Data quality enhancement: Missing values and outliers

A case study for a computational biology framework

T.P.A. Beishuizen (0791613)
Biomedical Engineering - Computational Biology
Computer Science - Data Mining
Eindhoven, University of Technology
Email: `t.p.a.beishuizen@student.tue.nl`

September 12, 2018

Contents

1 Introduction

Many biomedical datasets have been created to use for expansion of biomedical knowledge and improvement of healthcare. Biomedical data is a generalizing term that describes multiple data types[1]. Examples of biomedical data are micro-array data[2], mass spectrometry data[3, 4] and nuclear magnetic resonance data[5], but also clinically derived data[6, 7] and survey data[8]. From a bio-informatics perspective these biomedical data types vary significantly[1] and therefore extracting information out of biomedical data is not a trivial task. A framework for biomedical data analysis can help guiding biomedical engineers in their process of information extraction from their biomedical datasets. The framework can provide different options in processing the data, taking into account common dataset issues[9, 10, 11] and approaches to reach a certain goal[12, 13]. Currently available frameworks however mainly focus on the integration of databases[14, 15], are made specifically for one research area[16, 17, 18] or are limited to one specific type of analysis[19]. A framework that combines database integration, multiple research areas and multiple types of data analysis would be very beneficial for biomedical engineers, guiding them through their biomedical data analysis projects.

The quality of data should be as high as possible. The gathered biomedical data usually contains of several erroneous data entries, however, due to numerous reasons possibly causing those errors. A significant part of the data may be wrong due to a expected error rate, which differs greatly between dataset. For example, several works discussed this expected error rate and showed estimations from 0.3%[20] to up to 26.9%[21] of the complete dataset. Aside the presence of possible errors values in the datasets could be missing, for example due to patients not showing up or because of mistakes by the medical staff. Both missing values and errors hinder the data set quality significantly and for numerous datasets removing or replacing them would significantly help the increase the quality of the research. Several works already focused on error detection, also known as anomaly detection[22, 23, 24], and the handling of missing values[25, 26, 27]. A biomedical data analysis framework would benefit from the possibility to remove anomalies as well as remove or replace missing values. Therefore the research goal for this project is *to evaluate the performance of anomaly detection and missing value handling methods and make a choice on which methods should be added to the framework.* In this document we therefore present several of those algorithms and test their quality.

2 Background

2.1 Datasets

A total of four datasets were gathered for the analysis of missing values. These datasets all differ in the number of samples, the number of missing values and the origin of the data, and therefore represent a wider range of datasets. The four datasets are all separately explained and their specifications are summarized together (Table 1).

- *Heart Attack Echocardiogram dataset*
  For this dataset patients were selected that at one point in their lives suffered a heart attack and survived. A prediction model can be made that models the months of survival after the heart attack as the prediction variable. There are 9 features for 108 samples present, with a total of 97 missing values. Every feature but two have at least one missing value, but by far the most missing values are located in the feature ”alive-at-1” (still alive after 1 year) with 58 missing values and 54% of total number of samples. This high number of missing values can be partially explained because of lack of documentation, partially because no year had passed, yet, after data distribution[28].

- *Hepatitis dataset*
  The mortality rate of hepatitis was tested using 19 features over 155 samples. Two class types are distinguished as ”died” and ”lived” with thirteen boolean and six numeric attributes.

The total number of missing values is 167 spread over 15 classes, with the feature labeled ”histology” having the majority of 67 missing values. Two previous articles were written, using this dataset as an example for their methods[29, 30].

- *Primary Biliary Cirrhosis dataset*
  Patients were classified in three groups in this dataset: alive, transplanted and dead. 18 features are present in the dataset: one date-specific, two categorical, five boolean and ten numeric attributes. The dataset consists of 1945 instances and 1133 missing values. The missing values are distributed over six features, the feature ”serum_cholesterol” having 821 and the other five around 60 missing values. The missing values are noted as most likely not missing completely at random[31].

- *Cervical Cancer dataset*
  This dataset is consisting of 858 samples and is based on the increased risk of subjects becoming cervical cancer patients. Since a risk can be fairly subjective, four aspects were used to determine whether an increased risk is present or not: two doctors, a cytologic examination and a bioptic examination. For analysis a new increased risk factor was made by giving each subject a value in $[0, 4]$, implying the number of times this subject has been classified as an increased risk subject. The dataset consists of 31 features, 24 boolean and numeric features originated from a survey and 7 boolean and numeric features originated from the hospital. A total of 3622 values are missing of which 787 times the hospital features ”STDs: Time since first diagnosis” and ”STDs: Time since last diagnosis” caused by no sexually transmitted disease (STD) ever being present. All other missing values originate from subjects not answering survey questions, the ”Age” feature being the only survey question that was always answered[32].

Table 1: A schematic overview of the four datasets

Dataset focus	**Features**	**Samples**	**Classes**	**Total Missing values (%)**	**Most missing in single feature (%)**	**Features with missing values**	**Remarks**
Heart attack	9	108	53	8.29%	43.85%	7	- Output in months (Regression preferred) - Values are missing due to time constraints —
Hepatitis	19	155	2	5.67%	43.22%	15	- Used in previous published studies —
Cirrhosis	18	1945	3	3.24%	42.21%	6	- Missing not completely at random —
Cervical Cancer	28	858	5	15.07%	91.72%	26	- Missing values mainly from the survey part - Output comprised of four separate indicators

2.2 Missing Values

Numerous reasons can cause entries to be missing from a dataset. A sample can be missing, corrupted or contaminated, a measurement malfunction can occur, a patient can fail to show up at a scheduled meeting or not respond to a survey. These missing entries are different on both origin and possibly also on randomness. This also indicates that techniques differ in effectiveness on missing values caused by separate reasons. Therefore the different types of missing values will be explained, as well as techniques to cope with these missing values.

Since there are multiple steps to be taken for gathering data, a problem during an earlier step can generate more values to be missing. When a patient fails to come to the hospital for a second round of tests all values for the second test are absent, whereas a nurse accidentally skipping to note the weight of a patient will only cause the absence of one value. A difference is therefore made between item and unit non-responses. An item response is a single missing value in a data set and a unit non-response corresponds to a series of missing values.[33]

Another aspect of missing values is its randomness. A missing value can occur completely random without any relation with other values, however is not uncommon to have some relation with another value. Three different types of randomness are defined to explain this. The first type is *missing completely at random* (MCAR)[25, 26, 27], and means that missing entries have no relation with any part of the data. This means removing samples with missing values should not create any bias in the resulting dataset. An example would be accidentally dropping a blood vial, not being able to report the blood values. The opposite of MCAR is *missing not at random* (MNAR)[25, 27], also known as *non ignorable* (NI)[26]. MNAR means that values are missing are linked to the feature they are missing from, which makes it very difficult to find the relation between the missingness and a feature. An example of MNAR is patients that are unsuccessfully treated for a disease, and are therefore less likely to return for future tests, creating a bias in the values about disease severity. In between MCAR and MNAR is *missing at random* (MAR)[25, 26, 27]. MAR indicates missing entries that have no relation with the feature they are missing from, but have a relation with other features in the data sets. These entries can be estimated using the relation to better identify a possible value. Examples are specific diagnostic tests that are done for patients of a specific disease for increased health risks, whereas other (healthy) subjects were not tested[27]. Whereas MCAR values are relatively easy and MNAR values are almost impossible to cope with, MAR values can be treated properly with complex approaches so proper values are imputed in those cases. Therefore many imputation techniques are created on basis of the MAR assumption.

The techniques of coping with missing values are based on three principles. The first principle is list deletion (LD)[25, 27], deleting samples, features or both to create a smaller dataset without missing values. This is the most common way to treat missing values as it is quick and easy. If enough samples are present and there is not too much bias between samples including and excluding missing values, this is the way to go. If the number of samples is limited or a bias is present in the missing value samples, value imputation becomes more interesting. In this case valuable information will not be omitted and the possible bias is taken into account. Imputation can be split into two different types, single imputation (SI)[25, 26, 27] and multiple imputation (MI)[25, 33, 34]. Thirdly, instead of removing or adding values, the dataset can be directly used in a model, disregarding missing values in the model[35].

Since methods handling missing values can usually be best explained by showing example algorithms, several pseudo-algorithms were created for visual clarity. Variables used in these algorithms include:

- F: A list of present features. A subset x of F is named F_x and a when talking about a single feature, f is usually used.
- S: A list of all samples in a dataset. A subset x of S is named S_x and a single sample is usually called s.
- X: A matrix that contains all sample values for every feature. It is an n by m matrix with n being the number of samples and m being the number of features. If the values of a specific feature f or a subset of features F_x are used this is written as X_f and X_{F_x} respectively, hence the column f or columns F_x are selected. Similarly for X_s and X_{S_x} the row s or rows S_x respectively are collected from X.
- y: A vector that contains all class labels for every sample. It is a vector of size n with n being the number of samples.
- *missing_values()*: This simple function is made to illustrate whether a missing value is present in the values that are given.

2.2.1 List Deletion

List deletion (LD) is quick and easy manner to perform for coping with missing values. Samples or features are removed when they consist of at least one missing value until no more missing values are present. Three difference approaches are explained: *complete case analysis* (CCA), *available case analysis* (ACA) and *weighted case analysis* (WCA).

- *Complete case analysis*
  CCA is the most basic LD approach. CCA removes all samples that have missing values and therefore prepares only complete samples for future analysis (Algorithm 2). The dataset should consist of a reasonable number of complete samples to leave enough data for analysis. Also data should be MCAR, as removing cases that have a dependency with certain features can create a bias. CCA is most used as a way of removing missing values, even though not every time MCAR is available[27, 33, 35, 25]. In those cases CCA is proven to be biased[26].

Algorithm 1 Complete Case Analysis

```
procedure CCA($X$)
    $S \leftarrow$ range(#rows(X))                    ▷ Create a list of all samples
    for $s$ in $S$ do                                 ▷ For all samples in $S$
        if $missing\_values(X_s)$ is True then        ▷ find out if missing values are present
            remove $X_s$ from $X$                     ▷ Remove the sample with missing values
        end if
    end for
    return X
end procedure
```

- *Available case analysis*
  ACA tries to preserve more samples by only using useful features. ACA first makes a selection on relevant features in the dataset. Those features are selected by hand so they will be present in future analysis. A second approach is to remove features for which the percentage of missing values is higher than a threshold α. After selecting those features, samples without missing values in the relevant feature set are kept, all others are removed (Algorithm 2). This is similar to CCA, but will preserve more samples if several features have significantly more missing values. For this LD, MCAR would be desired. However if either the MAR values or the features it has a relation with are removed, there will be no relation between the MAR values and the dataset any more and therefore no bias will be created[27, 25, 33, 35]. Often though, this will not be the case, as the dependency is on either a multitude of features and the dependent feature is important for the research.

Algorithm 2 Available Case Analysis

```
procedure ACA($X$, $\alpha$)
    $F \leftarrow$ range(#columns(X))                 ▷ Create a list of all samples
    for $f$ in $F$ do                                 ▷ For all features in $F$
        if $perc\_missing\_values(X_f) > \alpha$ then ▷ $perc\_missing\_values$ finds out the percentage of missing values
            remove $X_f$ from $X$                     ▷ Remove the feature with too many missing values
        end if
    end for
    return CCA(X)
end procedure
```

- *Weighted case analysis*
  A third type of LD is WCA. WCA tries to overcome bias from CCA by assigning weights to

every value (Algorithm 3). If a sample is removed, the nearest complete neighbour is searched for (Algorithm 3 - row 5). This closest complete sample can be found by using one, multiple or available features. The complete sample is then given a higher weight corresponding to the number of closest omitted samples. A simple approach for giving a higher weight is simply by adding an additional complete case to the existing complete cases, choosing the one that is closest to the incomplete case. This technique tries to remove the bias as much as possible by maintaining the distribution of the samples. This technique specifically does not assume MCAR values, but MAR values[27, 25]. Also this technique is specifically useful for unit non-responsiveness for its simplicity to cope with multiple missing values in one sample[33].

Algorithm 3 Weighted Case Analysis

```
 1: procedure WCA($X$)
 2:     $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
 3:     for $s$ in $S$ do                                        ▷ For all samples in $S$
 4:         if $missing\_values(X_s)$ is True then              ▷ find out if missing values are present
 5:             $s_{nn}$ = $nearest\_complete\_neighbour(X, s)$ ▷ Find the nearest complete neighbour of $s$
 6:             $X_s \leftarrow X_{s_{nn}}$                      ▷ Change the values of $s$ to the values of $s_{nn}$
 7:         end if
 8:     end for
 9:     return X
10: end procedure
```

Whereas List Deletion is used most regularly and is quickest and easiest of all the missing value handling techniques, it is often criticized for its drawbacks. If the data is not MCAR, bias will most often be created, even in ACA and WCA[26]. Also the removal of samples is not always desired when not many samples were available at the start[27, 35, 25]. All things considered LD is a quick fix when needed. For better understanding of the data however, other approaches seem more appropriate.

2.2.2 Single Imputation

An intuitive approach of coping with missing values and not wanting to delete entire samples or features is to impute an appropriate entry in its place. If the imputed entry is a proper representation of all possible entries, the possibly valuable information within the sample or feature will be useful for data analysis. Imputing a replacement value for the missing value is called Single Imputation (SI). Several different approaches for SI are known and used.[27, 25, 35]

- *Missing indicator method*
  The simplest way of SI is missing indicator method. This approach imputes a generic variable for every missing value of a feature, for example imputing 0 for heart rate for every missing measurement. Aside from that, it also creates an additional boolean feature that indicated whether a value was missing or not (Algorithm 4). When using a generic feature no additional information is added to the data and no assumptions are made, which makes this method less dependent on assumptions[36]. Another possibility is to combine the missing indicator method with other imputation methods, for example the mean imputation, so from the dataset missing values can still be traced back. Any calculations or modelling done with the feature has an important dependency to the newly created missing value feature, though. Also if data is MAR and not MCAR, bias may be present due to worse representation[25].

Algorithm 4 Missing Indicator Imputation

```
procedure MISSING_INDICATOR_IMPUTATION($X$, $a$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $f$ in $F$ do                                       ▷ For all features in $F$
        if $missing\_values(X_f)$ is True then              ▷ find out if missing values are present
            append $X_{f'}$ to $X$                          ▷ Create a missing indicator feature for $f$
            for $s$ in $S$ do                               ▷ For all samples in $S$
                if $missing\_values(X_{s,f})$ is True then  ▷ Find out if missing
                    $X_{s,f} \leftarrow a$                  ▷ Impute generic value for missing vlaue
                    $X_{s,f'} \leftarrow True$              ▷ Assign true for missing indicator value
                else
                    $X_{s,f'} \leftarrow False$             ▷ Assign false for missing indicator value
                end if
            end for
        end if
    end for
    return X
end procedure
```

- *Mean/Median/Mode imputation*
  The values of a feature usually follow a certain distribution. In a continuous distribution values are most probable to be the mean of the distribution. Therefore if a value should be imputed, the mean would be the most logical imputation (Algorithm 5). For ordinal features this would be the median and for categorical the mode[27, 35, 26]. With this imputation the distribution centre stays the same, but the standard error is lowered due to addition of new values. This creates a bias for the variance of the feature. Also the data may not be MCAR but MAR, so additional bias would be created by the present values not being a representation of the total number of values[25, 36].

Algorithm 5 Mean Imputation

```
procedure MEAN_IMPUTATION($X$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $s$ in $S, f$ in $F$ do                             ▷ For all samples in $S$ and features in $F$
        if $missing\_values(X_{s,f})$ is True then          ▷ find out if missing values are present
            $X_{s,f} \leftarrow mean(X_f)$                  ▷ Assign the mean of the values for feature $f$
        end if
    end for
    return X
end procedure
```

- *Hot deck imputation*
  Whereas the application of hot deck imputations differs somewhat[35, 26, 27], the main concept is based on randomly picking a value from a set of possibilities. The set can be all known possible values for the feature, known from other samples in the dataset (Algorithm 6). Another approach would be to assume the feature is distributed a specific way (for example normally distributed) and randomly pick a value using the values from that distribution. A third possibility is to combine this with kNN imputation and randomly pick a value from the k nearest neighbours. Randomly picking a value instead of calculating a fixed value will not unjustifiably reduce the variance of the distribution. On the other hand, it also makes the estimation to be more imprecise, possibly creating bias this way[27].

Algorithm 6 Hot Deck Imputation

```
procedure HOT_DECK_IMPUTATION($X$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $s$ in $S, f$ in $F$ do                              ▷ For all samples in $S$ and features in $F$
        if $missing\_values(X_{s,f})$ is True then           ▷ find out if missing values are present
            $X_{s,f} \leftarrow random(X_f)$                 ▷ Assign a random choice of the values for feature $f$
        end if
    end for
    return X
end procedure
```

- *Multi-variate regression imputation*
  Instead of only using information of the feature that has missing values, other features can be used to predict the missing value replacements. Especially when multicollinearity is present, a prediction of the actual missing value can be computed quite precisely. A multivariate regression model can be made by using features without missing values as input and the features with missing values as output. For example, with linear regression a function can be made between the input and output, trained by all available values and used for predicting all missing values (Algorithm 7)[37]. This approach is especially suited for MAR values, in which the absence of feature values is related with other features. The regression imputation should be used with care, as it is unwise to be used if afterwards a similar regression technique is used for further analysis[25].

Algorithm 7 Multivariate Regression Imputation

```
procedure REGRESSION_IMPUTATION($X$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $f$ in $F$ do                                        ▷ For all features in $F$
        if $missing\_values(X_f)$ is True then               ▷ Find out if missing values are present
            create $regressor$                               ▷ Initialize a regressor
            fit regressor with $X_{F\setminus\{f\}}, X_f$     ▷ Fit the regressor by using other features as input
and the feature with missing values as output
            for $s$ in $S$ do                                ▷ For all samples in $S$
                if $missing\_values(X_{s,f})$ is True then   ▷ If a missing value is present
                    $X_{s,f} \leftarrow$ with $regressor$ predict $X_{s,F\setminus\{f\}}$   ▷ Impute a predicted value
                end if
            end for
        end if
    end for
    return X
end procedure
```

- *k-Nearest Neighbour imputation*
  Similarly to the multi-variable regression imputation, the k nearest neighbour (kNN) imputation can be used to predict missing values more precisely. This approach also makes use of features that do not have missing values. Similarly to kNN classification and regression it takes the values of the k samples that are closest to the sample with the missing value and combines those into a replacement for the missing value (Algorithm 8). This is also suited for MAR values, as relations with other features are used for prediction[35, 26]. For kNN imputation, further data analysis steps must also be taken with care as additional nearest neighbour analysis techniques may give biased results[25].

Algorithm 8 k Nearest Neighbour Imputation

```
procedure kNN_IMPUTATION($X$, $k$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $f$ in $F$ do                                       ▷ For all features in $F$
        if $missing\_values(X_f)$ is True then              ▷ Find out if missing values are present
            create $kNN\text{-}model(k)$                    ▷ Initialize a kNN-model
            fit kNN-model with $X_{F\setminus\{f\}}, X_f$   ▷ Add cases to the kNN-model
            for $s$ in $S$ do                               ▷ For all samples in $S$
                if $missing\_values(X_{s,f})$ is True then  ▷ If a missing value is present
                    $X_{s,f} \leftarrow$ from $kNN\text{-}model$ extract $X_{s,F\setminus\{f\}}$   ▷ Extract a value with kNN
                end if
            end for
        end if
    end for
    return X
end procedure
```

- *Worst Case Analysis*
  When samples in a dataset has missing values for an important feature, a safe choice can be made by replacing them with the best or worst possible value (Algorithm 9). An example would be if a feature depicts the severity of a disease. If false positives on severe patients are more justifiable than false negatives, all missing values can be replaced by the highest severity value. An advantage of this technique is that boundaries can be made with the analysis and for the example false negatives would occur less. Off course a disadvantage would be that the outcome will not be as precise, due to the imprecise assumption. Also not all data analysis only need a bounds and rather have strict models[36, 27].

Algorithm 9 k Nearest Neighbour Imputation

```
procedure WORST_CASE_IMPUTATION($X$, $worst\_cases$)
    $S \leftarrow$ range(#rows(X))                          ▷ Create a list of all samples
    $F \leftarrow$ range(#columns(X))                       ▷ Create a list of all features
    for $s$ in $S, f$ in $F$ do                             ▷ For all samples in $S$ and features in $F$
        if $missing\_values(X_{s,f})$ is True then          ▷ find out if missing values are present
            $X_{s,f} \leftarrow worst\_case(X_f)$           ▷ Assign the worst case of feature $f$
        end if
    end for
    return X
end procedure
```

Single imputation consists of the quickest and easiest types of approach to replace a missing value. The approaches mostly are very straightforward and protect samples from being removed from the dataset by deletion[27, 26]. Especially for small sampled datasets, this would be a quick outcome to not remove too many samples. Using SI methods should be done with care, since not all are fit to be used with MAR values. Also, except for hot deck imputation, all SI techniques cause the feature to have a lower variance that can cause bias in further research on the dataset[25].

2.2.3 Multiple Imputation

Multiple imputation (MI) creates additional datasets, for the dataset with missing values. These new datasets will have different values imputed, either from some kind of distribution similar to hot deck imputaion or by using a different algorithm. After doing multiple hot deck imputations, analysis is done on all different datasets and afterwards the results are combined (Figure 1). Three aspects are important: The value impuation, the creation of multiple datasets and the analysis before merging.

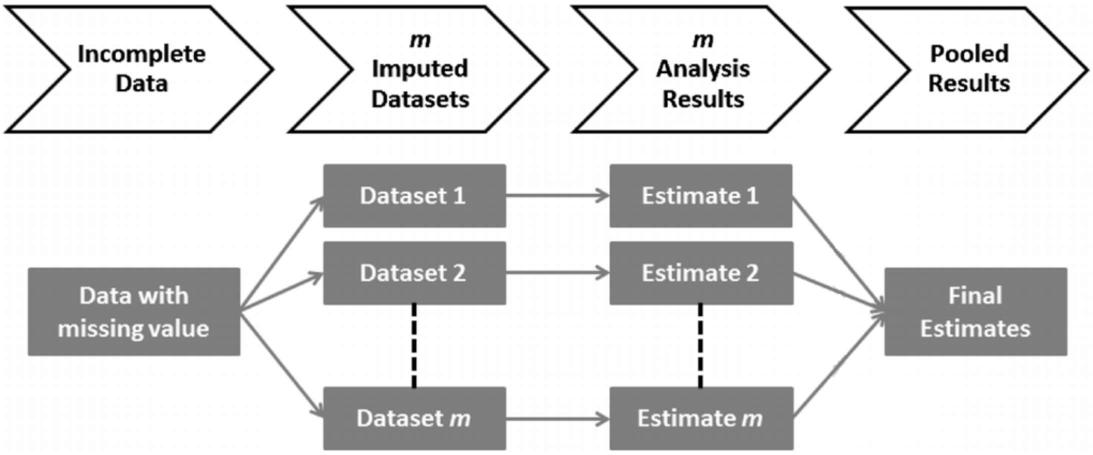

Figure 1: An graphical layout of multiple imputation. At the start a dataset with missing values is present. Missing values are imputed m times to create m imputed datasets. The m complete datasets are all analysed for m different sets of estimates. These results are then pooled into one complete set of estimates.[38, 39]

- *Value imputation*
  The value imputation per dataset has different approaches available. One is similar to the previously discussed SI hot deck imputation. The quickest imputation would be to randomly choose a value from the distribution of the feature. After analysing all different datasets and combining them, it would seem as if the missing values were following the same distribution. Other techniques are similar to the multivariate regression imputation and the kNN imputation[36, 40]. The variance can be added manually. The variance is then averaged for all variances and in the end using a new averaged variance[25]. Other mainly regression methods use different initial seeds as a way of creating a variance[41].

- *Additional datasets*
  The number of suggested datasets is different per study and usually differs between 3 and 10[42, 36, 43, 44, 45, 46], however outliers of 15[47] and even 25[37] are known. Other researchers even propose the number of datasets to be the same as the percentage of missing values[40]. The number of datasets is proportionally to the quality of the result and more datasets would indicate a better result. However generating and analysing more datasets also takes much more computation time, therefore a trade-off in computation time and quality should be considered.

- *Analysis*
  The analysis of the dataset is project specific. Fro examples, if the outcome should be a linear equation between features and output, linear regression should be used to compute such an equation. Usually the outcome of such an analysis is more precise than it actually should be. By doing the same analysis multiple times with MI, a measure of uncertainty can be given to the outcome. This is done by combining the different outcomes together to one final outcome, by for example show standard deviations per regression coefficient[25, 44].

The main advantage of multiple imputation is the addition of uncertainty for imputation. In single imputation this uncertainty is unfairly low, which can create bias in results. Because of it being an extension on single imputation, it also inherits the advantages of being able to handle MAR values and possibly MNAR, and raises the number of samples to be used for analysis[44, 36]. The disadvantages mainly consist of taking up more computation time, due to multiplication of the analysis for the datasets. Therefore complex analysis techniques that take longer to compute may become unusable. Another aspect is the error given to missing value imputations. Not all analysis techniques perform well with errors which may create worse results even after combining the results[36].

Several MI implementations are created and also added to mainly statistical analysis programs as SOLAS, SAS, SPSS, S-Plus and Stata[48, 49, 50]. The most used implementation is multiple imputation for chained equations (MICE)[45, 46], also known as sequential regression multiple implementation (SRMI)[41]. MICE stores the location of the missing values and iteratively replaces them with new values, according to a six step system[45].

1. Replace all missing values with the mean of the corresponding feature.
2. Choose a feature f and remove the imputed values on all missing value locations.
3. Create a model with all samples that do not have a missing value in f to predict the value in f by the other features.
4. Impute the values in f using the created model.
5. Repeat steps 2 to 4 for all features that have missing values.
6. Iteratively repeat steps 2 to 5 s cycles for better more precise imputed values to create a final dataset.

The six steps create one dataset. The steps are m times with different orders of choosing f to create m different datasets for analysis. The number of cycles used in the sixth step influences the precision of the outcome, as eventually the imputed missing values are expected to converge to one value. Usually about 10 cycles are done, as it is a good equilibrium between low computation time and high imputed value precision[45, 46].

3 Hypotheses

The main goal for this project is to find more insight in quality enhancement and add that insight to the final framework. Several hypotheses are made before designing experiments to test them and are discussed here:

1. **Distribution preservation**
   One thing missing value handling algorithms should do is preserve the distribution of a feature, not creating any bias in it. List Deletion methods CCA and ACA are not expected to do that properly in most cases, as most values are not MCAR. WCA might be able to efficiently tackle distribution problems, but is expected to be outperformed by imputation methods. Of the imputation methods MICE is expected to perform best, followed by kNN and regression imputation. Hot deck and mean imputation obviously are also expected to preserve distribution characteristics very well as their imputation method is based on it.
2. **Quality improvement** Missing values handling algorithms are necessary, however can show different levels of quality improvement. This difference in quality indicates best approaches to cope with missing values. Again, list deletion algorithms are expected to perform worst in quality improvement. Next are the easy single imputation methods, the mean imputation, hot encoding and missing indicator algorithms. These are followed by the more advanced single imputation algorithms the kNN and regression algorithms and the multiple imputation algorithm MICE is expected to outperform all.

4 Methods

To determine the quality of different missing value handling algorithms, two different approaches are used. The first approach focuses on the distributions of the different features and whether the algorithms create a bias in feature distributions. The second approach focuses on the evaluation quality and which missing value handling algorithm works best on predicting output. The tests are done in Python, using the Anaconda distribution, as with Anaconda external packages can be used quickly. The missing value handling alrogithms are custom made with several help functions from the pacakages scikit-learn[51] and NumPy[52]. The four example datasets (Subsection 2.1) were used as exemplary datasets for the experiments.

4.1 Bias Evaluation

After manipulating a dataset, the distributions of a feature can change significantly. This change in distribution can create a bias in the results and must be prevented as much as possible. To find out whether this bias is present, feature type specific values are compared between both the old and new feature values and those are compared using several statistical methods.

For future mentions of the two distributions that are compared the names 'old data' (before list deletion or value imputation) and 'new data' (after list deletion or value imputation) are used. The statistical methods that are used to find bias all are based on the same hypothesis H0: the old data and new data originate from the same distribution. This means the rejection hypothesis becomes H1: old and new data do not originate from the same distribution. Since features were nominal, ordinal or categorical, every feature type hypothesis is evaluated differently:

- *Nominal features*
  For nominal features the mean and variance are main aspects to compare nominal distributions. To either accept the hypothesis, these two aspects are tested. The mean comparisons were done by using the known t-test[53]. The equality of variance is tested by using a Levene's test with the Brown and Forsythe adaptation. This variance test was chosen due to not having any major assumptions on the distributions, for example the normality distribution[54].

- *Ordinal features*
  For ordinal features the mean cannot be used, as values are either not numeric or only give an indication with the numeric values. Because of that the median is used instead. The hypothesis is tested by checking whether this median is the same for both the old and new values. To have a better look at the distribution of the ordinal values, also a chi square test is done. This chi square test computes whether the distribution of ordinal values is the same for both the old and new values[55].

- *Categorical features*
  Categorical features do not have a mean or median, due to no ordering being present. Therefore the mode of the old and the new distribution is checked to be the same, to test the hypothesis H0 to be true. On top of that, the chi square test is here used, to find out whether the features from before and after the missing values algorithms can follow the same distribution[55].

All features from the four datasets are tested whether any bias is present or not. The missing values are handled by seven different algorithms: CCA (Algorithm 1), ACA (Algorithm 2), Mean/-Median/Mode imputation (Algorithm 5), Hot Deck imputation (Algorithm 6), kNN imputation (Algorithm 8), Regression imputation(Algorithm 7) and at last MICE (Subsection 2.2.3). An overview of the datasets, evaluation tests and the missing value handling algorithms is provided (Table 2).

The outcome of all the tests are shown in multiple tables. In those tables tests that show a rejection of H0 are highlighted. For the list deletion algorithms, additionally the type of missing

Table 2: The three different aspects of the bias evaluation. Four different datasets are used. Two tests are done per feature type, to test hypothesis H0 and seven missing value handling algorithms are implemented to be tested.

Datasets	Heart Attack dataset Hepatitis dataset Cirrhosis dataset Cervical cancer dataset	All datasets containing missing values that are evaluated
Test types	Nominal features (t-test, levene's test) Ordinal features (median, chi-square) Categorical features (mode, chi-square)	Different types of features have different tests for hypothesis H0: The old and new distributions are the same
Methods	CCA WCA Mean/Median/Mode imputation Hot deck imputation kNN imputation Regression imputation MICE	All methods to handle missing values: - Two List Deletion algorithms - Four Single Imputation algorithms - One Multiple Imputation algorithms

values is evaluated by also testing the change in distribution in values without missing values. If a feature distribution changes for a feature without missing values after removing all missing values, a relation is present and the missing values are not MCAR. Aside from these tables plots of the relation between the percentage of missing values and probability of bias being present are made for every algorithm. This can indicate quality between the different algorithms with regards of the statistical hypothesis. In those plots also a regression line is shown with also the R^2 value, to show the quality of the regression[56]. R^2 is measured by first measuring the variance sum towards the mean (SS_{tot}, y_i is the value for datapoint i, $\bar{y}$ is the mean of datapoints y) and the variance sum towards the regression line (SS_{reg}, f_i is regression value for i). Secondly the difference is measured and this difference is normalized by dividing it by the mean variance sum. R^2 should be a value on interval $[0, 1]$ with 1 being a good fit and 0 being a bad fit.

$$R^2 = \frac{SS_{tot} - SS_{res}}{SS_{tot}}), SS_{tot} = \sum (y_i - \ bary)^2, SS_{res} = \sum (y_i - f_i)^2 \tag{1}$$

4.2 Quality Evaluation

To determine the quality of the missing value imputation methods, the classification quality of the dataset was tested after handling the missing values. This is done for the Hepatitis, Cirrhosis and Cervical Cancer datasets, as on the Heart Attack dataset regression should be performed. This classification quality was tested by using the basic logistic regression algorithm from scikit-learn. This classification algorithm only accepts numerical data, therefore all categorical features are first hot encoded. Previously discussed missing value handling algorithms (Table 3) are tested on their accuracy, precision, recall and F1s score with a 10-fold cross validation. For the datasets an additional classification was done after removing the values with more than 15% missing values, to find out if that would give a difference in quality, as well.

Table 3: All tested missing value handling methods and possible parameters used during testing

Type	Method	Parameters
List Deletion	CCA (Algorithm 1)	
	ACA (Algorithm 2)	– - threshold: 10% missing values –
	WCA (Algorithm 3)	
Single Imputation	Mean/Median/Mode (Algorithm 5)	
	Hot deck (Algorithm 6)	
	Missing indicator (Algorithm 4)	– - Imputed value: Mean, Zero –
	Regression (Algorithm 7)	
	k-Nearest neighbour (Algorithm 8)	– - k = 1, 3, 5

Multiple Imputation	MICE
	(Subsection 2.2.3)
		s = 3, 5
		m = 3, 5

To show the differences between the datasets, the best outcomes are shown separately, as well as an average over the three datasets. This way interesting differences between the datasets can be located and discussed.

5 Results

5.1 Bias Evaluation Results

The outcome of the distribution tests for the heart attack dataset are visualised (Tables 4 and 6), as well as the tables for the other datasets (Appendix A). The tables are split between list deletion (Table 4) and imputation (Table 6) results.

Table 4: Testing for the heart attack data set whether the type of missing values can be represented by the remaining values. This is done by comparing distributions between the old and new data after CCA and WCA. For nominal values, two tests were used, an independent t-test for equality of mean and a Levene's test with the median in brackets to test equality of variance. For ordinal and categorical features the medians and modes where compared respectively as well as a chi squared test in brackets for equality of distribution. P-values lower than $p < 0.05$ are marked red for failure of representation, p-values higher than $p > 0.05$ are marked green for correctly being represented. If at least one feature is not represented after CCA, the missing values cannot be MCAR. If at least one feature is not represented after WCA, the pseudo-randomness cannot be corrected by only using weights for other values.

Feature name	**still alive**	**age**	**pericardial**	**fractional**	**epss**	**lvdd**	**wall score**	**wall index**	**alive at 1**
Value type	Cat	Nom	Cat	Nom	Nom	Nom	Nom	Nom	Cat
Missing	0%	3.85%	0%	5.38%	10.77%	7.69%	2.31%	0.77%	43.85%
p-values CCA	True (0.96)	<0.01 (0.39)	True (0.99)	<0.01 (0.83)	0.99 (0.74)	0.99 (0.94)	<0.01 (0.98)	0.09 (0.84)	True (0.94)
p-values WCA	True (1.00)	0.48 (0.82)	True (0.98)	<0.01 (0.45)	0.99 (0.39)	0.99 (0.28)	<0.01 (0.54)	0.99 (0.91)	True (0.92)

The tables show that in the heart attack, cirrhosis and cervical cancer datasets at least one feature shows a difference between the old and new data. Highlights are shown for changes in distribution for all values:

- **Heart attack:** *age*, *fractional* and *wall score* all have $p < 0.01$ for having the same mean before and after CCA. This indicates that the missing value type of at least one feature is MAR and for all three possibly even MNAR. WCA does help in creating a representative distribution for the feature *age*, however *fractional* and *wall score* still are differently distributed. Other approaches to handle missing values are most likely more effective

- **Hepatitis:** For no feature a reason to reject the distributions before and after deleting samples with missing values is present. The feature that is least likely to have the same distribution is *bilirubin* with having a $p = 0.14$ for the same mean and $p = 0.23$ for the same variance. We therefore cannot say that the missing values are MCAR and using only CCA to handle missing values is certainly possible. WCA has worse results for several features and show that WCA might not be the best way to improve the sample size for this dataset.

- **Cirrhosis:** *case number*, *status*, *day*, *albumin* and *SGOT* all have $p < 0.05$ if nominal for having the same mean and *False* if ordinal or categorical for having the same median or mode, respectively. These five features do not have missing values. This means that the bias in the distribution is caused by list deletion due to another feature's missing values. Therefore it can be concluded that least for one feature the missing values are at least MAR. WCA helps fixing the distributions for *case number* and *SGOT*, but also creates additional bias in *presence of ascites*, *presence of hepatocytes* and *presence of edema* and therefore is not a suitable approach to remove possible bias.

- **Cervical cancer:** *STDs*, *#STDs*, *condylomatosis*, *vulvoperineal condlymatosis* and *STDs #diagnosis* all have $p < 0.05$ if nominal for having the same mean and *False* if categorical for having the same mode. All but *STDs #diagnosis* also have a $p < 0.05$ for having the same variance if nominal or the same chi square distribution if categorical, which further strengthens hypothesis H1 of the distributions being different. This indicates that the missing values of at least one feature is MCAR.

Table 6: Testing for the heart attack data set if certain types of imputation create a vastly different distribution for features with missing values. The imputation values are generated with mean imputation, hot deck imputation, k-Nearest Neighbour imputation ($k = 3$), regression imputation and MICE (number of cycles is $s = 5$). For nominal values, two tests were used, an independent t-test for equality of mean and a Levene's test with the median in brackets to test equality of variance. For ordinal and categorical features the medians and modes where checked respectively to be similar as well as a chi squared test in brackets fro equality of distribution. P-values lower than $p < 0.05$ are marked red for failure of representation, p-values close to $p > 0.05$ are marked green for correctly being represented.

Feature name	**age**	**frac-tional**	**epss**	**lvdd**	**wall score**	**wall index**	**alive at 1**
Value type	Nom	Nom	Nom	Nom	Nom	Nom	Cat
Missing	3.85%	5.38%	10.77%	7.69%	2.31%	0.77%	43.85%
Mean Imputation	1.00 (0.76)	0.99 (0.60)	0.99 (0.39)	1.00 (0.48)	1.00 (0.88)	1.00 (0.98)	True (0.77)
Hot Deck Imputation	0.97 (0.98)	0.88 (0.88)	0.94 (0.98)	0.82 (0.91)	0.94 (0.86)	0.96 (0.96)	True (0.96)
kNN Imputation	0.86 (0.99)	0.81 (0.88)	0.69 (0.62)	0.94 (0.68)	0.96 (0.97)	0.96 (0.98)	True (0.91)
Regression Imputation	0.84 (1.00)	0.71 (0.95)	0.34 (0.57)	0.91 (0.76)	0.88 (1.00)	0.93 (0.98)	True (0.92)
MICE (5 cycles)	0.97 (0.81)	0.94 (0.80)	0.98 (0.90)	0.80 (0.84)	0.98 (0.98)	0.96 (0.97)	True (0.93)

Table 5: A table showing the number of samples that remained after performing CCA.

Dataset	**Total samples**	**Number of samples after CCA**
Heart Attack	108	61
Hepatitis	155	80
Cirrhosis	1945	1113
Cervical Cancer	585	59

The heart attack, cirrhosis and cervical cancer datasets all indicate possible bias after deleting samples with missing values. On top of that, after removing all samples with missing values not many samples are left for any of the datasets (Table 5). The hepatitis dataset loses almost 50% of its samples after CCA and the cervical cancer dataset drops down to only 10% of its original values. All four datasets would therefore potentially benefit from an imputation approach.

Most imputation methods show an improvement in results The heart attack dataset has no bias in distributions after any of the five tested imputation methods, the hepatitis dataset only has bias when using mean imputation or MICE, the cirrhosis dataset only shows bias when using mean imputation and at last in the cervical cancer dataset all but the two features with 91.75% missing values contain no bias between distributions. On top of that, the only features that had bias in the newly created distributions are features with more than 40% missing values, indicating that imputation methods have more problems when a higher missing/not missing ratio is present. this gives an indication that removal of features with missing values higher than 40% would greatly help the usefulness of a dataset.

A scatter plot with fitted linear curves is made to show a relation between the probability of the old and new mean (Figure 2) and variance (Figure 3) originating from the same distribution and the percentage of values missing. Although the R^2 values for some of the regression lines seem plausible, the value distribution does not seem to support this. Most features are within the first

15% of the missing values and the features above 15% missing values seem to deviate significantly from the others.

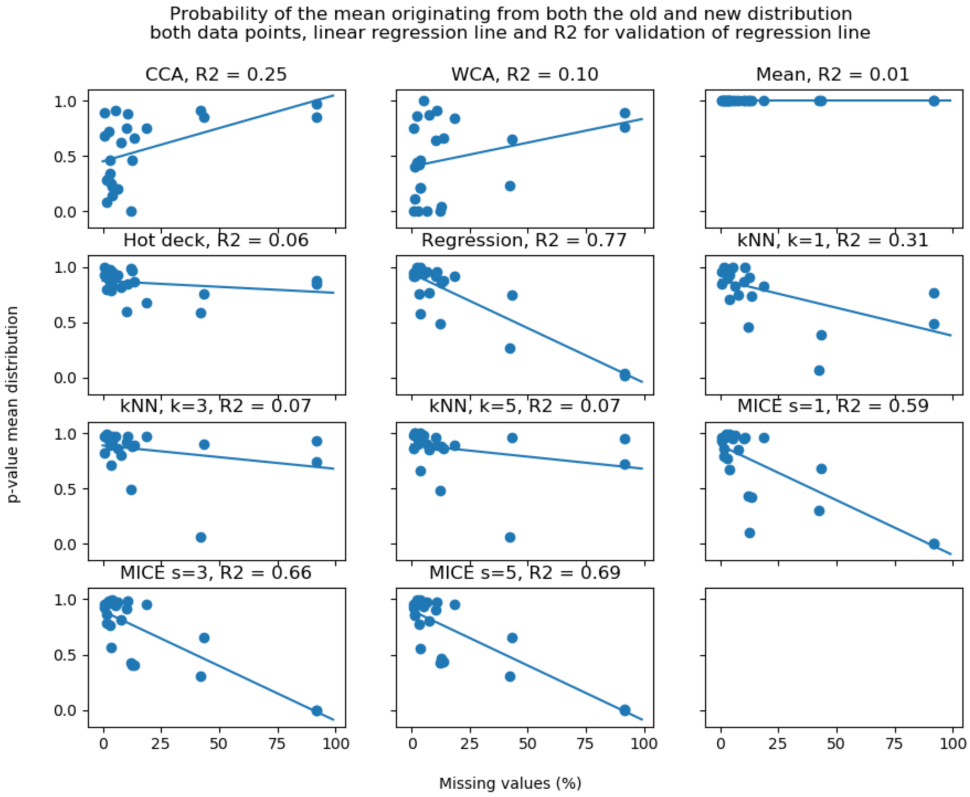

Figure 2: Plots showing the probability of the mean of the old and new distribution originating from the same distribution. On the x-axis the percentage of missing values is given for a feature and on the y-axis the p-value of the probability. For every missing value algorithm also a linear fit is made to show the trend of the scatter plot.

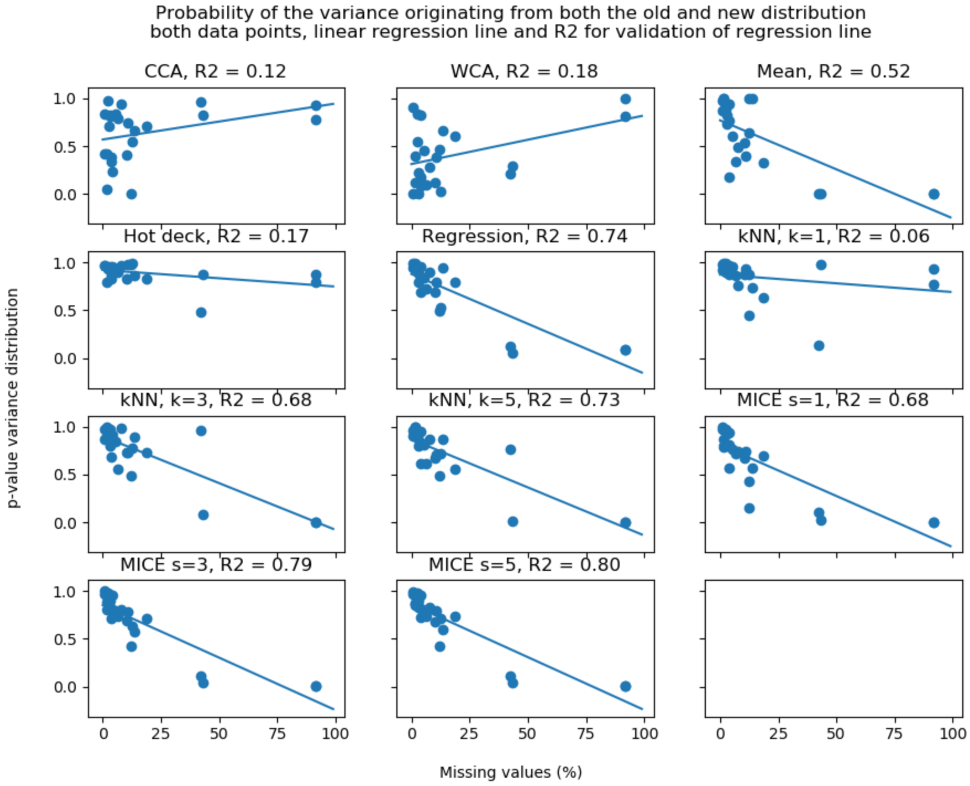

Figure 3: Plots showing the probability of the variance of the old and new distribution originating from the same distribution. On the x-axis the percentage of missing values is given for a feature and on the y-axis the p-value of the probability. For every missing value algorithm also a linear fit is made to show the trend of the scatter plot.

It seems that features with more than 15% missing values are very hard to find proper values for using any missing value handling technique. Therefore an additional approach was used, by first performing ACA for features with more than 15% missing values, followed by the known missing value handling techniques (Figure 4 and 5).

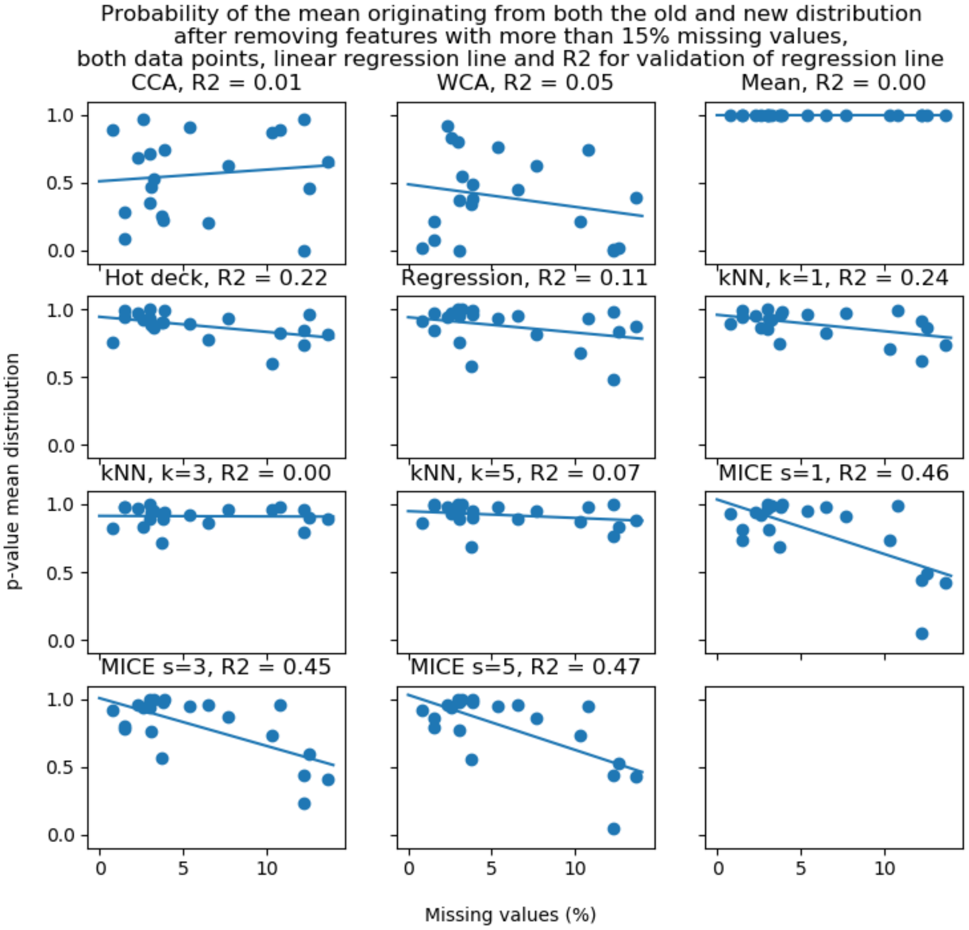

Figure 4: Plots showing the probability of the mean of the old and new distribution originating from the same distribution. On the x-axis the percentage of missing values is given for a feature and on the y-axis the p-value of the probability. For every missing value algorithm also a linear fit is made to show the trend of the scatter plot.

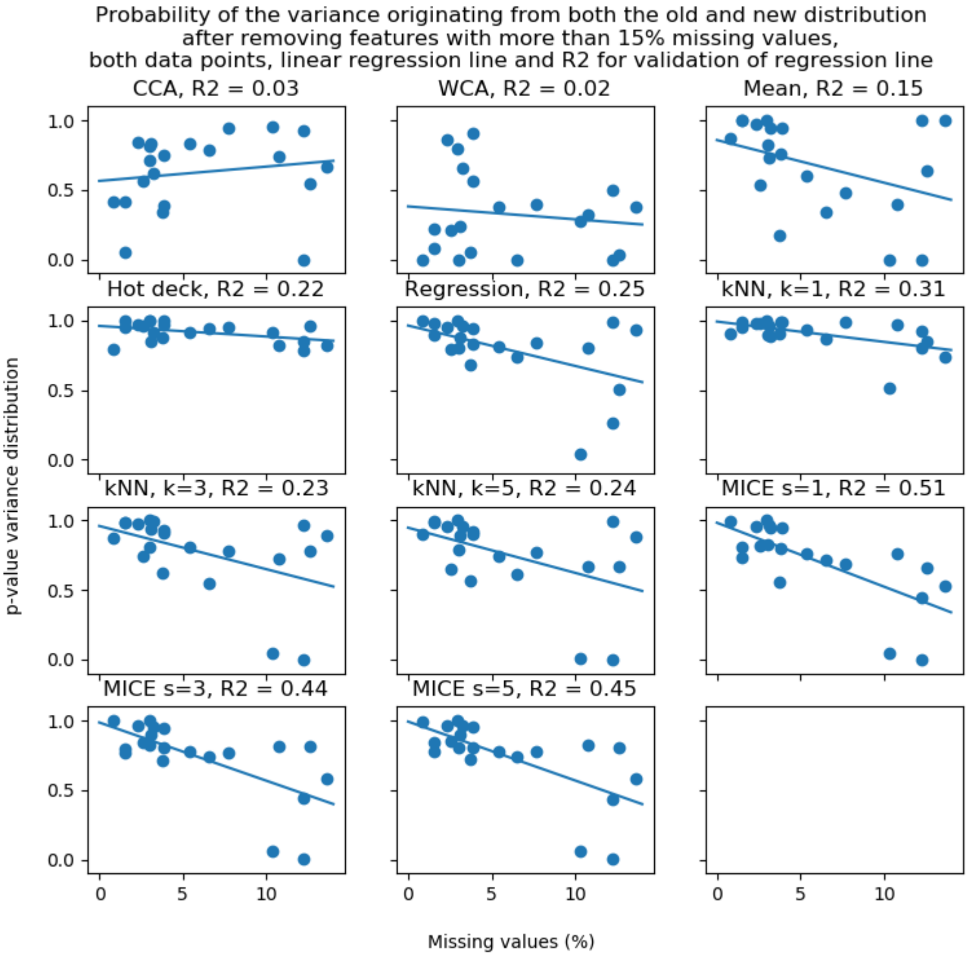

Figure 5: Plots showing the probability of the variance of the old and new distribution originating from the same distribution. Features with more than 15% missing values were removed before missing value handling was initiated. On the x-axis the percentage of missing values is given for a feature and on the y-axis the p-value of the probability. For every missing value algorithm also a linear fit is made to show the trend of the scatter plot.

List Deletion algorithms seem to be better at creating a representative distribution when more missing values are present. This is explainable for these datasets as all of them have a at least one feature with more than 40%. Most samples are removed because of this feature also removing significant portions of other features. This results in features having more missing values will leave them with a higher percentage of their original values, therefore being closer to the original values.

The imputation algorithm seem to show big differences between the results. Hot deck imputation seems to create the best representative distribution, being the best when comparing mean and variance for the old and new distribution. Mean Imputation obviously shows a perfect fit for comparing means, however for variance imputation it is worse than both hot deck, kNN and regression. kNN imputation shows the best result after hot deck imputation. Interestingly kNN with k=1 neighbours shows the p-values for variance whereas k=3 and k=5 show the best p-values for the mean. Regression imputation shows good results as well, but is overall worse than kNN. MICE unexpectedly performs worse than regression and kNN in both mean and variance, even though it was expected to perform better.

The regression lines for all imputations seem reasonable in performance. Either the p-value is almost independent of the percentage missing values (mean, hot deck, regression, kNN) or the $R^2 \approx 0.5$, showing somewhat of a trend in the data. Interestingly the p-value seems to deviate much more for higher percentages, something that is to be expected since the distribution can also change more when more values are missing.

The high quality for both mean and hot deck imputation can be explained by the evaluation methods. A mean comparison for mean imputation is obviously perfect, and the variance will always become smaller when more means are imputed. Hot deck imputation adds values according to the distribution of the features. This distribution follows the mean and variance of the original values and therefore the mean and variance will always be close to the original distribution. When looking at other aspects, for example the ability to predict the output with the dataset, hot deck imputation and mean imputation should show worse results as distribution specific estimations were imputed, instead of sample specific.

5.2 Quality Evaluation

The averaged results of the classification after using the missing value handling algorithms (Table 3) are shown in several figures. The average accuracy is shown for all three datasets with (Figure 7) and without (Figure 6) the removal of features with more than 15% missing values and without. The F1-score is also shown for the averaged dataset (Figure 8). At last a table to show the accuracy and F1 score is shown when all features with missing values are deleted to detect improvement for missing value imputation, as well as the best missing values handling algorithm for comparison (Table 7).

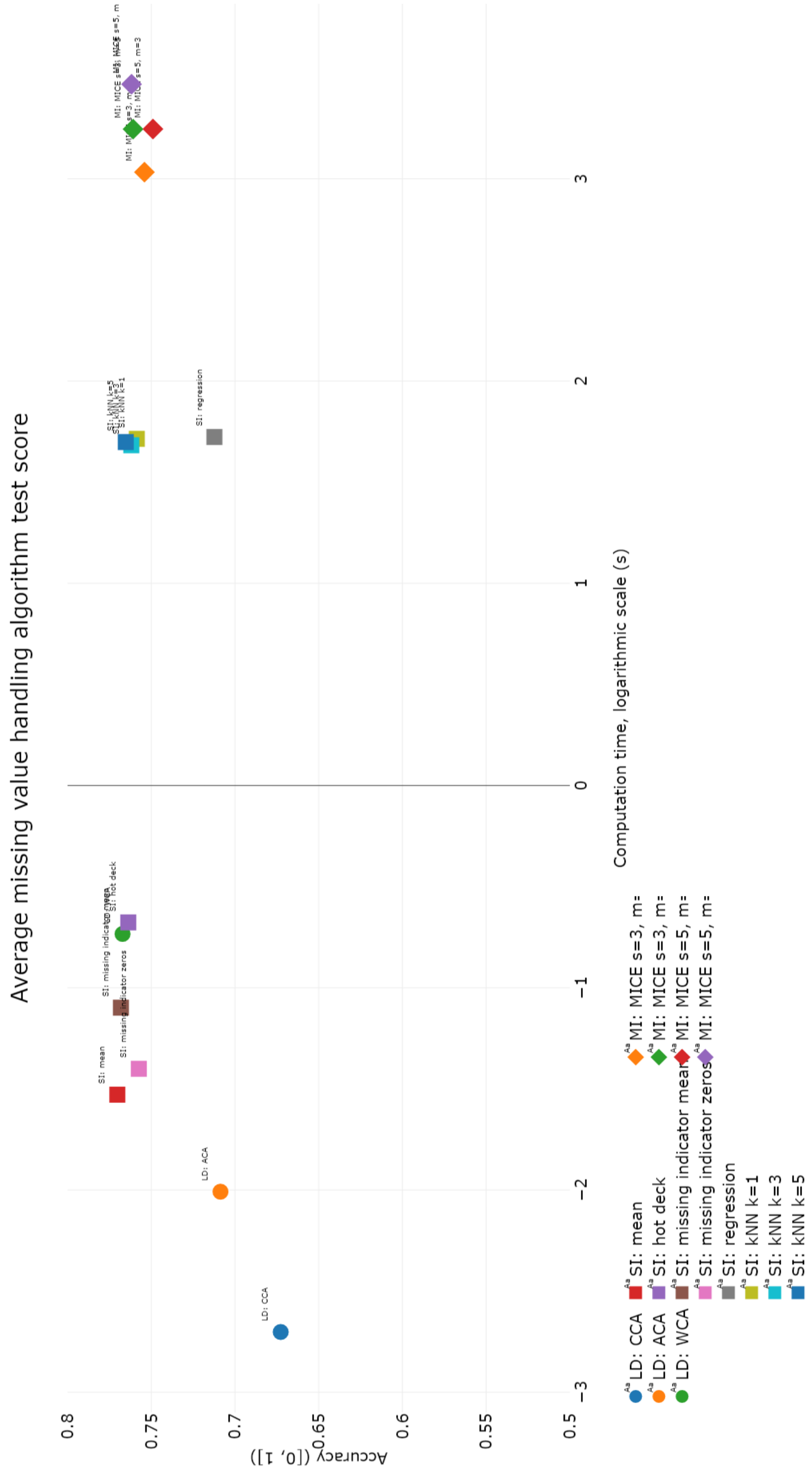

Figure 6: A plot showing the average accuracy over for the datasets classified after using a missing value handling algorithm. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.

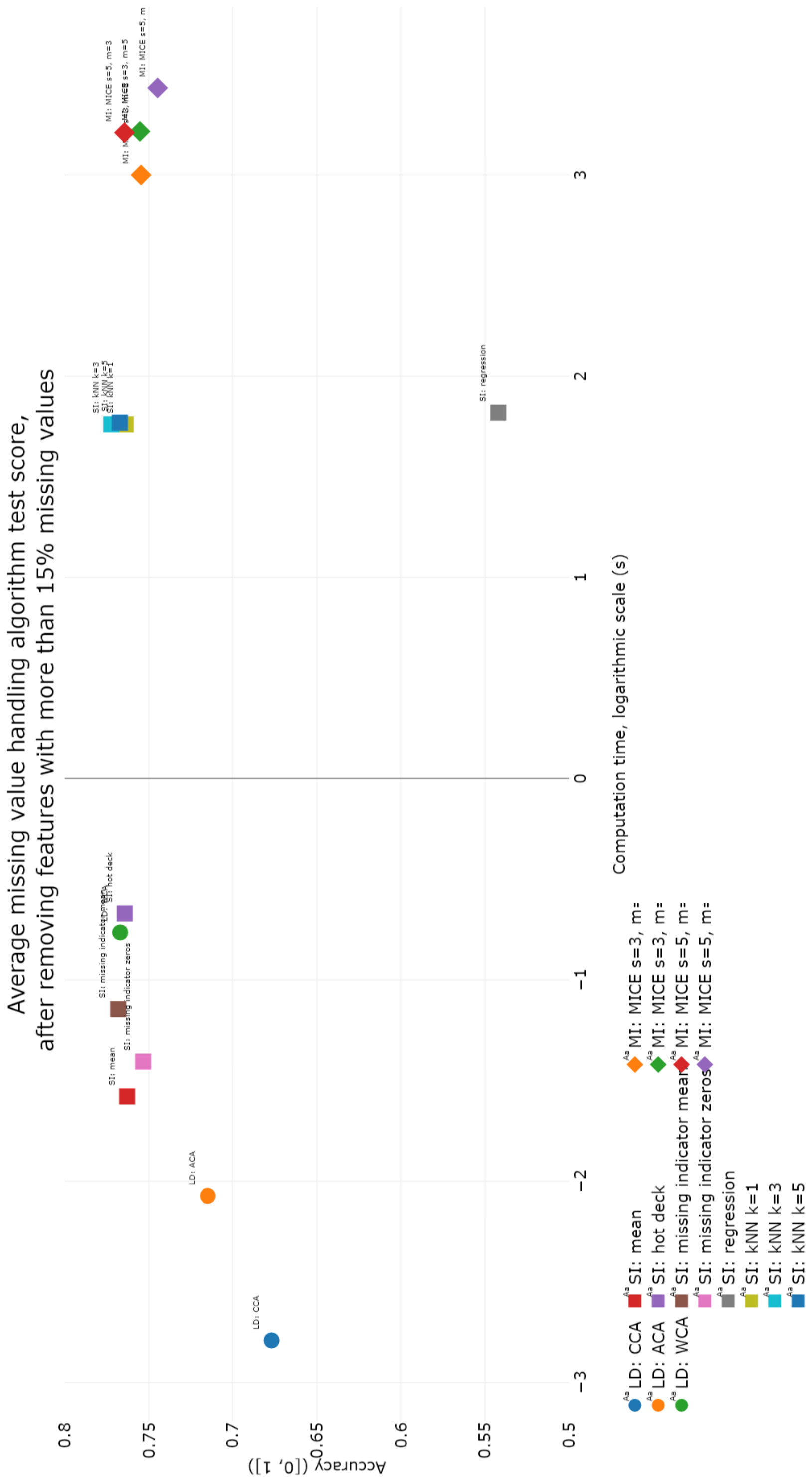

Figure 7: A plot showing the average accuracy over for the datasets classified after using a missing value handling algorithm and after removing all features with more than 15% missing values. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.

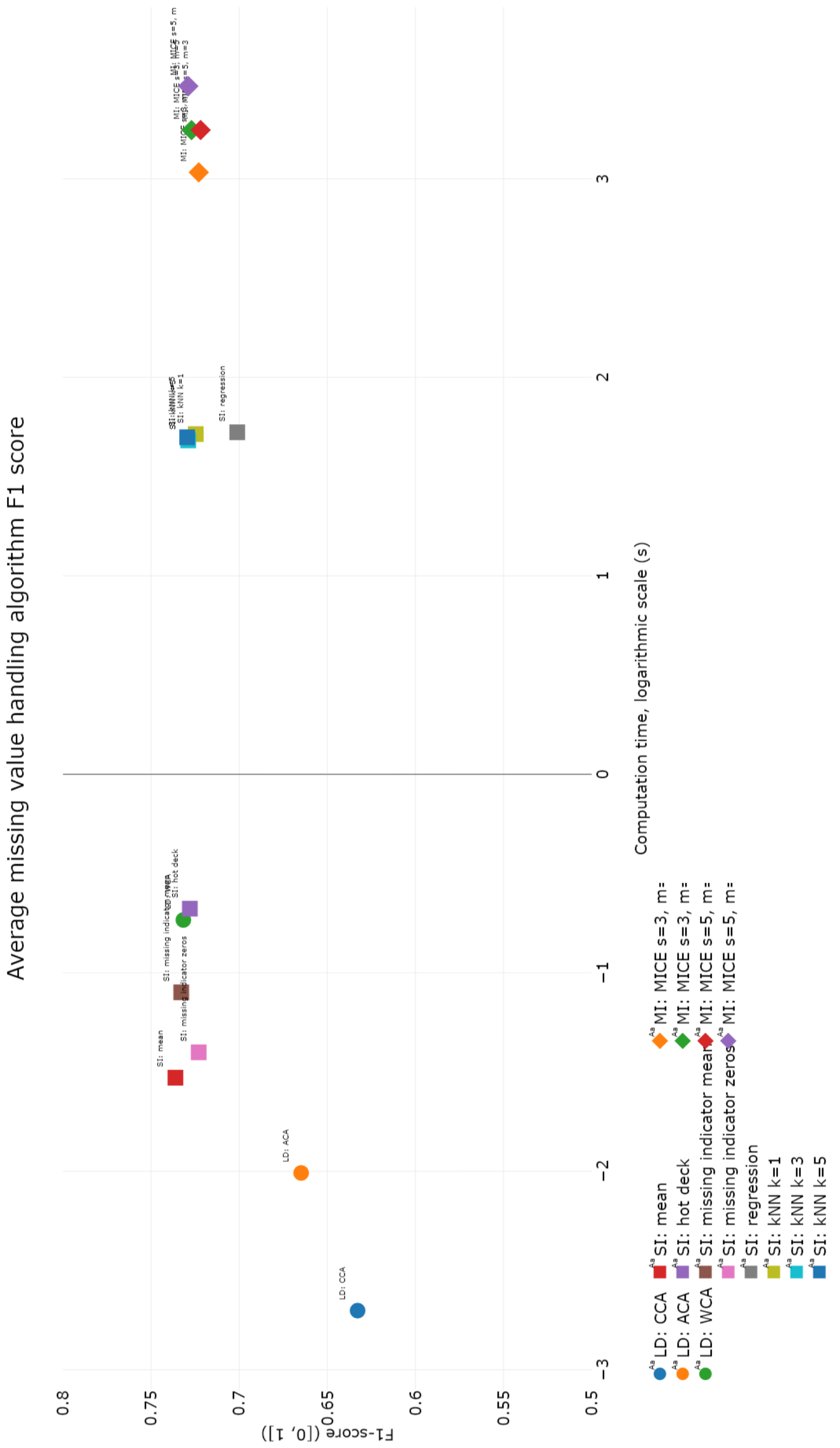

Figure 8: A plot showing the F1-score over for the datasets classified after using a missing value handling algorithm. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.

Table 7: The classification results of the datasets and the average after removing all features with missing values, as well as the best results after handling missing values.

	Without missing values		**Best classification**		
Dataset	**Accuracy**	**F1-score**	**Accuracy**	**F1-score**	**Algorithm**
Hepatitis	0.79	0.75	0.87	0.86	Mean imputation
Cirrhosis	0.56	0.50	0.57	0.52	– 1. CCA 2. Missing indicator mean 3. Missing indicator zeros –
Cervical	0.74	0.64	0.89	0.83	Mean imputation
Average	0.70	0.63	0.77	0.74	Mean imputation

The values of the F1 score (Figure 8) are lower than the accuracy (Figure 6), therefore the F1 score is also used in comparisons. The lack of difference in the accuracy in classification with (Figure 7) and without features with more than 15% missing values (Figure 6), shows the lack of contribution of these values in the classification. Even though the distribution of the features improves (Subsection 5.1), the classification does not show any difference. This was not the case for the dataset with regression imputation, though, as the accuracy severely dropped after removing features with more than 15% missing values from 0.72 to 0.54. Apparently important information was removed for the regression imputation to be successful.

When looking at separate methods, the difference in computation time can immediately be seen. List deletion algorithms and mean, missing indicator and hot deck single imputation algorithms all are very fast (less than a second), followed by k-nearest neighbour and regression single imputation (between 10 and 100 seconds) and at last multiple imputation methods (more than 1000 seconds). All imputation methods and WCA give the best accuracy, with little difference among them. The imputation methods and WCA all show an improvement compared with the accuracy when using only features without missing values (Table 7), therefore using features with missing values does improve accuracy. The size of improvement differs per dataset, though, as the improvement for the Cervical dataset is 10 times higher than the improvement of the Cirrhosis dataset.

Since the accuracy and F1-score of the datasets are significantly different (Table 7), the results are also plotted per dataset separately (Appendix B). The low accuracy for regression imputation seems to originate mainly for the Cervical Cancer dataset. This can be a result of features in the Cervical Cancer dataset mostly consisting of boolean values. Hot encoded boolean values contain less information for regression, making it harder to find good imputation values.

The imputation methods show some differences in classification quality when looking at the results for the datasets separately. The Cirrhosis dataset shows hardly any improvement after using missing value handling algorithms, but its best results are when using a missing indicator method. The Hepatitis dataset result on the other hand is significantly worse when using the missing indicator method. This is supported by the bias evaluation (Subsection 5.1) and the dataset description (Subsection 2.1), as both showed that data in the Cirrhosis is confirmed MAR, whereas the data in the Hepatitis dataset was least likely to be MAR. This means that when data is very likely to be MAR, missing indicator value is more likely to be beneficial to the outcome. Aside from that, the kNN, regression and MICE methods seem not worth the additional computation time for these datasets, even though in theory these methods make a more detailed guess of the outcome.

6 Conclusions

7 Discussion

References

[1] N. Gehlenborg, S. I. O'donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, *et al.*, "Visualization of omics data for systems biology," *Nature methods*, vol. 7, no. 3s, p. S56, 2010.

[2] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, *et al.*, "Minimum information about a microarray experiment (miame)—toward standards for microarray data," *Nature genetics*, vol. 29, no. 4, p. 365, 2001.

[3] J. S. Cottrell and U. London, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[4] K. Dettmer, P. A. Aronov, and B. D. Hammock, "Mass spectrometry-based metabolomics," *Mass spectrometry reviews*, vol. 26, no. 1, pp. 51–78, 2007.

[5] D. Capitani, A. P. Sobolev, and L. Mannina, "Nuclear magnetic resonance–metabolomics," *Food Authentication: Management, Analysis and Regulation*, p. 177, 2017.

[6] B. Liu, X. Zhou, Y. Wang, J. Hu, L. He, R. Zhang, S. Chen, and Y. Guo, "Data processing and analysis in real-world traditional chinese medicine clinical data: challenges and approaches," *Statistics in medicine*, vol. 31, no. 7, pp. 653–660, 2012.

[7] D. F. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, E. Campbell, and D. W. Bates, "Grand challenges in clinical decision support," *Journal of biomedical informatics*, vol. 41, no. 2, pp. 387–392, 2008.

[8] G. Magni, C. Caldieron, S. Rigatti-Luchini, and H. Merskey, "Chronic musculoskeletal pain and depressive symptoms in the general population. an analysis of the 1st national health and nutrition examination survey data," *Pain*, vol. 43, no. 3, pp. 299–307, 1990.

[9] P. Bertolazzi, G. Felici, P. Festa, and G. Lancia, "Logic classification and feature selection for biomedical data," *Computers & Mathematics with Applications*, vol. 55, no. 5, pp. 889–899, 2008.

[10] G. Piatetsky-Shapiro and P. Tamayo, "Microarray data mining: facing the challenges," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 2, pp. 1–5, 2003.

[11] A. Lommen, "Metalign: interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing," *Analytical chemistry*, vol. 81, no. 8, pp. 3079–3086, 2009.

[12] A. Holzinger, M. Dehmer, and I. Jurisica, "Knowledge discovery and interactive data mining in bioinformatics-state-of-the-art, future challenges and research directions," *BMC bioinformatics*, vol. 15, no. 6, p. I1, 2014.

[13] M. Wilkins, "Proteomics data mining," *Expert review of proteomics*, vol. 6, no. 6, pp. 599–603, 2009.

[14] D. Teodoro, R. Choquet, E. Pasche, J. Gobeill, C. Daniel, P. Ruch, and C. Lovis, "Biomedical data management: a proposal framework.," in *MIE*, pp. 175–179, Citeseer, 2009.

[15] M. Y. Galperin, "The molecular biology database collection: 2008 update," *Nucleic Acids Research*, vol. 36, no. suppl1, pp. D2–D4, 2008.

[16] A. Sturn, J. Quackenbush, and Z. Trajanoski, "Genesis: cluster analysis of microarray data," *Bioinformatics*, vol. 18, no. 1, pp. 207–208, 2002.

[17] A. Karnovsky, T. Weymouth, T. Hull, V. G. Tarcea, G. Scardoni, C. Laudanna, M. A. Sartor, K. A. Stringer, H. Jagadish, C. Burant, *et al.*, "Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data," *Bioinformatics*, vol. 28, no. 3, pp. 373–380, 2011.

[18] D. Tabas-Madrid, R. Nogales-Cadenas, and A. Pascual-Montano, "Genecodis3: a non-redundant and modular enrichment analysis tool for functional genomics," *Nucleic acids research*, vol. 40, no. W1, pp. W478–W483, 2012.

[19] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behavior research methods*, vol. 39, no. 2, pp. 175–191, 2007.

[20] N. C. Dracopoli, P. O'Connell, T. I. Elsner, J.-M. Lalouel, R. L. White, K. H. Buetow, D. Y. Nishimura, J. C. Murray, C. Helms, S. K. Mishra, *et al.*, "The ceph consortium linkage map of human chromosome 1," *Genomics*, vol. 9, no. 4, pp. 686–700, 1991.

[21] S. I. Goldberg, A. Niemierko, and A. Turchin, "Analysis of data errors in clinical research databases," in *AMIA annual symposium proceedings*, vol. 2008, p. 242, American Medical Informatics Association, 2008.

[22] T. Stibor, P. Mohr, J. Timmis, and C. Eckert, "Is negative selection appropriate for anomaly detection?," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 321–328, ACM, 2005.

[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[24] S. J. Roberts, "Extreme value statistics for novelty detection in biomedical data processing," *IEE Proceedings-Science, Measurement and Technology*, vol. 147, no. 6, pp. 363–367, 2000.

[25] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[26] M. H. Cartwright, M. J. Shepperd, and Q. Song, "Dealing with missing software project data," in *Software Metrics Symposium, 2003. Proceedings. Ninth International*, pp. 154–165, IEEE, 2003.

[27] J. S. Haukoos and C. D. Newgard, "Advanced statistics: missing data in clinical research—part 1: an introduction and conceptual framework," *Academic Emergency Medicine*, vol. 14, no. 7, pp. 662–668, 2007.

[28] G. Kan, C. A. Visser, J. J. Koolen, and A. J. Dunning, "Short and long term predictive value of admission wall motion score in acute myocardial infarction. a cross sectional echocardiographic study of 345 patients.," *Heart*, vol. 56, no. 5, pp. 422–427, 1986.

[29] P. Diaconis and B. Efron, "Computer-intensive methods in statistics," *Scientific American*, vol. 248, no. 5, pp. 116–131, 1983.

[30] B. Cestnik, "Kononenkoj., bratkoj.(1987): Assistant-86: A knowledge elicitation tool for sophisticated users," *Progress in machine learning*.

[31] P. A. Murtaugh, E. R. Dickson, G. M. Van Dam, M. Malinchoc, P. M. Grambsch, A. L. Langworthy, and C. H. Gips, "Primary biliary cirrhosis: prediction of short-term survival based on repeated patient visits," *Hepatology*, vol. 20, no. 1, pp. 126–134, 1994.

[32] K. Fernandes, J. S. Cardoso, and J. Fernandes, "Transfer learning with partial observability applied to cervical cancer screening," in *Iberian conference on pattern recognition and image analysis*, pp. 243–250, Springer, 2017.

[33] P. A. Patrician, "Multiple imputation for missing data," *Research in Nursing & Health*, vol. 25, no. 1, pp. 76–84, 2002.

[34] J. A. Sterne, I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter, "Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls," *Bmj*, vol. 338, p. b2393, 2009.

[35] I. Myrtveit, E. Stensrud, and U. H. Olsson, "Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods," *IEEE Transactions on Software Engineering*, vol. 27, no. 11, pp. 999–1013, 2001.

[36] A. B. Pedersen, E. M. Mikkelsen, D. Cronin-Fenton, N. R. Kristensen, T. M. Pham, L. Pedersen, and I. Petersen, "Missing data and multiple imputation in clinical epidemiological research," *Clinical Epidemiology*, vol. 9, p. 157, 2017.

[37] T. E. Raghunathan, J. M. Lepkowski, J. Van Hoewyk, and P. Solenberger, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Survey methodology*, vol. 27, no. 1, pp. 85–96, 2001.

[38] S. Chevret, S. Seaman, and M. Resche-Rigon, "Multiple imputation: a mature approach to dealing with missing data," *Intensive care medicine*, vol. 41, no. 2, pp. 348–350, 2015.

[39] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[40] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: issues and guidance for practice," *Statistics in medicine*, vol. 30, no. 4, pp. 377–399, 2011.

[41] Y. He, A. M. Zaslavsky, M. Landrum, D. Harrington, and P. Catalano, "Multiple imputation in a large-scale complex survey: a practical guide," *Statistical methods in medical research*, vol. 19, no. 6, pp. 653–670, 2010.

[42] S. Van Buuren, "Multiple imputation of discrete and continuous data by fully conditional specification," *Statistical methods in medical research*, vol. 16, no. 3, pp. 219–242, 2007.

[43] S. Van Buuren, H. C. Boshuizen, D. L. Knook, *et al.*, "Multiple imputation of missing blood pressure covariates in survival analysis," *Statistics in medicine*, vol. 18, no. 6, pp. 681–694, 1999.

[44] G. J. Van der Heijden, A. R. T. Donders, T. Stijnen, and K. G. Moons, "Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: a clinical example," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1102–1109, 2006.

[45] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?," *International journal of methods in psychiatric research*, vol. 20, no. 1, pp. 40–49, 2011.

[46] P. Royston *et al.*, "Multiple imputation of missing values," *Stata journal*, vol. 4, no. 3, pp. 227–41, 2004.

[47] E. Martín-Merino, A. Calderón-Larrañaga, S. Hawley, B. Poblador-Plou, A. Llorente-García, I. Petersen, and D. Prieto-Alhambra, "The impact of different strategies to handle missing data on both precision and bias in a drug safety study: a multidatabase multinational population-based cohort study," *Clinical epidemiology*, vol. 10, p. 643, 2018.

[48] N. J. Horton and S. R. Lipsitz, "Multiple imputation in practice: comparison of software packages for regression models with missing variables," *The American Statistician*, vol. 55, no. 3, pp. 244–254, 2001.

[49] P. D. Allison, "Multiple imputation for missing data: A cautionary tale," *Sociological methods & research*, vol. 28, no. 3, pp. 301–309, 2000.

[50] P. Royston, I. R. White, *et al.*, "Multiple imputation by chained equations (mice): implementation in stata," *J Stat Softw*, vol. 45, no. 4, pp. 1–20, 2011.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[52] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[53] R. M. Heiberger and B. Holland, *Statistical analysis and data display.* Springer, 2004.

[54] M. B. Brown and A. B. Forsythe, "Robust tests for the equality of variances," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 364–367, 1974.

[55] A. Satorra and P. M. Bentler, "A scaled difference chi-square test statistic for moment structure analysis," *Psychometrika*, vol. 66, no. 4, pp. 507–514, 2001.

[56] N. R. Draper and H. Smith, *Applied regression analysis*, vol. 326. John Wiley & Sons, 2014.

A Feature Distribution Tables

The distribution tables for the hepatitis (Tables **??** and **??**), cirrhosis (Tables 10 and 11) and cervical cancer (Table **??**) datasets. For the hepatitis and cirrhosis datasets the tables are split in list deletion (Tables **??** and 10) and imputation (Tables **??** and 11). For the cervical cancer dataset both tables were combined due to not being any significant difference between the imputation technique outcomes.

Table 10: Testing for the cirrhosis dataset whether the type of missing values can be represented by the remaining values. This is done by comparing distributions between all sample values and remaining sample values after CCA and WCA. For nominal values, two tests were used, an independent t-test for equality of mean and a Levene's test with the median in brackets to test equality of variance. For ordinal and categorical features the medians and modes where checked respectively to be similar as well as a chi squared test in brackets fro equality of distribution. P-values lower than $p < 0.05$ are marked red for failure of representation, p-values higher than $p > 0.05$ are marked green for correctly being represented. If at least one feature is not represented after CCA, the missing values cannot be MCAR. If at least one feature is not represented after WCA, the pseudo-randomness cannot be corrected by only using weights for other values.

Feature name	**case number**	**number of days**	**status**	**drug**	**age**	**sex**	**day**	**pres. of ascites**	**pres. of hep.**
Value type	Nom	Nom	Ord	Cat	Nom	Cat	Nom	Cat	Cat
Missing	0%	0%	0%	0%	0%	0%	0%	3.08%	3.13%
p-values CCA	<0.01 (0.13)	0.69 (<0.01)	False (<0.01)	True (0.99)	0.65 (0.82)	True (0.97)	<0.01 (<0.01)	True (0.99)	True (1.00)
p-values WCA	0.40 (0.62)	0.75 (0.16)	False (0.74)	True (1.00)	0.86 (0.26)	True (0.26)	0.48 (0.02)	True (0.02)	False (0.97)
Feature name	**pres. of spiders**	**pres. of edema**	**serum bilirubin**	**serum cholesterol**	**albumin**	**alkaline phosphatase**	**SGOT**	**platelets**	**prothrombin time**
Value type	Cat	Ord	Nom	Nom	Nom	Nom	Nom	Nom	Nom
Missing	3.23%	0%	0%	42.2%	0%	3.08%	0%	3.75%	0%
p-values CCA	True (0.99)	True (0.71)	0.08 (0.07)	0.91 0.97	0.03 (<0.01)	0.47 (0.83)	0.01 (0.10)	0.25 (0.34)	0.79 (0.90)
p-values WCA	True (0.99)	False (0.58)	0.29 (0.22)	0.23 (0.21)	0.78 (<0.01)	0.43 (0.23)	0.31 (0.17)	0.21 (0.08)	0.44 (0.21)

Table 11: Testing for the cirrhosis data set if certain types of imputation create a vastly different distribution for features with missing values. The imputation values are generated with mean imputation, hot deck imputation, k-Nearest Neighbour imputation ($k = 3$), regression imputation and MICE (number of cycles is $s = 5$). For nominal values, two tests were used, an independent t-test for equality of mean and a Levene's test with the median in brackets to test equality of variance. For ordinal and categorical features the medians and modes where checked respectively to be similar as well as a chi squared test in brackets fro equality of distribution. P-values lower than $p < 0.05$ are marked red for failure of representation, p-values close to $p > 0.05$ are marked green for correctly being represented.

Feature name	**pres. of ascites**	**pres. of hep.**	**pres. of spiders**	**serum cholesterol**	**alkaline phosphatase**	**platelets**
Value type	Cat	Cat	Cat	Nom	Nom	Nom
Missing	3.08%	3.13%	3.23%	42.2%	3.08%	3.75%
Mean Imputation	True (0.99)	True (0.98)	True (0.98)	1.00 (<0.01)	1.00 (0.73)	1.00 (0.18)
Hot Deck Imputation	True (0.99)	True (1.00)	True (1.00)	0.95 (0.79)	0.97 (0.95)	0.93 (0.84)
kNN Imputation	True (0.98)	True (0.99)	True (0.98)	0.69 (0.40)	0.89 (0.82)	0.50 (0.86)
Regression Imputation	True (0.97)	True (0.98)	True (0.97)	0.59 (0.62)	0.64 (0.85)	0.32 (0.89)
MICE (5 cycles)	True (0.96)	True (0.98)	True (0.97)	0.31 (0.11)	0.77 (0.90)	0.56 (0.72)

B Classification Plots

The classification results for each dataset separately. A figure for the Hepatitis (Figure 9), Cirrhosis (Figure 10) and the Cervical Cancer dataset (Figure 11) are shown.

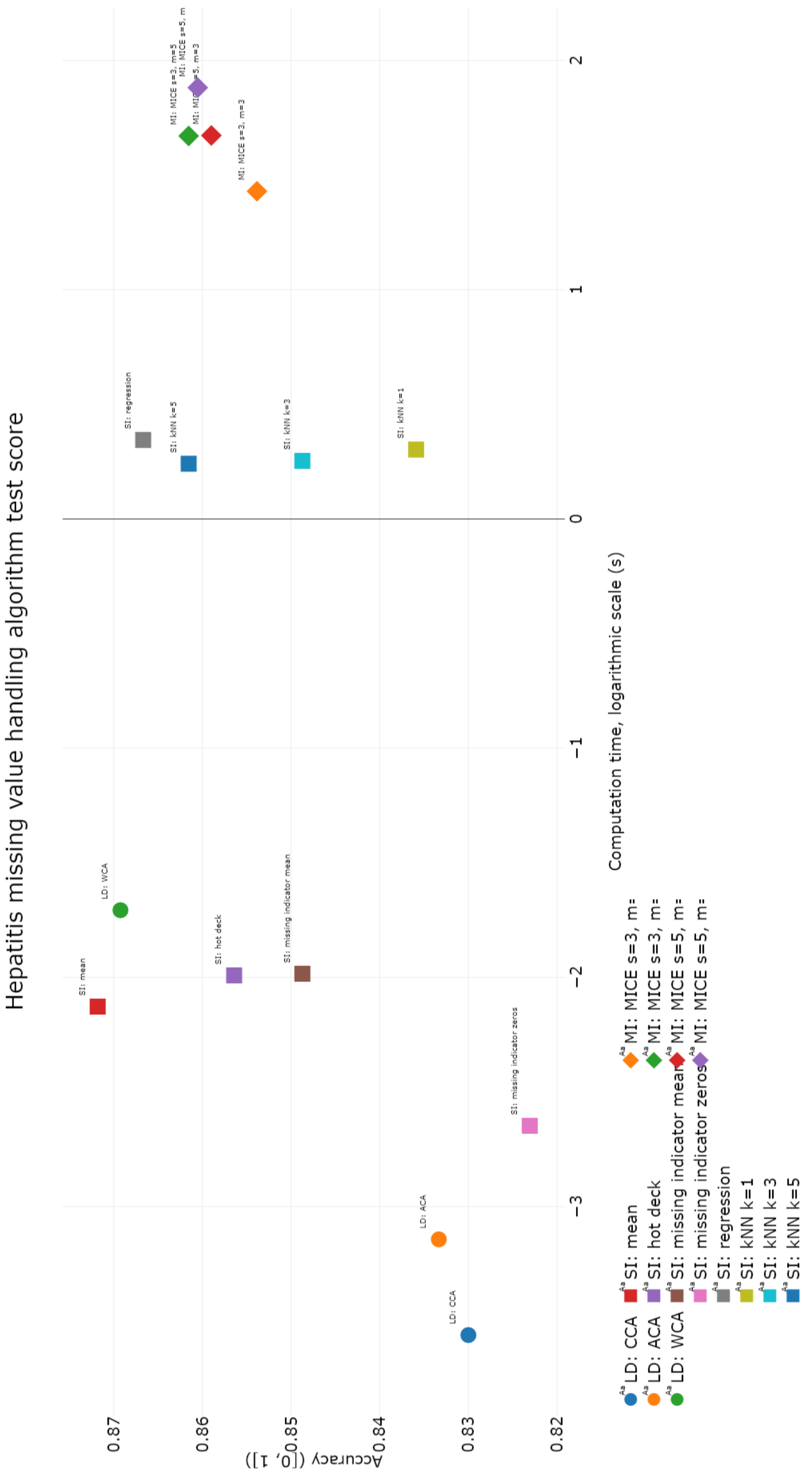

Figure 9: A plot showing the accuracy for the Hepatitis dataset after using a missing value handling algorithm. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.

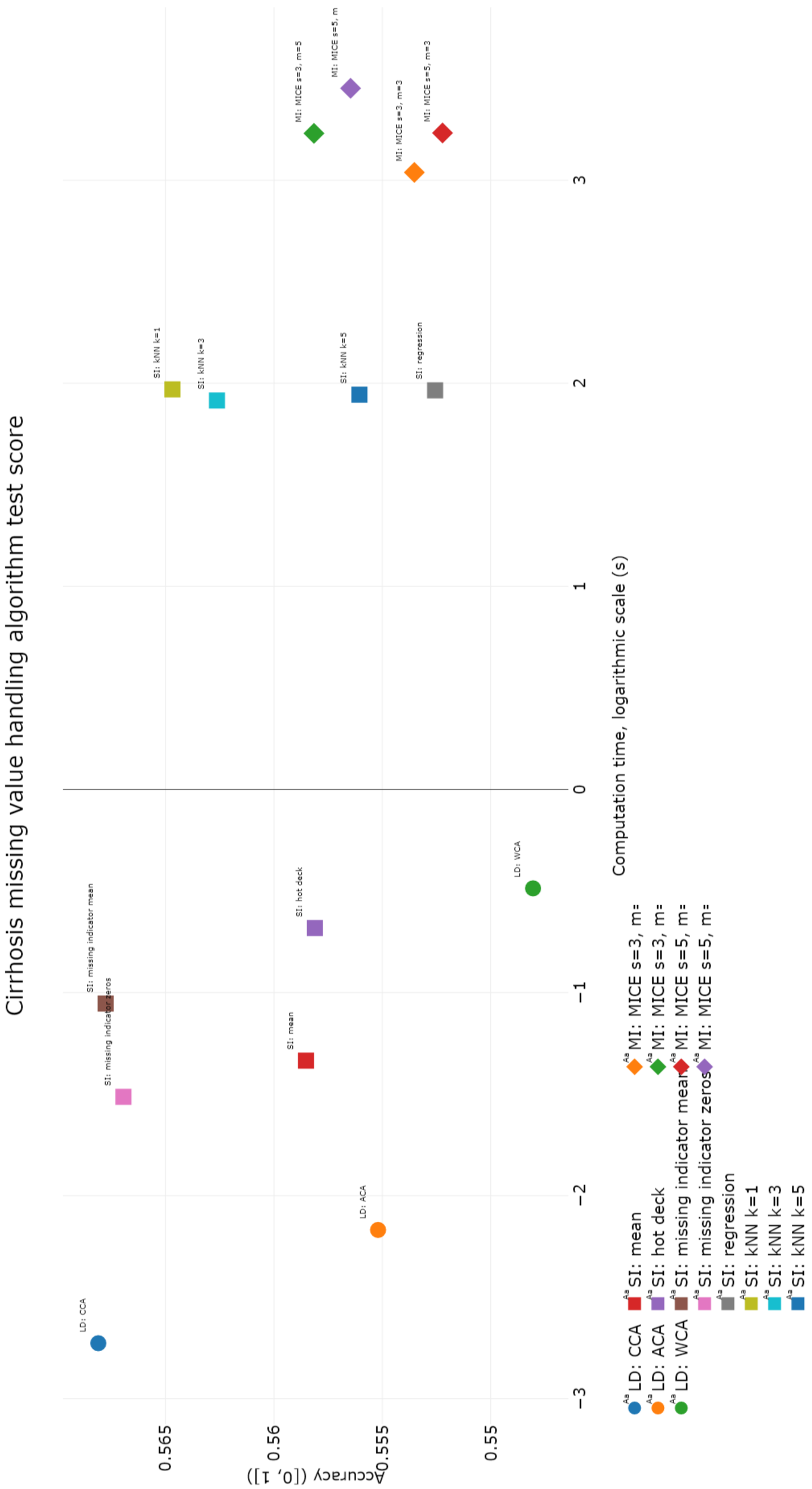

Figure 10: A plot showing the accuracy for the Cirrhosis dataset after using a missing value handling algorithm. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.

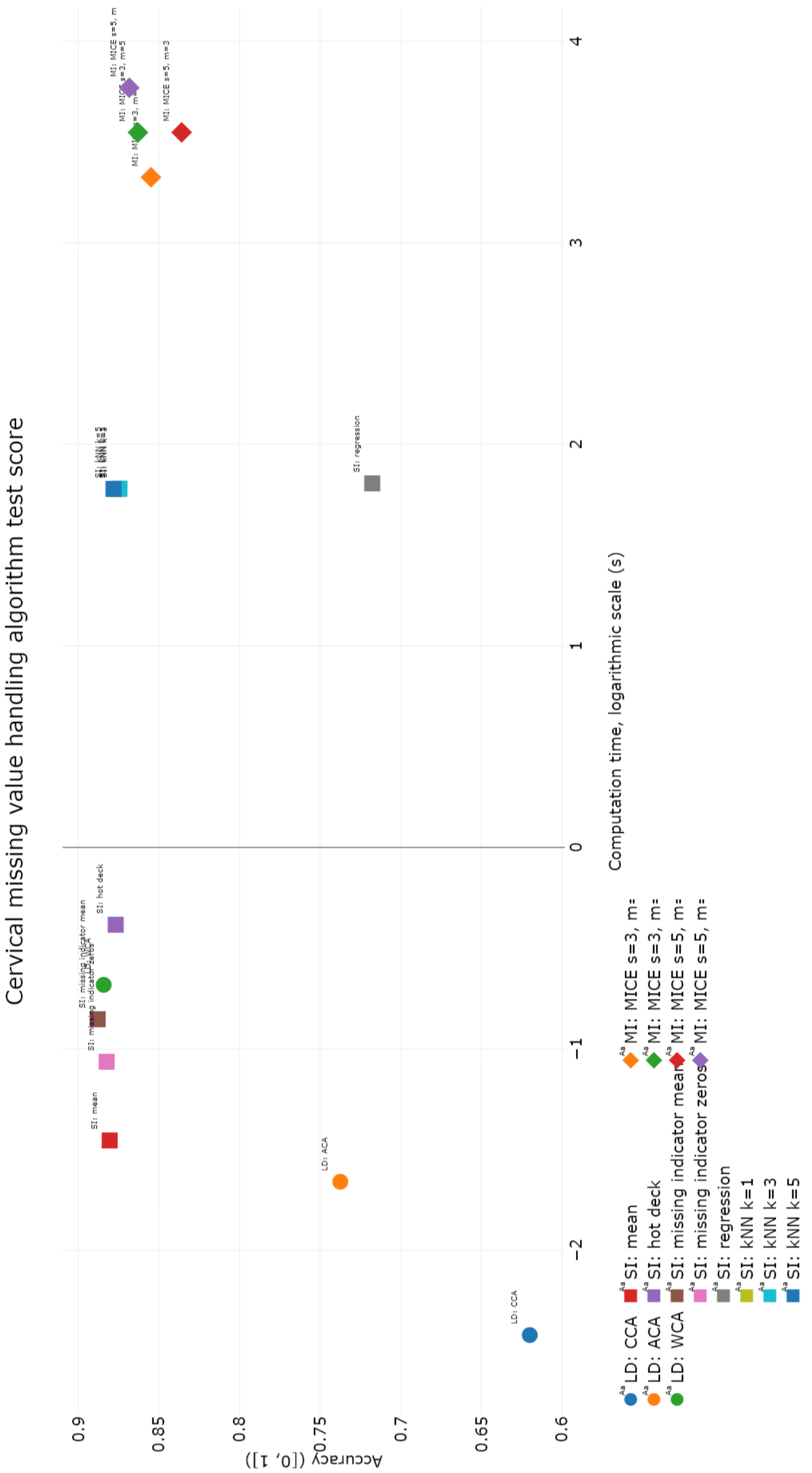

Figure 11: A plot showing the accuracy for the Cervical cancer dataset after using a missing value handling algorithm. The x-axis shows the computation time for the missing values, done on a 10-logarithmic scale due to the big differences between them. The y-axis shows the accuracy on a scale of 0 to 1. The explanation of every data point can be found both in the text on the top right of the data point as well as in the legend.