LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR DATENBANKSYSTEME
UND DATA MINING

Bachelor Thesis
in Computer Science

Feature Learning and Importance Scoring on Incomplete Anomaly Datasets

Julian Hoffmeister

Supervisor:	Prof. Dr. Peer Kröger
Advisor:	Andreas Lohrer
Submission date:	04.10.2022

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.
This paper was not previously presented to another examination board and has not been published.

Munich, 04.10.2022

.....
Julian Hoffmeister

Abstract

Clustering algorithms often rely on complete sets of data. This is, however, not practical in real-world use cases. Simply removing rows of data with missing values is not recommended, as it not only reduces the sample size but also opens up the possibility of information loss. Therefore, value imputation methods are an important pre-processing technique to run clustering algorithms on incomplete data sets. They are becoming more advanced and no longer simply employ heuristics such as mean substitution. In recent years, machine learning methods have proven effective in filling missing values. Nonetheless, all imputation strategies can be expected to have an impact on the distribution of the feature in question and have difficulties when dealing with outlier datasets.

In this work, a methodology is being investigated to fill tabular outlier datasets with missing values in such a way that the original distribution is preserved. Additionally, the outlierness of the distribution is considered in order to optimise subsequent tasks that benefit from distribution-aware information representations (like clustering tasks). By adapting the TabNet framework using customized loss functions such as the Kullback–Leibler divergence loss, it is possible to encourage adherence to the original distribution. The experiment results show, that the target to preserve the feature distribution and outlierness can be achieved for multiple datasets at different ratios of missing values in one or more features. The approach also performed well when applied to lower dimensional versions of the datasets. On the other hand, there is no clear indication, that the approach can improve subsequent clustering performance. In some cases, value imputation with TabNet could reach ARI scores close to those of a clustering on the original data. Regression imputation did, however, provide much more reliable results in that regard.

Contents

Chapter 1

Introduction

Real data often contain missing values (MVs). Among other things, this can happen due to refusal to answer a survey, manual typing errors or equipment malfunction [19]. Clustering algorithms often cannot handle MVs. Many of them, such as DBSCAN, rely on distance functions, that cannot be computed for data points with MVs in one or more of their dimensions. A common approach to deal with this is to simply ignore the affected data rows. This has significant downsides like information loss resulting in biased decision making. The impact is also depending on the kind of MV pattern, see Chapter 1.2. Missing value imputation is an active field of research and many different imputation strategies have been developed. They can help to fill the gaps in data. For cluster datasets, MV imputation can have different consequences. New clusters could be created, existing clusters can be merged or reinforced or data with imputed values could simply become noise.

This chapter serves as an introduction to the problems and theoretical foundations and also describes the research questions derived from them.

1.1 Problem Statement & Motivation

Many machine learning clustering algorithms, e.g. DBSCAN[18] or k-means[27], fail for missing value datasets, as the distance function relies on full data. For classification problems there are algorithms, that can deal with missing values (for instance ensemble of networks or fuzzy methods)[20]. Adaptions of existing algorithms, such as k_m-means[30], are trying to solve the MV problem for clustering purposes.

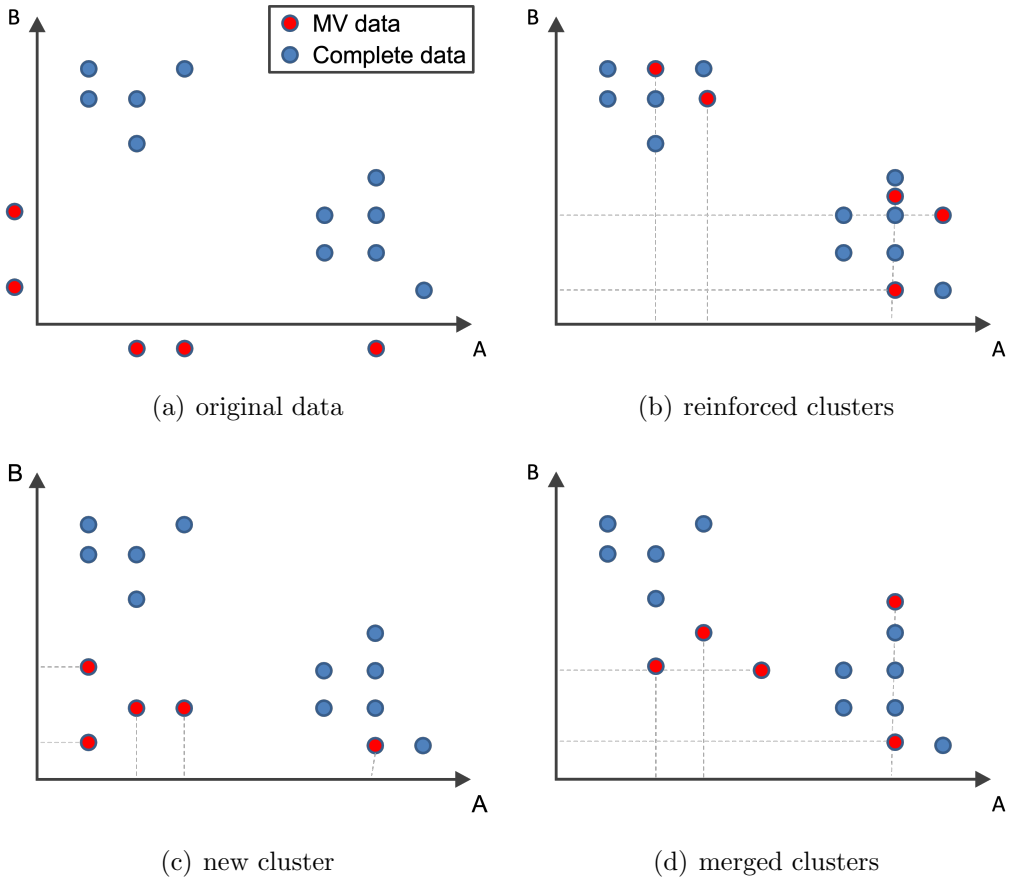
Simply deleting rows with MVs could lead to a loss of information. There might be a correlation between missing values and other features. So the fact, that a value is missing can be important information by itself (cf. Chapter 1.2). This method should only be used for datasets with small amounts of

missing values that do not cause a bias in the resulting dataset [33].

Thus it would be practical to fill the dataset with values, that can be expected to be as close to their actual value as possible. There are a lot of different imputation strategies, ranging from the insertion of heuristic values (such as the mean or median of a feature) to model-based approaches (using machine learning techniques).

The precision of these strategies varies, depending on the used algorithm, hyperparameters and datasets. Ultimately, all these approaches will impact the distribution within a feature as well as the clustering of the dataset.

This problem is only worsened when dealing with outlier datasets, i.e. datasets with a significant amount of data points that do not follow known patterns of the rest of the data. Outliers are difficult to predict, as their values can hardly be learned by the data available (or else they would not be outliers).



(a) original data

(b) reinforced clusters

(c) new cluster

(d) merged clusters

Figure 1.1: Influence of differently imputed missing values.

The imputation of MVs can have drastic consequences for subsequent clus-

tering of MV datasets. Even small rates of missing values in datasets can significantly impact the clusters identified[4]. Given the simple example of a 2-dimensional dataset as shown in (cf. Figure 1.1) it can be seen, that clusters can change dramatically depending on the number of outliers and their imputed values.

Thus it is interesting and necessary to examine the impact of imputation strategies on subsequent clustering of data (cf. RQ1 1.3).

Furthermore, the clustering can be highly dependent on the dimensionality of the dataset, i.e. the number of features. Clusters can be hidden in a subset of the dimensions and might not be detected in a higher dimension. Not only the clustering depends on the dimensionality of the dataset. The prediction of missing values is also based on other features (with existing values in the same row). With that in mind, it makes sense to also examine the impact of missing dimensions (cf. RQ2 1.3).

Last, but not least, with modern model-based imputation approaches it can become difficult to explain, how these values were derived. Especially for critical use cases such as healthcare applications, there is a strong need for explainable machine learning algorithms. Consequently, the imputation strategy used previously to clustering should be explainable (cf. RQ3 1.3).

1.2 Theoretical Background on Missing Values in Datasets and Filling Strategies

Missing values will occur in most real-world datasets. As stated before, this can significantly impact the performance of algorithms using the data. The causes of missing values, be it as refusal to answer a survey, manual typing errors or equipment malfunction [19], can hardly be avoided. For that reason, it is important to find ways of dealing with them.

It is common to distinguish three types of missing values [31]:

- **Missing completely at random (MCAR)**: The reason for the missing value is completely random, i.e. the probability of data missing is not influenced by other features of the dataset[8]. As a consequence, the is no inherent information in the fact, that the data is missing (it could have just happened by accident, e.g. a malfunction of the measuring device).
- **Missing at random (MAR)**: The missing value depends on an already known value (another observed feature of the data point) and does not

depend upon the missing value itself.

- **Not missing at random (NMAR)**: the probability of a value missing correlates with another unrecorded feature (or itself). I.e. the missingness is not completely at random, but cannot be explained by the dataset either.

If an algorithm depends on complete datasets, there are generally two ways to deal with missing values: discarding them or replacing them with values (e.g. by estimation). The literature describes different ways of categorizing imputation strategies. Commonly used is a differentiation between 1) deletion of data, 2) single imputation and 3) multiple imputation [8]. Furthermore, 4) machine learning approaches are becoming extremely common and can be added as a fourth category[20].

1) Deletion of data

This is the simplest option for dealing with missing values, including listwise deletion (excluding the entire data row) and pairwise deletion (using only the available features of each data point for further analysis) [19]. However, it comes with significant disadvantages [2], such as biased data if MVs are not MCAR. In the case of MCAR missing values, unbiased results can still be obtained, but it might not be the best option due to the reduced dataset size.

2) Single Imputation

Single imputation includes approaches, that create a single version of the MV dataset with replaced values. Among others, this covers heuristics substitution methods (replacing MVs with heuristics, such as the feature mean or median). Substitution can produce high biases, especially when the share of missing values is high [34]. This phenomenon occurs in Chapter 4.4.

Regression imputation[20] is another single imputation approach, trying to derive missing values by their correlation to other (available) features of the dataset. This could be done by linear or non-linear regression. In contrast to heuristics substitution, this approach will preserve the variance and covariance of the reconstructed data.

3) Multiple Imputation

This method of missing value imputation was developed by Rubin[33]. Its intuition is to create multiple versions of the imputed dataset and use the

average imputation as the final value. The approach follows three steps: imputation of missing values in each copy of the dataset (resulting in multiple different complete datasets) followed by an analysis of the datasets and at last the pooling of the datasets into one final complete dataset.

4) Machine Learning Approaches

Multiple machine learning-based imputation approaches have emerged over the past years, among others [19][20][1]:

- Imputation with K-nn
- Recurrent Neural Network
- Random Forest
- Support Vector Machine
- Multi-Layer Perceptron
- Attention-Based Transformer Networks

Furthermore, deep learning approaches have been developed to work on tabular data. The Value Imputation and Mask Estimation (VIME) framework[16] claims to allow for feature imputation by masking missing values and learning from other non-masked features.

1.3 Research Questions

Given the problem stated above, Chapter 2 will derive three research questions.

Research Question 1 (RQ1)

Since MV imputation strategies can drastically affect clusters in a dataset, it is important to examine their impact on clustering algorithms applied to imputed data. Several single, multiple and machine learning-based imputation approaches have been investigated (cf. Chapter 1.2) and partially studied regarding their clustering performance (cf. Chapter 2.1). Consequently, research question 1 is derived in Chapter 2 focusing on model-based imputation methods:

- How is it possible to improve the clustering of missing values/dimensions datasets with model-based filling strategies?

Research Question 2 (RQ2)

To further investigate the impact of the dimensionality of the datasets, research question 2 is:

- How does the number of dimensions influence the cluster/outlier results?

Research Question 3 (RQ3)

The third research question is dedicated to the explainability of model-based imputation strategies:

- How is it possible to make model-based filling strategies explainable?

Chapter 2

Related Work

The problems described in the introduction 1 have already been researched to varying degrees. Existing work on the topics of MV imputation and its influence on subsequent clustering as well as on importance-based feature weighting is summarised in this chapter. Three research questions are then derived from the identified research gaps.

2.1 Existing Approaches to Fill Missing Values for Subsequent Clustering

When viewing related work for *clustering* and *missing value imputation* the first thing to find are imputation strategies based on clustering algorithms such as DBSCAN[32], K-means[10], CMI[25]. This does however not necessarily mean, that the imputation strategies are beneficial for the performance of a subsequent clustering.

Other research is focusing on the development of clustering algorithms, that work on MV datasets, such as k_m-means[30] or k-means-FWPD and HAC-FWPD algorithms[23]. This eliminates the need to impute missing values before clustering.

Some papers examine the influence of missing values and imputation strategies on subsequent tasks. In [20] the authors describe and test different imputation approaches for following classification tasks. As a performance metric to optimise, they are using PAC (predictive accuracy) and DAC (distributional accuracy). The classification is done by an artificial neural network.

Luengo et al.[14] propose three different groups of classifiers. According to

the authors, using determined MV imputation methods for each group could improve the accuracy of respective classifiers, also stating that there "is no universal imputation method that performs best for all classifiers". The performance of each combination of an imputation and classification method was measured by the classifications accuracy.

There are several studies investigating the effect of different imputation strategies on downstream clustering within the context of gene expression clustering[5][15][11][17]. They follow an approach to use either real-world MV datasets or artificially create them, then run imputation algorithms on them and measure the quality of subsequent clustering with different metrics.

According to [5], most studies are using imputation accuracy as a metric, e.g. by calculating the RMSE. The authors claim, that a more practical approach would be to also consider the discriminative/predictive power for classification/clustering purposes. The study examines multiple imputation strategies and their effect on different classification and clustering techniques (such as k-medoids and hierarchical clustering). Clustering quality was measured via the corrected Rand score. The results indicate no statistical evidence, that the imputation strategies used had a significant impact on the classification/clustering quality. Complex imputation strategies could not outperform even simple approaches like mean substitution. The study is however limited to low rates of MVs with a maximum of 9%. Compared to [5], this thesis will use different performance scores and investigate higher rates of MVs.

Another study on gene expression microarray data[15] investigates different imputation strategies and their ability to reproduce complete datasets and subsequently the original gene partitions. The authors used eight datasets and generated MV data at different ratios ranging from 0.5% to 20%. After imputing the MV with different strategies, the k-means algorithm was applied for clustering using varying values for k. Clustering results were compared to the original clustering via the Average Distance Between Partition (ADBP) error. The authors observed, that using imputation strategies even with strongly varying accuracy ratings made little difference for subsequent clustering. However, the investigated imputation strategies were reported to outperform simple methods such as mean substitution or ignoring missing data. Similar results were observed by [11].

Furthermore, Celton et al. [17] evaluated the impact of 12 different imputation strategies on the quality of gene clustering with five biological datasets. While the authors underline the efficiency of imputation methods (measuring

the RMSE), they also observed that imputation quality decreases with the number of missing values. Furthermore, even low MV rates can result in very unstable clusters. The study additionally investigates extreme values (top and bottom 1% of the distribution), concluding that most strategies decrease in effectiveness. This effect does however depend on the strategy of choice, as well as on the dataset. The authors proceed to apply hierarchical and k-means clustering to the datasets reconstructed with the 12 imputation strategies. Two metrics were used to evaluate the clustering performance, namely the Conserved Pairs Proportion (CPP) and the Clustering Agreement Ratio (CAR). Both metrics are based on a comparison of clusters obtained by clustering the original data (reference clustering) and the reconstructed data (generated clustering).

The majority of work regarding imputation for subsequent tasks appears to focus on a biology context. Little research could be found on the distribution of the data within the datasets and only one study investigates the impact of outliers in the data. Several machine learning (ML) approaches for MV imputation have been developed in recent years. However, no research could be found related to their impact on subsequent clustering performance (the studies presented above did not consider ML approaches). Consequently, RQ1 1.3 is addressing these gaps. Most of the studies observed a strong dependency of the imputation and clustering performances with the datasets used. There is a lack of research on the correlation between the dataset dimensionality and the MV distribution among features. I.e. whether there is a correlation between the features used for MV prediction, the number of features containing MVs and the performance of the clustering. The gap regarding the dimensionality is addressed by RQ2 1.3. Lastly, none of the studies are addressing the model-based explainability of imputation-related machine learning approaches. The topic does however gain importance recently and should be addressed when applying imputation and clustering approaches. RQ3 1.3 addresses the lack of research regarding the explainability of ML imputation approaches.

2.2 Importance Based Feature Weighting

In the studies described in section 2.1, the imputation accuracy did often have little impact on the cluster performance. Does this also apply to lower-dimension versions of the same dataset? If some features were to be more important for the MV imputation, omitting them could drastically reduce the accuracy and with it the clustering performance. The term feature importance can thus be understood as the relevance a feature has for the prediction

of missing values.

When investigating clustering algorithms, feature importance could also be seen as a dependence of the clustering on that feature, i.e. how well would a given clustering hold, if that feature was omitted? However, with an increasing number of dimensions, data points in a cluster tend to move away from each other as their distance grows by adding a dimension. This ultimately results in the cluster dissolving. Consequently, omitting features could improve the clustering. The dimensionality of a dataset generally plays a major role in clustering, as the curse of dimensionality[3] effect can drastically impact the results. Some clusters can only be detected in a certain subspace of the dimensions of the datasets. This also implies that the meaning of clustering highly depends on the context, i.e. the dataset itself. There is a large amount of research in this field, also covering different approaches to handling the problem of clustering for high-dimensional datasets. Examples are projected clustering methods like PROCLUS[9], subspace clustering like Clustering in Quest (CLIQUE)[22] and its adaption Merging of Adaptive Finite Intervals (MAFIA)[24] as well as dimensionality reduction methods, such as Principal Component Analysis (PCA). Projected clustering is a top-down approach, trying to find an optimal subset of dimensions for each cluster. Subspace clustering methods aim to identify clusters in all possible subspace projections (bottom-up) using a given cluster definition. PCA on the other hand is not a clustering technique but allows us to find correlations in the data and reduce the number of dimensions for subsequent cluster analysis. The k-medoid-based PROCLUS algorithm even ranks dimensions for each cluster by their relative spread within the cluster.

In contrast to clustering algorithms, there are approaches to weight features based on their predictive value in regression or classification problems. A simple example would be to obtain the coefficients of a linear regression model or the importance scores of a classification and regression tree. These methods are already common and can be easily accessed via libraries such as sklearn (for python)[12]. However deep learning techniques are often considered black boxes and explainability has become an important topic. Using feature weights can be a useful tool to describe the decision process of a deep learning model. There is a limited amount of approaches for applying deep learning techniques to tabular data. In recent years methods like VIME[16], Neural Additive Models (NAM)[21], TabNet[1] and Layer-wise Relevance Propagation (LRP)[13] have gained attention.
Value Imputation and Mask Estimation (VIME) is a self- and semi-supervised learning framework for tabular data (cf. Figures 2.1 and 2.2). The masking mechanism of this approach allows to mask and learn missing values from

other non-masked features. The authors claim to achieve high performances on various datasets.

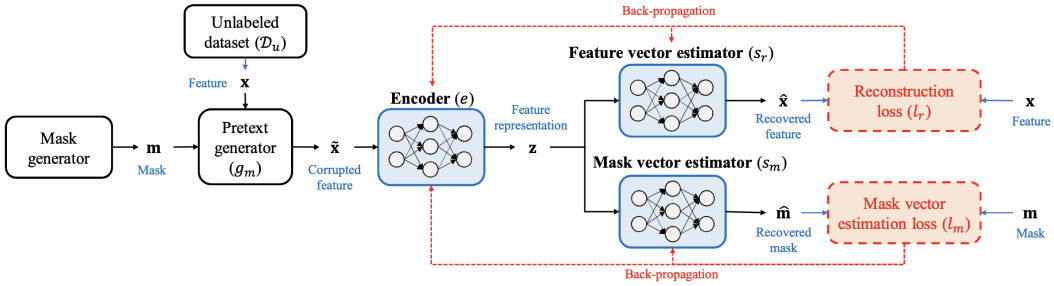


Figure 2.1: VIME - self-supervised learning framework [16]

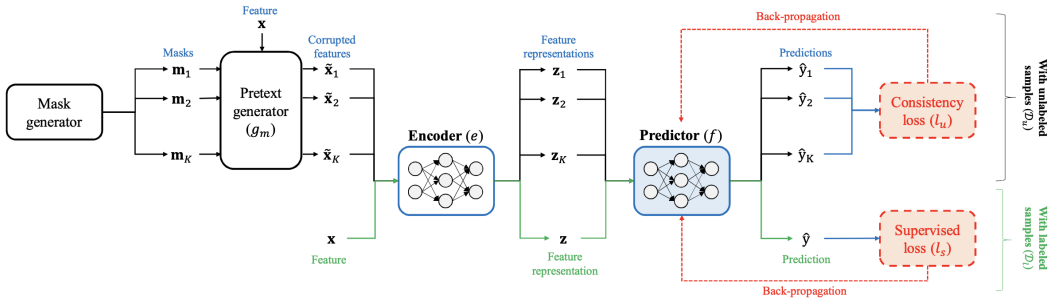


Figure 2.2: VIME - semi-supervised learning framework [16]

Neural Additive Models (NAMs) are using a linear combination of neural networks (cf. Figure 2.3). Each network learns the coefficient of its respective input feature. They are trained via backpropagation and can learn arbitrarily complex shape functions. A major advantage of the approach is its explainability. With each input feature being fed into a different neural network and their outputs simply summed up, it is easy to interpret the results of the NAM.

TabNet is a deep neural network (DNN) architecture that works with tabular data using gradient descent-based optimization. It employs a sequential attention mechanism choosing the most salient features at each decision step by a feature transformer block. This mechanism allows for improved interpretability as decisions can be tracked via the attention TabNet gives to the respective features. TabNet generally consists of an encoder (cf. Figure 2.4) and a decoder (cf. Figure 2.5).

In each step, a mask is generated by an attentive transformer block to extract information about the model. This enables us to gain feature importance

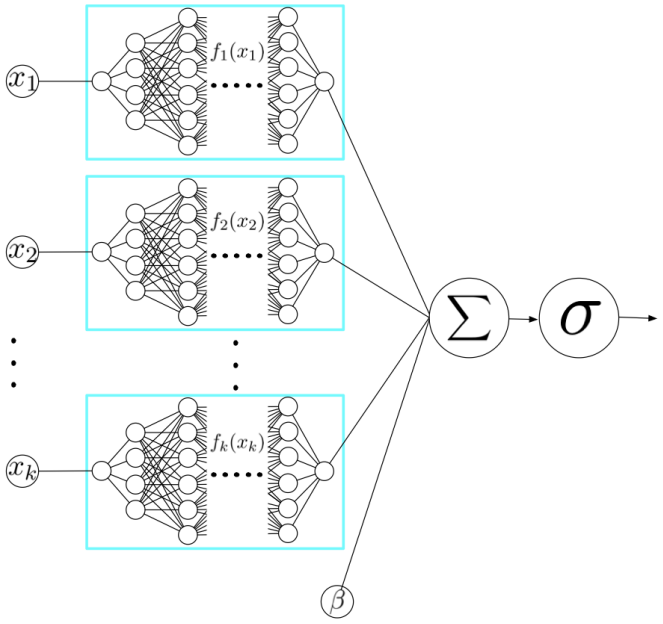


Figure 2.3: NAM - architecture for binary classification [21]

values for each prediction (e.g. of missing values using the TabNet regressor). For a trained model an attention matrix can be obtained as shown in figure 2.6, showing the local feature importance for each decision step as well as the aggregated, global importance.

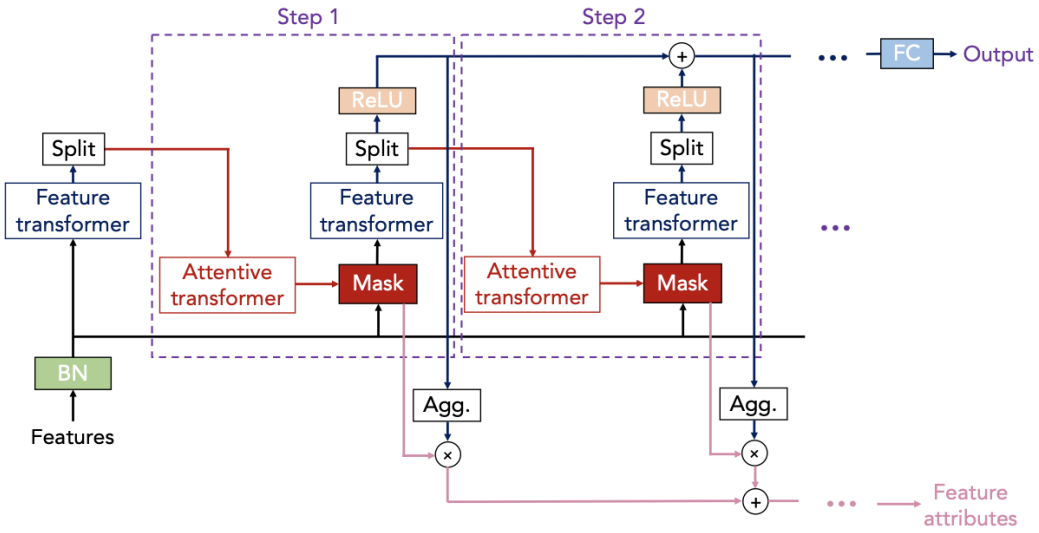


Figure 2.4: TabNet - encoder architecture [1].

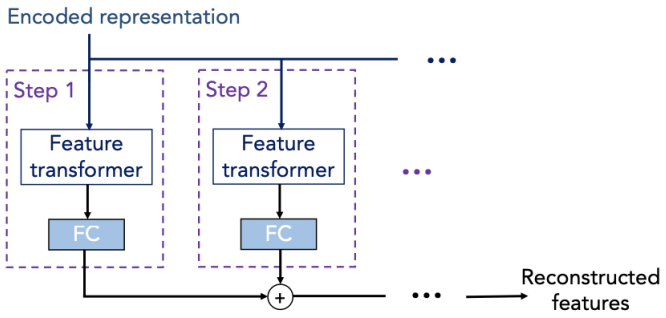


Figure 2.5: TabNet - decoder architecture [1].

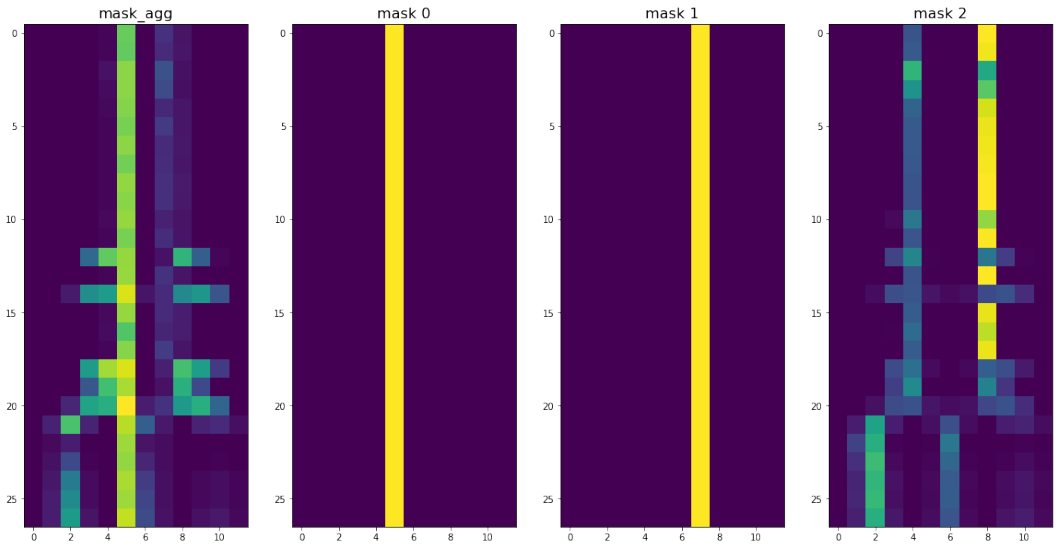


Figure 2.6: TabNet - attention matrix[1] of a model trained on the wine dataset[7] with 20% missing values. Different features were reasoned from different decision steps. In total, feature 6 had the largest influence on the predictions.

Chapter 3

Concept

Given the research questions described in chapter 1.3, a concept is proposed for generating missing value datasets, then imputing the MVs and evaluating the subsequent clustering performance. Figure 3.1 shows the general approach on a top level.

Figure 3.1: Concept - Overview.

The concept realizes the five steps as follows:

1. Datasets: the first step covers the choice of datasets to be used for missing value imputation and clustering.

2. Filling Strategies: This step uses the MV datasets from 1 and applies different imputation strategies to them. The outcomes are predictions (or replacement values) for the MVs of the dataset.

3. Filled Datasets: the imputed values from step 2 are merged with the MV dataset, resulting in reconstructed, complete datasets.

4. Clustering: a clustering algorithm is run on the reconstructed datasets, trying to reproduce the clusters of the original dataset.

5. Performance Metrics: finally, the clustering performance is analysed using different indices.

The relevant parameters and their abbreviations can be seen in Table 3.1.

With slight adaptions, this concept can be used for all three research questions. The concept for RQ1 1.3 follows the general approach: multiple filling strategies are applied to different MV datasets. The resulting complete datasets are fed to a clustering algorithm. Finally, the approaches are evaluated regarding their clustering performance using the scores described in 4.1. For RQ2, instead of using all features of the datasets, a percentage of randomly chosen features is omitted in steps 1 and 2. Thus, the predictions are based only on a subset of the features available. The subsequent clustering is done on the full-dimensional dataset. The same indices as for RQ1 are used. Regarding RQ3, a decision tree is derived from the imputed values of the reconstructed dataset (after step 3) to showcase, how explainability can be increased.

3.1 Datasets

This step covers the choice of datasets, the selection of features used for the analysis as well as the generation of missing values for complete datasets (see figure 3.2).

Figure 3.2: Concept - Generation of Missing Values.

Choice of datasets There are multiple possible sources for datasets with

Parameters	
k	hyperparameter k (# of clusters) used for k-means clustering
mv	percentage of original data to be replaced by NaN (missing value)
$mv_{outlier}$	percentage of missing values, that were outliers in the original distribution
$mv_{features}$	percentage of **additional** (non-imputation) features that have missing values generated
mv_{mask}	mask value for MVs in non-imputation features
n_{models}	number of TabNet models trained per run with different train/valid sets
n_{runs}	number of dataset versions generated with mv% of MVs
std	distance from the mean used for outlier definition (in standard deviations)
$min_{outlier\_+/-}$	min. percentage of outliers required in both tails of the distribution (used for feature selection)
$min_{outlier\_total}$	min. percentage of outliers in a feature to be considered
$n_{omitted\_features}$	percentage of randomly selected features removed from the dataset
train/valid-split	training-validation-split for TabNet fitting process
α_l	custom loss weights for loss components $l \in$ {REC, D, CM}
max_epochs	max. number of epochs during model training
patience	number of consecutive epochs without improvement before early stopping

Table 3.1: Parameters and Abbreviations.

missing values. One option is to use real-world data with values missing. In most cases, that implies, that data is missing at random (MAR), i.e. a correlation between the missingness and other features of the dataset can be expected. Another option is to use complete (real or artificial) datasets. MVs can then be randomly generated (MCAR or MAR). In the following, complete datasets are used and missing values are created randomly. Multiple datasets are used for the approach to be tested (cf. Chapter 4.2).

Selection of features: to limit complexity, the focus lies on only one single feature at a time (although several features with missing values can exist). As one goal is to examine the effect of outliers on the predictions and subsequent clustering (cf. RQ1 1.3), the feature of interest (imputation feature) should contain a sufficient amount of outlier values. Outliers are defined by their distance from the mean. A feature is considered a candidate if it contains a minimum number of data points with a minimum distance of *std* standard deviations from the mean.

Generation of missing value: MVs are generally created randomly by replacing a defined percentage of values with NaN. To later examine the approach regarding its ability to deal with outliers it is also demanded, that a minimum percentage of the generated MVs have to be in the outlier region of the distribution of the feature (i.e. within a certain distance from the mean). Missing values are being generated for only the selected imputation feature or for several features. The imputation strategies described in the next section are, however, only applied to the selected feature.

3.2 Filling Strategy

Multiple imputation strategies are used to analyze their influence on clustering performance. Single imputation (via mean substitution) and regression imputation are considered as a baseline to compare to a model-based deep learning approach. There are different possible machine learning and deep learning approaches, as described in chapters 1.2 and 2.2. This work uses TabNet, as it allows for custom loss functions, working on tabular data (there are only a few deep learning approaches on tabular data) and also enables a higher level of explainability via its attention matrix mechanism (which is promising for RQ2 1.3 and RQ3 1.3). For each dataset and its MV versions generated in step 1, the missing values are imputed as shown in figure 3.3.

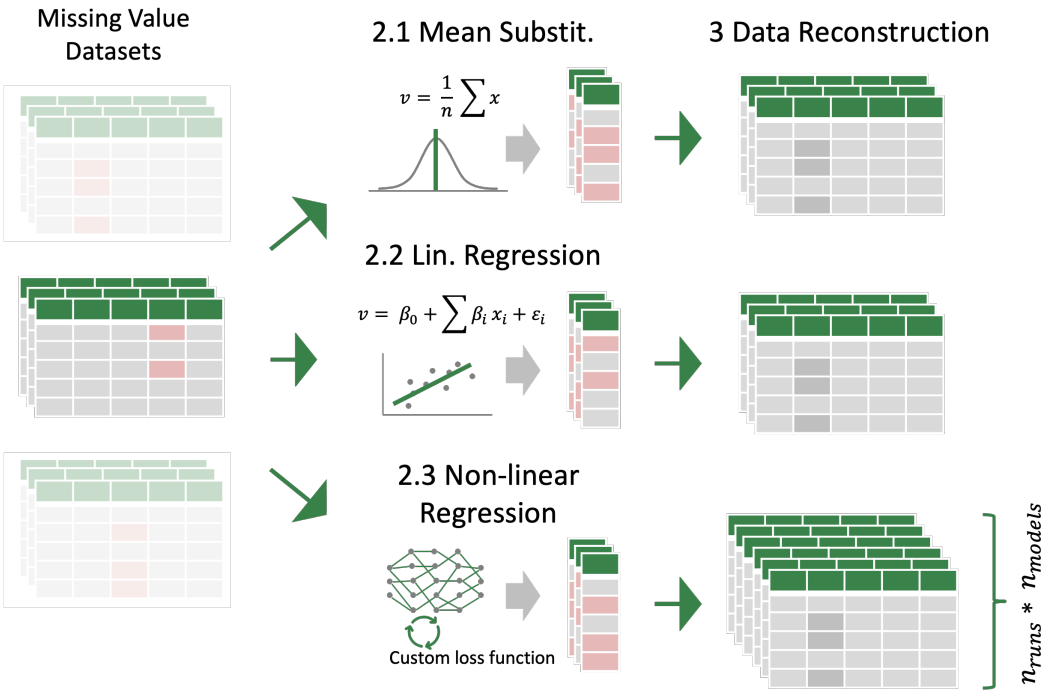


Figure 3.3: Concept - Imputation Strategies and Data Reconstruction.

3.2.1 TabNet Imputation

To utilise TabNet as an imputation tool, the imputation feature (containing missing values) is used as the target y of a TabNet regressor. The model is trained on the complete dataset (omitting data points with missing values in the imputation feature) using the available non-imputation features x_i. The set of incomplete data points serves as the test set. If a non-imputation feature x_i contains missing values, these are replaced by a mask value (e.g. the value -1000) that is to be expected far outside the feature value range. The intention is to enable TabNet to learn from the missingness of other values, i.e. TabNet can use the NaN information, if the target feature y correlates with NaN values in non-imputation features x_i. An alternative would be to only use complete data rows (with the disadvantages described in 1.2) or to impute the missing values (adding another layer of complexity to the problem).
The **custom loss (L_C) function** optimises the TabNet training process, helping to improve prediction accuracy and subsequent clustering performance. It aims to optimize according to three different objectives represented by the respective partial loss functions using the following formula:

$$L_C = \alpha_{REC} * L_{REC} + \alpha_D * L_D + \alpha_{CM} * L_{CM} \tag{3.1}$$

The partial losses are described as followed:

- **Reconstruction Loss** (L_{REC}): The objective of this component is to optimise the prediction accuracy and recreate the original values as well as possible by minimizing the difference between actual values (y) and predicted values ($\hat{y}$). This is the first component of the custom loss and is implemented as the RMSE (root-mean-square error) using the formula

$$L_{REC} = \sqrt{\frac{\sum_{i=1}^{N_y}(y_i - \hat{y}_i)^2}{N_y}} \tag{3.2}$$

  where N_y is the number of predictions made (i.e. the number of missing values in the context of this work).

- **Distribution Loss** (L_D): This loss intends to preserve the target feature distribution. Even for features with a high amount of outliers (and potential outliers marked as missing values), the predicted distribution shall be as close to the original distribution as possible. The loss represents the second component of the custom loss and is implemented as the Kullback–Leibler (KL) divergence[29]. This loss is punishing deviations from the feature distribution (more details in Chapter 4.1).

- **Cluster Mean Loss** (L_{CM}): The last component of the custom loss function has the goal to reinforce existing clusters (cf. Figure 1.1). Depending on knowledge about clusters and noise within the dataset, it can be argued, that missing values should be predicted in a way, that the data point will end up within an existing cluster. To achieve this, a Cluster-Mean-Loss (CML) is introduced. The intuition is to calculate the sum of distances of MV data points to their nearest clusters and minimize this value within the training process. The Cluster-Mean-Loss L_{CM} is intended to pull the data points with missing values towards their closest cluster centre. The respective formula is:

$$L_{CM} = \sum_{i=1}^{N_y} \min\left(\sqrt{\sum_{l=1}^{d}(x_{i,l} - c_{1,l})^2}, ..., \sqrt{\sum_{l=1}^{d}(x_{i,l} - c_{k,l})^2}\right) \tag{3.3}$$

  with D = {1, ..., d} being the dataset dimensions (including the imputation dimension), X = {$x_{1,1}$ ... $x_{N_y,d}$} being the data points with their predicted missing values, C = {($c_{1,1}$, ..., $c_{1,d}$), ..., ($c_{k,1}$, ..., $c_{k,d}$)} being the set of cluster centres (described by their coordinates) and k the number of clusters. L_{CM} is reflecting the sum of Euclidean distances of data

points with imputed values to their closest cluster. During the training process, TabNet is minimizing this sum by choosing the missing values in such a way, that the points are drawn towards the clusters. This loss does, however, not guarantee for the points to end up within the clusters, since it can only move them on their respective MV axis.

The coefficients (weights) α_{REC}, α_D and α_{CM} of the components can be chosen manually. This way, it is possible to either use loss combinations, focus on one objective only or weigh the degree of magnitude of each loss (this could be useful, as the Cluster-Mean-Loss is typically much larger than the Distribution Loss and the Reconstruction Loss). Several loss functions are used to investigate the concept performance with experiments in Chapter 4, including each of the three partial losses L_{REC}, L_D and L_{CM} on their own as well as the mean absolute error (MAE) and a combination of L_{REC} and L_D (cf. Chapter 4.3.2). Since L_D and L_{CM} are pursuing contradictory goals, no respective combination of them will be investigated.

To use the Cluster-Mean-Loss, it is necessary to first identify an original clustering and calculate the cluster means. This is done using the k-means algorithm, as DBSCAN has proven to perform worse for higher dimensional datasets, resulting in a very unstable clustering (either all data points were assigned to one cluster or all data points were considered noise). This could likely be due to the curse of dimensionality (see Chapter 2.2) .

Furthermore, for certain training-validation sets, TabNet has shown to fit models that perform exceptionally bad. To counteract this phenomenon, multiple (n_{models}) models with different train/valid-split sets are trained, reducing the likelihood of that event occurring.

3.2.2 Other Imputation Strategies

As a baseline to compare the TabNet approach to, regression imputation and mean substitution are used. For each MV dataset version created in the first step, both methods are applied. Just as for the datasets reconstructed by TabNet, they are then used for subsequent clustering and analysis.

3.3 Reconstructed Datasets

The missing values can now be filled by the imputation methods described in section 3.2. The resulting reconstructed datasets are the input for the subsequent downstream task (clustering), which is described in the next section

3.4 (see figure 3.3). These datasets are expected to have different feature characteristics depending on the imputation strategy used. As mean substitution is replacing all MVs with the same value, this method heavily influences the resulting distribution and the downstream clustering task (creating or reinforcing a high-density subspace cluster in the dimension of the missing values). Furthermore, mean substitution is not able to predict outliers. Regression imputation and model-based imputation can both be expected to provide more diverse results, closer to the original distribution. The impact on clusters and outliers should generally be smaller (closer to the original). The model-based TabNet approach furthermore allows for enhanced distribution preserving via the distribution loss. This in turn is also expected to improve the prediction of outlier values. Datasets, that are reconstructed using predictions based on L_{CM} should not preserve the original distribution, but increase the cluster performance for datasets with very distinct clusters (and little noise). The actual impact of the loss functions can be seen in the Figures and Discussion of Chapter 4.

3.4 Downstream Task Clustering

Generally, a variety of tasks can be performed on the reconstructed dataset, including classification, anomaly detection and clustering. The focus of this work is subsequent clustering. Similar concepts can, however, be derived for other tasks. Running a clustering algorithm on the reconstructed data (see figure 3.4) provides the assigned clusters based on the predicted values. Just as for the original clustering, k-means is used.

3.5 Performance Criteria

Finally, performance criteria can be calculated to determine the clustering performance, as shown in figure 3.4. The performance indices are defined in Chapter 4.1 and chosen to reflect the objectives described in Chapter 3.2.

3.6 Imputation Explainability

Regarding RQ3, TabNet generally promises a high level of explainability via its attention matrix output. This can be seen in Chapter 2.2. This topic is further investigated by extending the five steps above. A regression decision tree is derived from the reconstructed dataset (filtered by the formerly predicted

Figure 3.4: Concept - Clustering and Performance Criteria.

values). This way, it is possible to reproduce the decision-making process by following the branches of the tree. The resulting tree still needs to be validated (e.g. by experts in the field of the related dataset), but it can help to explain some of the predictions on a high level. A showcase of this method is presented in Chapter 4.4.

Chapter 4

Experiments

With the concept proposed in chapter 3, it is possible to verify the approaches for RQ 1 - 3 1.3. The following sections specify the evaluation criteria (cf. Chapter 4.1), datasets (cf. Chapter 4.2) and the experiment setups and parameters used (cf. Chapter 4.3). The experiment results are then presented and discussed in section 4.4.

4.1 Evaluation

All experiments done related to RQ1 and RQ2 are working with the following scores and indices:

- **Adjusted Rand Index** (ARI): The ARI is an extension of the Rand Index (RI) and is commonly used for measuring clustering performance when true labels are available. To calculate it, the pairs of predicted and true labels of each data point are counted in a contingency table. For a completely random clustering, given a true cluster label, the predicted labels would spread equally among all possible clusters. I.e. for the iris dataset containing 50 instances of each of the 3 labels, a random clustering would represent a contingency table with all values being 50/3 (of course only integer values are possible, so in this case, a completely random cluster assignment is not possible). In this case, the ARI score would be close to 0. If all cluster predictions match the true labels, the ARI score is 1. The RI score can be computed using the formula

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

  with TP being the true positives, TN the true negatives, FP the false positives and FN the false negatives. The ARI is then adjusted using

the formula

$$ARI = \frac{(RI - RI_{expected})}{(RI_{max} - RI_{expected})} \tag{4.2}$$

This index is used to address RQ1 1.3 and RQ2 1.3 and reflects the approaches ability to recreate the original dataset clusters. For the experiments, the median ARI is reported for each imputation strategy.

- **Kullback–Leibler (KL) divergence** [29]: the KL divergence quantifies the difference between a probability distribution Q (of a feature after generation and imputation of MVs) and a reference probability distribution R (in the context of this work: the true / original distribution). For two identical distributions, the KL divergence is 0. The further the distributions diverge from each other, the larger the KL value will be. The KL divergence is not symmetrical. It adds up probability differences using the formula:

$$D(Q \parallel R) = \sum Q(x_i) * \log(\frac{R(x_i)}{Q(x_i)}) \tag{4.3}$$

Outlier values are punished with larger divergence values, while divergences closer to the mean have little impact. This behaviour can be flipped by calculating $D(R \parallel Q)$ instead of $D(Q \parallel R)$. As a consequence, mean substitution is expected to perform worse since the probability of the mean value is far higher than in the original distribution. On the other hand, it could be valuable to punish outliers more when computing the KL divergence loss of the TabNet model, thus valuing the predictive performance for outliers higher. The KL divergence is meant to address RQ1 1.3 and 1.3 by measuring the distribution preservation. In the discussion of results (cf. Chapter 4.4), the median value of each imputation strategy is reported.

- **Outlier Score**: the intention is to measure the performance of outlier predictions. Each outlier removed in a MV dataset version is checked, whether or not it was later predicted correctly as an outlier. The outlier score is then defined as a value between 0 and 1 using the formula

$$Outlier\_Score = \frac{n_{outliers\_correct}}{n_{all\_outliers}} \tag{4.4}$$

This measure is not perfect, as it would consider an outlier predicted correctly even though it might be on the other tail of the distribution in the reconstructed dataset. However, in combination with the KL divergence, it should be possible to make a statement about the approach‘s distribution preservation. Just like for the KL divergence, the index addresses RQ1 1.3 and RQ2 1.3 and its median score is reported in the results.

4.2 Datasets

Three datasets with varying sizes are chosen for the experiments and accessed via the UCI Machine Learning Repository [7]. The iris flower dataset [26] is fairly small (n=150) and has only four features. The wine dataset has a similar size (n=178), but significantly more features ($n_{features}$=13). The dry bean dataset [28] has a large size (n=13611) and 16 features. For each dataset, one imputation feature was chosen for further analysis, as explained in 3. The characteristics and distributions of the chosen features of each dataset can be seen in table 4.1 and figure 4.1.

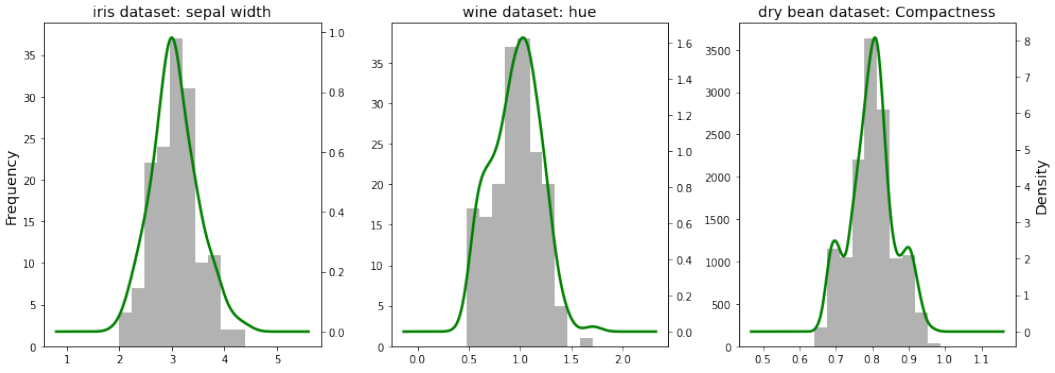


Figure 4.1: Experiments - Distributions of selected features.

iris dataset

This dataset is among the best known for pattern recognition literature[7]. The data describes three different types of iris plants by their shapes (sepal length/width and petal length/width). There are 50 data entries for each type of plant. The true labels are available as $y_i \in \{0, 1, 2\}$. Feature *sepal width* was chosen for the analysis, using the outlier definition and parameters from table 4.2.

wine dataset

As stated on the UCI Machine Learning Repository [7], this dataset is the results of a chemical analysis of three types of wines, grown in the same region in Italy by three different cultivars. The features represent 13 chemical attributes of the wines. Again, the true labels are available as $y_i \in \{0, 1, 2\}$. Feature *hue* was chosen for the analysis.

dry bean dataset

The dry bean dataset contains data on seven different kinds of beans and is the largest of the three datasets in sample size, amount of features as well as labels. Features are mainly describing the geometrical shape of beans. True labels are available as $y_i \in \{0, 1, 2, 3, 4, 5, 6\}$. The feature *Compactness* was chosen for the analysis.

Characteristic	iris	wine	dry beans
n	150	178	13611
$n_{features}$	4	13	16
imputation feature	sepal width	hue	Compactness
$n_{outliers}$	23	28	2506
$ratio_{outliers}$	0.153	0.157	0.184

Table 4.1: characteristics of the datasets and their imputation features.

4.3 Experiment Setups and Parameters

To run experiments, the concept is implemented as described in Chapter 4.3.1. Several experiments to investigate the research questions (cf. Chapter 1.3) are defined in Chapter 4.3.2. The experiments vary different parameters, including general parameters (such as the ratio of missing values generated), outlier definition parameters and model training parameters (cf. Chapter 4.3.3 and 4.3.4). The following sections explain the implementation, the experiment setups, the choice of fixed and variable parameters as well as describing their impact on the general concept.

4.3.1 Implementation

The concept is implemented as a Jupyter notebook plus an adapted version of the TabNet library[1]. The code and experiments can be found in the related git repository[2].

The notebook realises a pipeline allowing to define experiments via a config file. The experiment can then be executed without code adaptions. Experi-

[1] S. Ö. Arik and T. Pfister, *TabNet: Attentive Interpretable Tabular Learning*, 2020, `https://github.com/dreamquark-ai/tabnet`

[2] J. Hoffmeister, *Feature Learning and Importance Scoring on Incomplete Anomaly Datasets*, 2022, `https://github.com/JulianH-LMU/feature_learning`

ment outputs, such as graphs, data and models can be saved optionally. The code is structured similar to the concept.

Slight adaptions to the TabNet code are necessary to integrate the custom loss function. The custom TabNet version is part of the repository[2] and allows the user to pass arbitrary keyword arguments to the fit() method of the TabNet model. These arguments are passed to the loss function. This feature is used to provide the cluster centres for the calculation of Cluster-Mean-Losses, as described in 3.

To enable reproducibility of experiments, it is possible to define seed values in the config file. If no seeds are given, they are chosen randomly and saved within the config file of the output folder.

4.3.2 Experiment Setups

RQ1 1.3 and RQ2 1.3 are answered following the concept described in Chapter 3. In the first step, n_{runs} different versions with missing values are generated for the chosen dataset. This is done to as a stabilisation mechanism, as very rare missing value distributions could distort the results (for instance, when all outlier values in the original distribution are replaced by missing values). The initial clustering is done before applying the imputations strategies, as the cluster centres are required to calculate the Cluster-Mean-Loss. The imputation strategies *mean substitution*, *regression imputation* and *model based imputation* are then used to fill MVs with actual values. The model-based imputation with TabNet is applied using different loss functions:

- **Root-Mean-Square Error** (RMSE)
- **Mean-Absolute Error** (MAE)
- **Distribution Loss** (DL, as described in Chapter 3.2, using the custom loss and setting the weights for L_{REC} and L_{CML} to zero)
- **Cluster-Mean Loss** (CML, as described in Chapter 3.2, using custom loss with zero-weights for L_{REC} and L_D)
- **Custom Loss** (CL, as described in Chapter 3.2, using a combination of L_{REC} and L_D with equal weights and setting L_{CML} to zero)

For each dataset version, TabNet fits n_{models} models using different train-/validation-sets. The random seeds used for the train/validation-sets are saved to the output config file to enable reproducibility of the results. The TabNet model with the lowest cost is used to predict the MVs.

The imputed values of all strategies are used to reconstruct the MV dataset versions, obtaining multiple complete versions for each approach. In the case of TabNet, the MV dataset versions are filled with the model predictions. Mean substitution and regression imputation are easily available using the sklearn library. Clustering is performed subsequently using k-Means with k being the number of unique labels in the original dataset. The performance of the clustering is finally analysed using the indices of chapter 4.1.

The same concept is used for RQ2 1.3, investigating how predictions based on a varying amount of features influence the subsequent clustering. I.e. for a given dataset, several versions are generated by removing a certain amount of randomly chosen features. The imputation strategies are then applied to these dataset versions. The TabNet prediction quality is not expected to depend on different features equally, as some features might have a stronger correlation with the feature of interest. For that reason, a high variance of results is expected based on which features were omitted. While this approach should not impact the mean substitution method at all (as it is independent of other features), it can be expected, that regression imputation performs worse with an increasing number of omitted features. To evaluate the prediction results in an isolated fashion, the subsequent clustering is performed on the full, reconstructed dataset (using the imputed values). That means, that MVs are imputed based on lower dimensional datasets, but the clustering is performed with all dimensions.

Another approach to answer RQ2 could be to not only predict using a lower dimensional version of the dataset but also apply the subsequent cluster algorithm to the reconstructed version of this dataset. A problem with this approach is, that when clustering in a lower subspace, it cannot be expected to identify the original clusters. As a consequence, the performance cannot be measured as before (using the true labels). For that reason, the concept is limited to only investigating the impact of the imputation strategy on the clustering of the full-dimensional reconstructed dataset version.

To answer RQ3 (cf. Chapter 1.3) and further improve explainability, an approach is tested to train a **regression decision tree** explaining the decision process of TabNet imputations.

4.3.3 Fixed Parameters

Unless specified differently, the parameters used for the experiments are as defined in table 4.2. For the experiments of concept 3 only some of them are

varied to measure their impact on the results (cf. Chapter 4.3.4).

The missing value parameters were chosen based on the datasets to guarantee a minimum amount of outliers within the MV distribution as described in 3. n_{models} and n_{runs} were chosen as a compromise, reducing rare occurrences of distributions that might distort results on the one hand (as explained in 3, e.g. when all outliers are marked as missing values) and not increasing run times too much on the other hand. The values for std, as well as $min_{outlier\_+/-}$ and $min_{outlier\_total}$, are chosen mostly depending on the three datasets used, as the goal is to investigate data with different distributions and outlier ratios. $n_{omitted\_features}$ is set to 0%, as this parameter is only relevant for the experiments related to RQ2. Tryout experiments have led to the choice of the train/valid-split, max_epochs and patience parameters. The TabNet parameters were constant for all datasets used. The parameters α_{RMSE}, α_{KLD} and α_{CML} were used to vary the loss function between Distribution Loss L_D only ([0,1,0]), Cluster-Mean-Loss L_{CM} only ([0,0,1]) and a default Custom Loss L_C of ([1,1,0]) combining Reconstruction Loss L_{REC} and Distribution Loss L_D. For TabNet, the original paper states that its performance is not very sensitive to most hyperparameters. For the following experiments, the default parameters are being used.

4.3.4 Experiments Definition

Experiments for each RQ were performed based on the setup described in Chapter 4.3.2. The following section explains the details and variable parameters for the experiments on *Imputation Optimisation* and *Imputation Explainability*.

Imputation Optimisation

To evaluate the optimisation of the clustering performance after imputation (cf. RQ1 1.3), the capabilities of the proposed loss objectives (cf. Chapter 3.2) are investigated by the following experiments. The results can be seen in Chapter 4.4. The missing value parameters are varied as defined in table 4.3:

- Ratio of Missing Values (mv): the intention is to determine, how the MV ratio impacts the imputation and subsequent clustering. As the training and validation datasets shrink with an increasing amount of missing value data points, it is to be expected, that prediction performance is suffering.
- Ratio of Missing Value Features ($mv_{features}$): in this case, we want to examine, how much the predictions of the feature of interest depend on

Experiment Parameters		
k	hyperparameter k (# of clusters) used for k-means clustering	3 (wine), 3 (iris), 7 (dry bean)
mv	percentage of original data to be replaced by NaN (missing value)	15%
$mv_{outlier}$	percentage of missing values, that were outliers in the original distribution	20%
$mv_{features}$	percentage of **additional** features that have missing values generated	0
mv_{mask}	mask value for MVs in non-imputation features	-1000
n_{models}	number of TabNet models trained per run with different train-val-sets	5
n_{runs}	number of dataset versions generated with mv% of MVs	10
std	distance from the mean in standard deviations used for outlier definition	1.5
$min_{outlier\_+/-}$	min. percentage of outliers required in both tails of the distribution (used for feature selection)	5%
$min_{outlier\_total}$	min. percentage of outliers in a feature to be considered	15%
$n_{omitted\_features}$	percentage of randomly selected features removed from the dataset	0%
train/valid-split	training-validation-split for TabNet fitting process	75%
α_l	custom loss weights for loss components l $\in$ {REC, D, CM}	depends on experiment
max_epochs	max. number of epochs during model training	250
patience	number of consecutive epochs without improvement before early stopping	50

Table 4.2: Configuration parameters for the experiments.

Variable Parameters	
Research Question 1	
mv	[0.05, 0.1, 0.15, 0.2, 0.25]
$mv_{features}$	[0.1, 0.2, 0.3, 0.4, 0.5]
Research Question 2	
$n_{omitted\_features}$	[0.1, 0.3, 0.5, 0.7, 0.9]

Table 4.3: Variable experiment parameters for RQ1 and RQ2.

missing values in other features. To do so the same amount of missing values are generated in a certain percentage of other randomly selected features. These incomplete datasets are used for the predictions only. For the subsequent clustering, the original dataset is used, only replacing the predicted value in the feature of interest. I.e. we assume, that the missing values of all other features were predicted perfectly and we only examine the predictions of the feature of interest.

Both experiments were performed on all three datasets with the five loss functions explained in sections 4.3.3. For the example of an experiment with varying MV ratios ([0.05, 0.10, 0.15, 0.20, 0.25]) and the setup described above n_{runs}=10 random dataset versions are generated for each of the MV ratios, then n_{models}=5 TabNet models are trained with random train-validation-splits for each of the versions. This is done for the 3 datasets using 5 different loss functions, resulting in a total of 3750 models. For each run, only the best of the 5 trained models is chosen to do the predictions. Consequently, there are 750 reconstructed datasets. In addition, each of the 50 MV dataset versions is filled with mean substitution and regression imputation. For all three datasets, this results in another 300 datasets.

The same approach is used to investigate the influence of the dataset dimensionality on clustering optimisation and its dependence on a subset of the features available (cf. RQ2 1.3), however, varying the **amount of omitted features** (cf. Table 4.3. For this experiment the Cluster-Mean-Loss are ignored, as a meaningful calculation of a lower dimensional distance to the full-dimension cluster centres is not possible. The results are described in 4.4

Imputation Explainability

To further improve explainability (cf. RQ3 1.3), an approach is tested to train a **regression decision tree** explaining the decision process of TabNet imputations for the iris dataset (for results, cf. Chapter 4.4). The decision tree is trained on the subset of data with imputed values gained from the best-performing imputation model (out of 5 trained models) and a MV ratio of 15%. Custom Loss (as explained in 4.3.2 is used for the imputation model training. The sklearn [12] regressor tree is used with its default parameters.

4.4 Results and Discussion

This section goes through each research question (see 1.3) and describes the results of the related experiments. The results of each experiment are then evaluated and discussed.

Research Question 1

How is it possible to improve the clustering of missing values/dimensions datasets with model-based filling strategies?

Variable Ratio of Missing Values

The first experiment intends to investigate the optimisation of downstream clustering based on the proposed imputation strategy at different ratios of missing values. 10 versions of MV datasets were generated for each of the three datasets and MV ratios of 5%, 10%, 15%, 20% and 25%. The imputation was done via mean substitution, regression imputation and TabNet (using five different loss functions). While mean substitution generally struggled to preserve the feature distribution, both other strategies performed fairly well (cf. Figure 4.2).

Figure 4.3 shows the resulting median KL divergences of each MV ratio and imputation strategy. As is to be expected, mean substitution has the largest impact on the feature distribution. Increasing MV ratios cause an influx of mean values in the distribution, consequently raising the KL divergence. TabNet and the regression imputation perform relatively similar. Both strategies perform worse with increasing MV ratios. For the wine dataset, TabNet appears to be slightly better, while regression is showing the best performance for all MV ratios for the dry bean dataset. One exception is TabNet using the Cluster-Mean-Loss. This strategy causes larger divergences, especially for

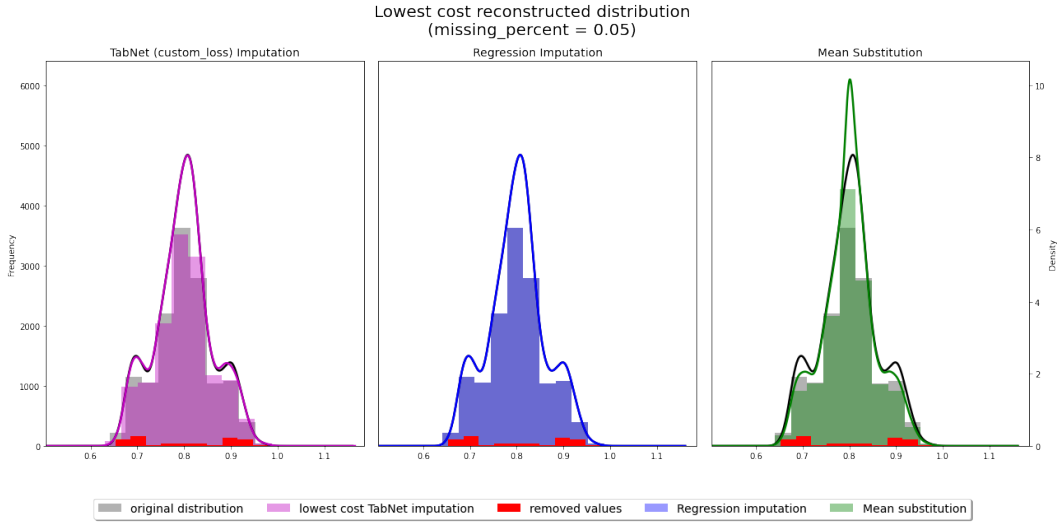
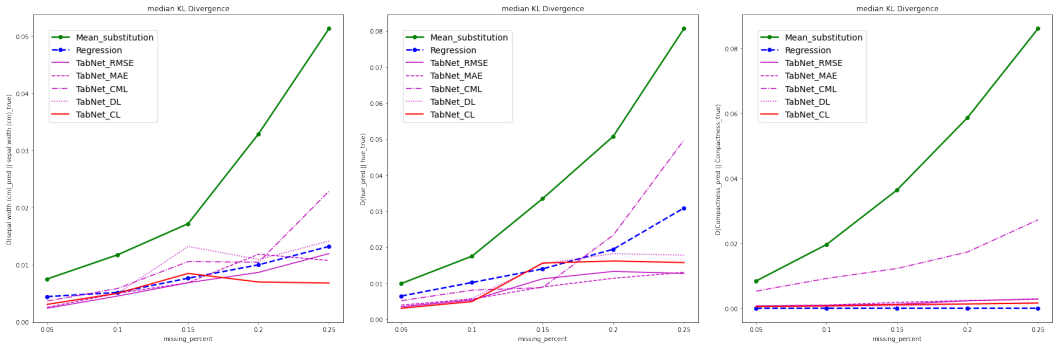


Figure 4.2: Distribution recreated by the best performing TabNet model using KL divergence loss for a MV ratio of 5% on the dry bean dataset.

higher MV ratios.

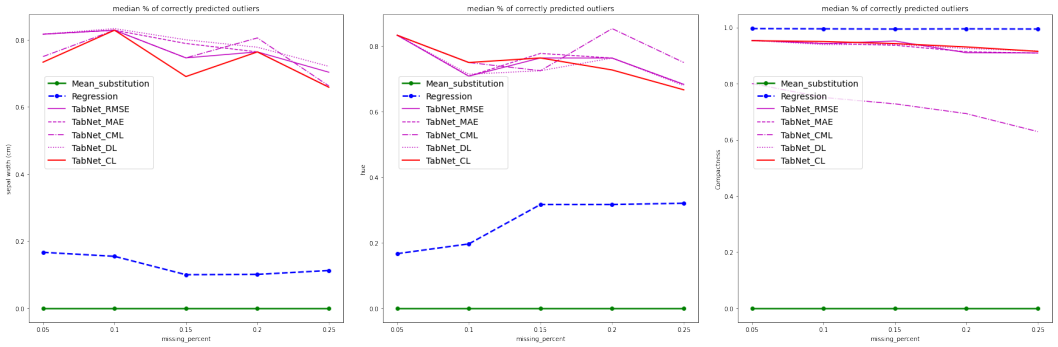
The outlier scores related to the experiment are shown in figure 4.4. While mean substitution is incapable of predicting outliers, its outlier score is zero for all MV ratios. TabNet on the other hand shows very good outlier scores of over 95%. While regression imputation underperforms for the smaller datasets with scores around 20%, it outshines even TabNet for the large dry bean dataset with values at 99.5%. Again, the models using CML have the lowest scores of the TabNet imputations.

Figure 4.5 shows the ARI scores for all three datasets. The ARI score is meant to measure how well the clustering is performed after using different imputation strategies. In this category, mean substitution performs very well for the smaller iris and wine datasets (sometimes even surpassing the ARI score of the clustering performed on the original iris dataset). It does however clearly suffer for the larger dry bean dataset. Regression imputation performs extremely well on all datasets, recreating the original clustering in most cases. The TabNet approach varies strongly, especially for the iris dataset. It does however rarely reach the other imputation strategies. TabNet seems to work well on the wine dataset at lower MV ratios, however, the scores decrease drastically at MV ratios larger than 15%. ARI scores for the dry bean dataset are relatively high. Only the CML approach is underperforming, barely competing with mean imputation.



(a) iris dataset (b) wine dataset (c) dry bean dataset

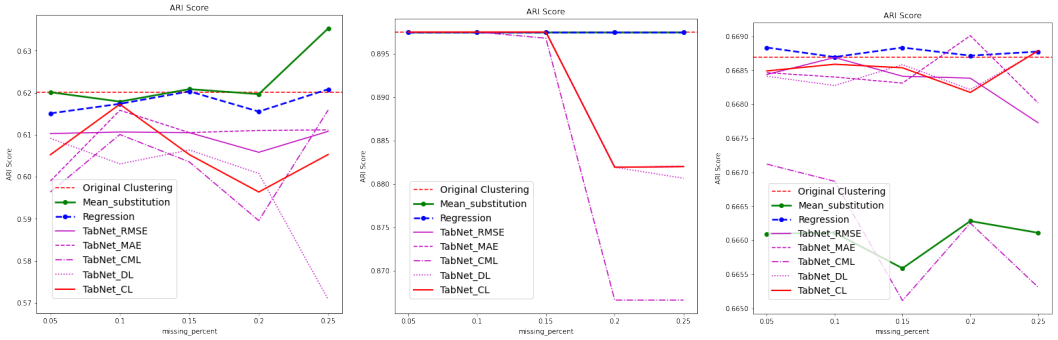
Figure 4.3: Variable *mv* - Median KL divergence of all three datasets after mean substitution, regression imputation and TabNet imputation.



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.4: Variable *mv* - Median outlier scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

Discussion of results: The bad performance of TabNet with CML regarding KL divergence and outlier score is to be expected, as the Cluster-Mean-Loss is not designed to preserve the feature distribution. However, the approach does not seem to perform well with the intent to improve the clustering accuracy either. In general, the TabNet approaches do not show better results than regression imputation. However, it outperforms mean substitution regarding the preservation of feature distributions. The use of different loss functions (apart from CML) does not have a clearly distinguishable impact on the scores.



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.5: Variable *mv* - Median ARI scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

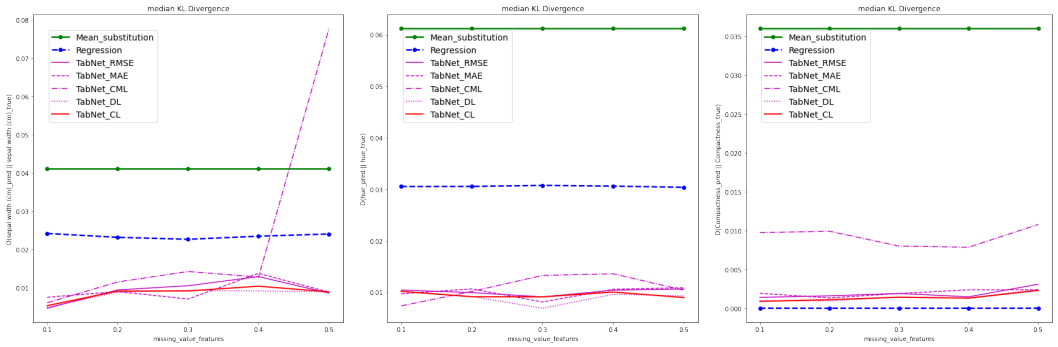
Variable Ratio of Missing Value Features

The second experiment examines the influence of the ratio of MV features in the datasets. I.e. instead of only generating MVs only in the feature of interest, they are now generated in a certain percentage of randomly chosen additional features. 10 versions of MV datasets were created for each of the three datasets and $\text{mv}_{\text{features}}$ ratios of 10%, 20%, 30%, 40%, 50%. The imputation, subsequent clustering and analysis were done just as in experiment 4.4.

Just like for experiment 4.4 the median KL divergences for all datasets and imputation strategies were plotted (cf. Figure 4.6). For this setup, both mean substitution, as well as regression imputation, could achieve constant performance. Mean substitution is completely independent of other feature values. However, regression imputation being almost independent of MVs in other features is surprising. All TabNet loss functions also seem to be rather stable with a rather small variance. Only CML showed a large increase in KL divergence for a higher amount of features with MVs. Once more CML underperformed for the large dry bean dataset. TabNet has the lowest KL divergences for the wine and iris datasets and appears to do very well on lower dimensional versions of the dataset.

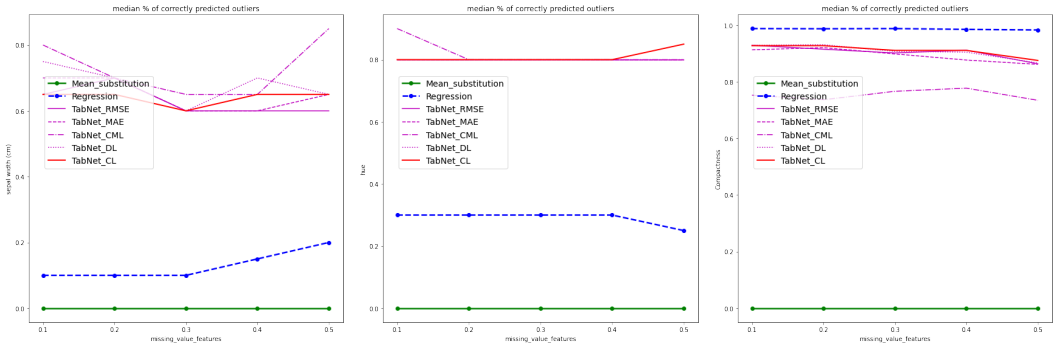
Similar results were obtained for the outlier scores (cf. figure 4.7). All approaches have a stable performance independently of missing values in other features. CML again has worse values for the dry bean dataset. TabNet is doing best on the iris and wine data, while regression is constantly best for dry beans prediction.

Finally, the ARI scores (cf. Figure 4.8) appear to vary strongly depending on the datasets. All imputation strategies perform well on the iris dataset,



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.6: Variable $mv_{features}$ - Median KL divergence of all three datasets after mean substitution, regression imputation and TabNet imputation.

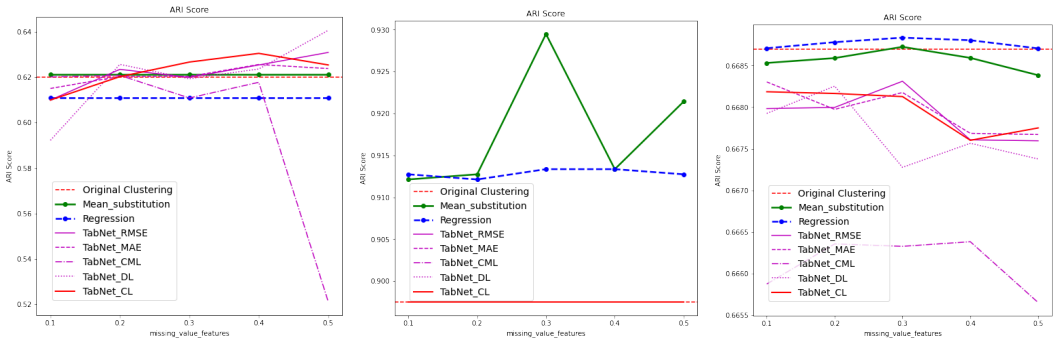


(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.7: Variable $mv_{features}$ - Median outlier scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

being close to the performance of the original clustering. Only TabNet using CML obtained a very low score for higher values of $\text{mv}_{\text{features}}$. TabNet could recreate the original clustering with every loss function independently of $\text{mv}_{\text{features}}$. The other two approaches reached even higher ARI scores. For the dry bean data, mean substitution and regression imputation performed well, only TabNet had lower values for all loss functions and slightly decreased with growing $\text{mv}_{\text{features}}$ values. Again, CML has significantly lower scores than all other approaches.

Discussion of results: experiment 4.4 results suggest, that all three strategies are almost independent of the number of features with missing values. Just like in the first experiment, TabNet could reach very good outlier



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.8: Variable $mv_{features}$ - Median ARI scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

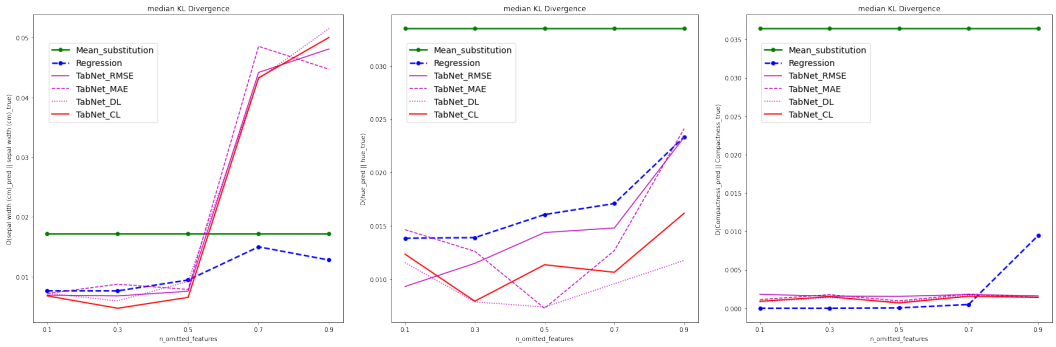
scores for small datasets and does however not outperform regression imputation. Also, the experiments support the conclusion, that the use of different loss functions apart from CML does not have a clearly distinguishable impact on the scores.

Research Question 2

How does the number of dimensions influence the cluster/outlier results?

In this experiment, each dataset version has mv=20% missing values. Additionally, a percentage of randomly selected features gets deleted before applying the imputation strategies. The respective parameter $n_{omitted\_features}$ is varied from 10%, 30%, 50%, 70% up to 90%. Again, all three datasets are used with varying filling methods. The Cluster-Mean-Loss is not being considered, as it is not meaningful in lower dimensions of the dataset (the calculation using the distance to the cluster centres of the complete dataset).

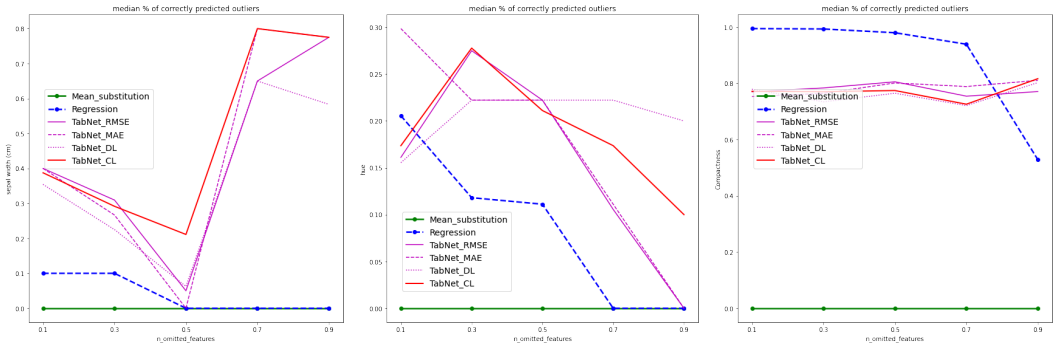
The resulting median KL divergence values are shown in figure 4.9. As is to be expected, mean substitution has a relatively high divergence from the original distribution for all datasets. Regression and TabNet imputation perform similarly in most cases. However, the TabNet approach is suffering for the iris dataset when more than 50% of the features are omitted. For the wine data, TabNet hast consistently lower divergences than regression. Both methods behave similarly with increasing values for $n_{omitted\_features}$. Lastly, both approaches perform well on the dry beans dataset. Only regression appears to perform worse for very high rates of omitted features.



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.9: Variable $n_{omitted\_features}$ - Median KL divergence of all three datasets after mean substitution, regression imputation and TabNet imputation.

The outlier scores (cf. Figure 4.7) imply slightly different results. Mean substitution scores are zero per cent for all datasets. Regression imputation performs well only for low ratios of omitted features and only for wine and dry beans data. While TabNet achieves good results for lower dimensional iris dataset versions, the outlier results behave reversely for the wine data. TabNet achieves a consistently high rate for the dry bean dataset. It is, however still outclassed by regression imputation for dataset versions with fewer omitted features.

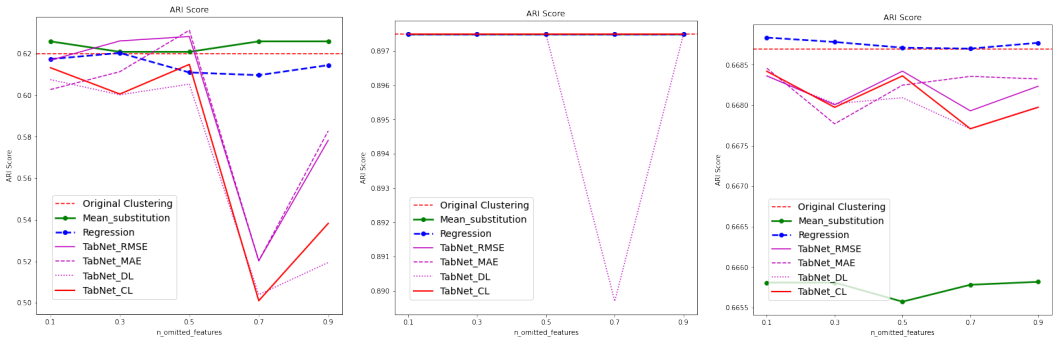


(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.10: Variable $n_{omitted\_features}$ - Median outlier scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

Lastly, the ARI scores (cf. Figure 4.11) of mean substitution and regression

are consistently close to the original clustering for the iris and wine datasets. TabNet is matching these values only for lower rates of omitted features. For high rates around 70%, the ARI of TabNet plummets but appears to improve for even higher rates. On the dry bean dataset, all approaches show fairly consistent behaviour. Mean substitution is performing the worst, while TabNet is close to the original clustering and regression matches it almost perfectly. The absolute values are, however, almost identical for all approaches.



(a) iris dataset (b) wine dataset (c) dry bean dataset

Figure 4.11: Variable $n_{omitted\_features}$ - Median ARI scores of all three datasets after mean substitution, regression imputation and TabNet imputation.

Discussion of results: The results show, that the TabNet approach can compete with regression imputation in many cases. Though it has to be noted, that the training of TabNet takes significantly more time than using regression. For the large dry bean dataset, TabNet can however outperform regression when only very few features are available to reason from. The impact of the chosen loss function barely seems to have an impact on the performance scores. Furthermore, there is a clear correlation between the scores for the iris datasets. The KL divergences and outlier scores of TabNet are increasing with missing features. This might seem like a contradiction. It is however possible, that TabNet correctly predicted the outlierness, and yet imputed an incorrect value. For example, if a value of 5 is counted as an outlier in the original dataset and TabNet predicts a value of 10, it would be correctly classified as an outlier, despite the value being far off the ground truth. This is reflected by the increasing KL divergences. At the same time, the same incorrectly predicted data points could be the reason for the shrinking ARI. In the case of the wine dataset, increasing KL divergence values are going along with decreasing outlier performance, while the ARI score is almost constant. One explanation would be, that TabNet falsely predicted outlier values to be

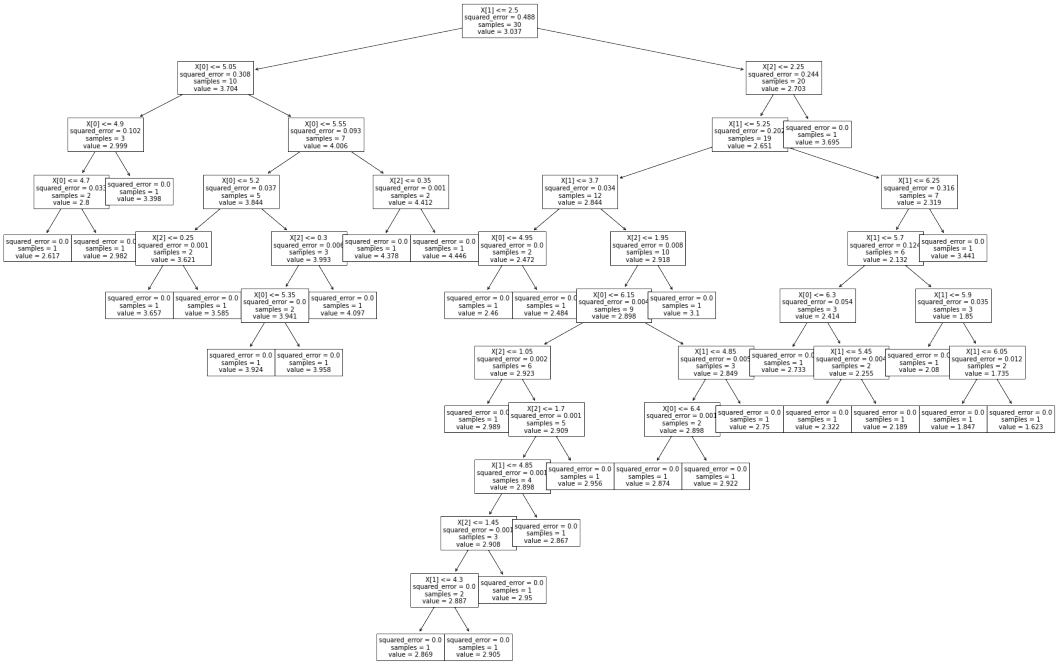


Figure 4.12: Decision tree based on TabNet predictions for missing values in the iris dataset.

close to the mean. Mean values seem to be a good estimation for obtaining a decent ARI. The observation is consistent with mean substitution performing well in regards to the ARI score for most of the experiments and datasets. Regarding RQ2 1.3, the results indicate, that the performance at different ratios of omitted features strongly depends on the imputation strategy and datasets. No one strategy has consistently good score values in all cases.

Research Question 3

How is it possible to make model-based filling strategies explainable?

Finally, for RQ3 1.3, the concept was extended to train a decision tree from one of the earlier trained TabNet model predictions. The regression tree is implemented using the sklearn library with default parameters. The idea is, to explain the decision-making process of the TabNet model in a way, that is easily understandable by humans.

For the training of the tree, the first dataset chosen is the iris dataset and the best-performing model with a MV ratio of 15%. The custom loss is used

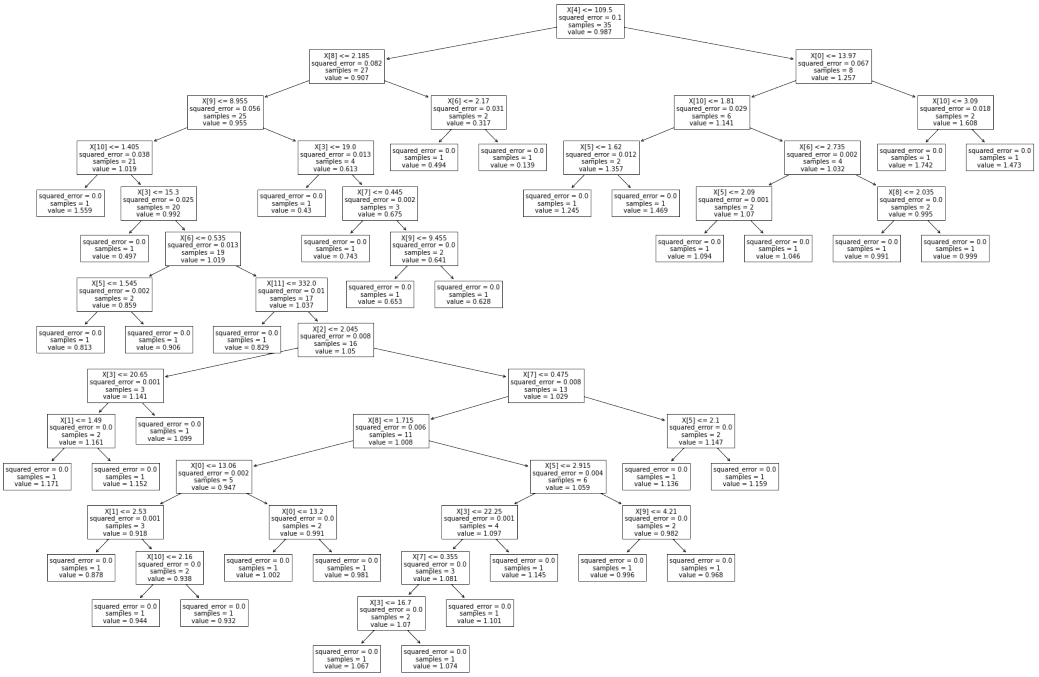


Figure 4.13: Decision tree based on TabNet predictions for missing values in the wine dataset.

for the TabNet training process. The training set for the tree only contains the predicted values during the imputation step.

The resulting tree can be seen in figure 4.12. This tree is fairly large considering the low dimensionality and sample size of the dataset. Yet, it is still small enough, that a human with knowledge about the context could comprehend it. However, the viability of the decision-making is not proven. Ideally, the tree would need to be verified by an expert in the field of iris flowers.

When training a tree for a dataset with more features (cf. Figure 4.13) the resulting tree is growing in depth. At this point, it might become cumbersome to trace. Furthermore, the verification by an expert could become increasingly hard. The situation worsens drastically for the large dry bean dataset. The tree derived from the imputed values is too large to properly fit into one figure.

Discussion of results: the approach presented is just an example showcasing the possibility of training a decision tree on the data generated before. While this can help to increase explainability for smaller datasets, it is not yet viable for high-dimensional datasets.

Chapter 5

Conclusion

The presented work aims to fill the research gap related to the optimisation of downstream clustering of missing value datasets by applying suitable machine-learning-based imputation strategies. A concept was developed based on TabNet [1] to impute missing values utilizing multiple different loss functions. The performance of the imputation as well as a subsequent clustering was then compared to that of other filling strategies.

The concept was first tested in two experiments with variable values for the ratio of missing values as well as the ratio of features containing missing values. The goal of the experiment was to investigate, how the clustering performance can be improved with model-based filling strategies (cf. RQ1 1.3). While the proposed concept performed well on most datasets and missing value ratios regarding distribution and outlier preservation, the results do not imply a reliable improvement of the ARI scores in comparison to regression imputation and a clustering on the original data (ground truth).

In a second experiment, the same approach was utilized to examine the impact of missing dimensions in the original data (cf RQ2 1.3. For three datasets, a variable percentage of randomly chosen features was omitted before imputing the missing data of the feature of interest and subsequently clustering the reconstructed dataset. The developed concept was able to generate consistent and promising outputs, especially for the large dry bean dataset (low KL divergences, high outlier and ARI scores even for low dimensional data). The results were less consistent for smaller datasets, as high ratios of omitted features could cause either good or bad outlier scores, depending on the dataset.

Finally, an approach is presented to increase the explainability of the model-based imputation (cf. RQ3 1.3). A regression tree was trained on data imputed

by the developed TabNet concept. Provided, that an expert verifies the resulting tree, it can be argued, that this approach helps to interpret the TabNet decision-making process for small datasets. For larger sets of data, however, the tree will grow to a degree that is hardly comprehensible by humans.

Chapter 6

Future Work

The concept introduced can be seen as a basic foundation to research on the impact of MV imputation on clustering algorithms. However, some aspects have not been covered or could be the subject of further research:

- The datasets used had their missing values artificially generated (MCAR). In the future, the concept should additionally be tested with real-world missing datasets that include missing values in a MAR or MNAR fashion.
- The experiment chapter 4.1 establishes the KL divergence as an index used to evaluate distribution preservation and as a loss function for TabNet. KL divergence is not a symmetric function and could be integrated in both directions, i.e. $D(X_{pred} \,||\, X_{true})$ as well as $D(X_{true} \,||\, X_{pred})$. With the current implementation of $D(X_{pred} \,||\, X_{true})$, divergences closer to the mean are associated with higher KL divergence values. This punishes approaches like mean substitution and encourages TabNet to also predict outlier values. However, it could also lead to incorrect predictions in the mean region, as TabNet predictions close to the mean increase the KL divergence loss. A future adaption could implement a symmetric function (such as the Jeffrey Divergence[6]) or use the KL divergence in both directions.
- To properly benchmark the performance of the approach, it is necessary to include a wider range of imputation techniques, such as multiple imputation and other machine learning approaches.
- The current concept does not make use of the TabNet feature importance. Evaluation of the attention matrices of the trained models in comparison to experiments realted to RQ2 1.3 could give deeper insights into the impact of omitted features in datasets.

- As already discussed in 4.4, using decision trees on the imputed data can be complex and hard to comprehend. This approach could be extended, for example by using the TabNet feature importance to detect the most salient features and base the decision tree on the features only.
- The statistical evidence for the effectiveness of the approach could be improved in a simple way. More datasets could be randomly created and more models trained. This was not done in this work due to performance reasons.

List of Figures

List of Tables

Bibliography

[1] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning (v5). In *arXiv Computer Science*, 2020.

[2] A. N. Baraldi and C. K. Enders. An introduction to modern missing data analyses. *Journal of School Psycholology*, 48(1):5–37, 2010.

[3] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[4] A. G. de Brevern et al. Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinformatics*, 5(114), 2004.

[5] M. C. P. de Souto et al. Impact of missing data imputation methods on gene expression clustering and classification. *BMC Bioinformatics*, 16(64), 2015.

[6] M. Deza and E. Deza. *Encyclopedia of Distances*. Springer Dordrecht Heidelberg London New York, 2009.

[7] D. Dua and C. Graff. *UCI Machine Learning Repository*. University of California. School of Information and Computer Science, 2017.

[8] A. Rogier et al. Review: A gentle introduction to imputation of missing values. In *Journal of Clinical Epidemiology*, volume 59, pages 1087–1091, 2006.

[9] C. C. Aggarwal et al. Fast algorithms for projected clustering. *ACM SIGMOD Record*, 22(2):61–72, 1999.

[10] D. Li et al. Towards missing data imputation: a study of fuzzy k-means clustering method. In *Proceedings of 4th international conference of rough sets and current trends in computing (RSCTC)*, page 573–579, 2004.

[11] D. Wang et al. Effects of replacing the unreliable cdna microarray measurements on the disease classification based on gene expression profiles and functional modules. *Bioinformatics*, 22(23):2883–2889, 2006.

[12] F. Pedregosa et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] I. Ullah et al. Explaining deep learning models for tabular data using layer-wise relevance propagation. *Applied Sciences*, 12(1), 2022.

[14] J. Luengo et al. On the choice of the best imputation methods for missing values considering three groups of classification methods. In *Knowledge and Information Systems*, volume 32, page 77–108, 2012.

[15] J. Tuikkala et al. Missing value imputation improves clustering and interpretation of gene expression microarray data. *BMC Bioinformatics*, 9(202), 2008.

[16] J. Yoon et al. Vime: Extending the success of self- and semi-supervised learning to tabular domain. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, page 11033–11043, 2020.

[17] M. Celton et al. Comparative analysis of missing value imputation methods to improve clustering and interpretation of microarray experiments. *BMC Bioinformatics*, 11(15), 2010.

[18] M. Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231, 1996.

[19] Mustafa Alabadla et al. Systematic review of using machine learning in imputing missing values. In *IEEE Access*, volume 10, pages 44483–44502, 2022.

[20] P. J. García-Laencina et al. Pattern classification with missing data: a review. In *Neural Computing and Applications*, volume 19, page 263–282, 2010.

[21] R. Agarwal et al. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems*, volume 34, pages 4699–4711. Curran Associates, Inc., 2021.

[22] R. Agrawal et al. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 27(2):94–105, 1998.

[23] S. Datta et al. Clustering with missing features: a penalized dissimilarity measure based approach. *Machine Learning*, 107:1987–2025, 2018.

[24] S. Goil et al. Mafia: Efficient and scalable subspace clustering for very large data sets. *Technical Report No. CPDC-TR-9906-010*, NWU, 1999.

[25] S. Zhang et al. Missing value imputation based on data clustering. In *Transactions on Computational Science I*, volume 4750 of *Lecture Notes in Computer Science*, pages 128–138. Springer, Berlin, Heidelberg, 2008.

[26] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(2):179–188, 1936.

[27] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979.

[28] M. Koklu and I. A. Ozkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture*, 174:105507, 2020.

[29] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[30] A. Lithio and R. Maitra. An efficient k-means-type algorithm for clustering datasets with incomplete records. *Statistical Analysis & Data Mining*, 11(6):296–311, 2018.

[31] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 1987.

[32] A. Purwar and S. K. Singh. Dbscani: Noise-resistant method for missing value imputation. *Journal of Intelligent Systems*, 25(3):431–440, 2016.

[33] D. Rubin. *Multiple imputation for nonresponse in surveys*. Wiley Series in Probability and Statistics. Wiley, 1987.

[34] T. Thomas and E. Rajabi. A systematic review of machine learning based missing value imputation techniques. In *Data Technologies and Applications*, volume 55, page 558–585, 2021.