performing clustering in deep learning models was analyzed and verified. Different ideas were studied, tested, and analyzed, including experiments using imputed weights and multiple sampling, one of the greatest factors that offered the most improvements in the results was the usage of a binary mask in the loss function.

Starting with a simple variational autoencoder that evolved into a more complex architecture dependent on data and that through the usage of a binary mask in the loss function, was able to overcome some of the impact created when missing data in images was added to the problem.

VII. Limitations and Future Work

Due to the flexibility of the implemented architecture and the usage of a binary mask when facing missing data problems, this work could be potentially used to continue researching and exploring more AE-based architectures. The adaptation of this idea to other data types could also be a possibility and was even lightly tested out with simple data.

REFERENCES

[1] nbip/iwae: Importance weighted autoencoders in tensorflow 2, reproducing results from the iwae paper with 1 or 2 stochastic layers.

[2] hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2:205, 3 2017.

[3] Deep learning for clustering of multivariate clinical patient trajectories with missing values. *GigaScience*, 8, 11 2019.

[4] Melanoma lesion detection and segmentation using deep region based convolutional neural network and fuzzy c-means clustering. *International Journal of Medical Informatics*, 124:37–48, 4 2019.

[5] Faroudja Abid. A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology*, 57:559–590, 3 2021.

[6] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[7] Thyago P. Carvalho, Fabrízzio A.A.M.N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto, and Symone G.S. Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers Industrial Engineering*, 137:106024, 11 2019.

[8] Shlomo E Chazan, Sharon Gannot, and Jacob Goldberger. Deep clustering based on a mixture of autoencoders. *IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING*, 2019.

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[10] Chung Yuan Cheng, Wan Ling Tseng, Ching Fen Chang, Chuan Hsiung Chang, and Susan Shur Fen Gau. A deep learning approach for missing data imputation of rating scales assessing attention-deficit hyperactivity disorder. *Frontiers in Psychiatry*, 11:673, 7 2020.

[11] Jocelyn T. Chi, Eric C. Chi, and Richard G. Baraniuk. k-pod: A method for k-means clustering of missing data. *http://dx.doi.org/10.1080/00031305.2015.1086685*, 70:91–99, 1 2016.

[12] Hyun Soo Choi, Jin Yeong Choe, Hanjoo Kim, Ji Won Han, Yeon Kyung Chi, Kayoung Kim, Jongwoo Hong, Taehyun Kim, Tae Hui Kim, Sungroh Yoon, and Ki Woong Kim. Deep learning based low-cost high-accuracy diagnostic framework for dementia using comprehensive neuropsychological assessment profiles. *BMC Geriatrics*, 18:1–12, 10 2018.

[13] Nat Dilokthanakul, Pedro A M Mediano, Marta Garnelo, Matthew C H Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders.

[14] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization.

[15] Simão Eduardo, Alfredo Nazábal, Christopher K I Williams, and Charles Sutton. Robust variational autoencoders for outlier detection and repair of mixed-type data. 2020.

[16] Yu Gong, Hossein Hajimirsadeghi, Jiawei He, Megha Nawhal, Thibaut Durand, and Greg Mori. Variational selective autoencoder. pages 1–17, 2019.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015.

[18] Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. not-miwae: Deep generative modelling with missing not at random data. *arXiv preprint arXiv:2006.12871*, 2020.

[19] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[20] Zhentian Jiao, Youmin Zhang, Jing Xin, Lingxia Mu, Yingmin Yi, Han Liu, and Ding Liu. A deep learning based forest fire detection approach using uav and yolov3. *1st International Conference on Industrial Artificial Intelligence, IAI 2019*, 7 2019.

[21] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 3 1955.

[22] Fengfu Li, Hong Qiao, Bo Zhang, and Xuanyang Xi. Discriminatively boosted image clustering with fully convolutional auto-encoders. 2017.

[23] Pierre Alexandre Mattei and Jes Freiisen. Miwae: Deep generative modelling and imputation of incomplete data. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:7762–7772, 12 2018.

[24] Ryan McConville, Raúl Santos-Rodríguez, Robert J. Piechocki, and Ian Craddock. N2d: (not too) deep clustering via clustering the local manifold of an autoencoded embedding. *Proceedings - International Conference on Pattern Recognition*, pages 5145–5152, 8 2019.

[25] Leland Mcinnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. 2020.

[26] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl.*, 6(1):90–105, jun 2004.

[27] Michael Phillips, Helen Marsden, Wayne Jaffe, Rubeta N. Matin, Gorav N. Wali, Jack Greenhalgh, Emily McGrath, Rob James, Evmorfia Ladoyanni, Anthony Bewley, Giuseppe Argenziano, and Ioulios Palamaras. Assessment of accuracy of an artificial intelligence algorithm to detect melanoma in images of skin lesions. *JAMA Network Open*, 2:e1913436–e1913436, 10 2019.

[28] D Pimentel, R Nowak, and L Balzano. On the sample complexity of subspace clustering with missing data.

[29] D. Pimentel-Alarcon, L. Balzano, R. Mareia, R. Nowak, and R. Willett. Group-sparse subspace clustering with missing data. *IEEE Workshop on Statistical Signal Processing Proceedings*, 2016-August, 8 2016.

[30] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846, 12 1971.

[31] Pedro T ´ Avora Santos, Doutor Pedro, Tomás Tom´, Tomás Doutora, Helena Aidos, Doutora Tereza, Vazão Vaz˜, Vazão Supervisor, : Doutor, Pedro Tomás, and Tom´ Tomás. A mixture-of-experts approach to deep image clustering estimating latent sizes and number of clusters electrical and computer engineering examination committee, 2020.

[32] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

[33] Casper Kaae Sø nderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sø nderby, and Ole Winther. Ladder variational autoencoders. 29, 2016.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. pages 1–9, 2015.

[35] Jannis Tautz-Weinert and Simon J. Watson. Using scada data for wind turbine condition monitoring – a review. *IET Renewable Power Generation*, 11:382–394, 3 2017.

[36] Kim Han Thung, Pew Thian Yap, and Dinggang Shen. Multi-stage diagnosis of alzheimer's disease with incomplete multimodal data via multi-task deep learning. *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,...*, 10553:160–168, 2017.

[37] Fei Wang, Lawrence Peter Casalino, and Dhruv Khullar. Deep learning in medicine—promise, progress, and challenges. *JAMA Internal Medicine*, 179:293–294, 3 2019.

[38] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[39] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

[40] Congyuan Yang, Daniel Robinson, and René Vidal. Sparse subspace clustering with missing entries.

[41] Dejiao Zhang, Ann Arbor, Mi Yifan Sun Technicolor Los Altos, Ca Brian Eriksson Adobe San Jose, and Ca Laura Balzano. Deep unsupervised clustering using mixture of autoencoders.

[42] L ; Zou, J ; Zheng, C ; Miao, M J Mckeown, and Z J Wang. 3d cnn based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural mri diagnosis of attention deficit hyperactivity disorder using functional and structural mri. 3d cnn based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural mri. 2017.