technische universität
dortmund

Estimating the Contaminationrate in unsupervised Anomaly Detection

Meinschien, Finn

Bachelorarbeit

Studiengang: Bachelor Informatik
Matrikelnummer: 235399
Erstgutachter: Prof. Dr. Emmanuel Müller
Zweitgutachter: Dr. Simon Klüttermann

Bearbeitungszeit: 17.06.2025 – 16.10.2025

Lehrstuhl für Data Science and Data Engineering
Fakultät für Informatik

Zusammenfassung

Anomalieerkennungsalgorithmen geben normalerweise nur einen Wert aus, der angibt wie wahrscheinlich der Datenpunkt im Vergleich zu den anderen Datenpunkten eine Anomalie ist. Die absolute Wahrscheinlichkeit, ob ein Datenpunkt eine Anomalie ist, ist jedoch häufig unklar, genau wie die Anzahl der Anomalien. Diese Arbeit stellt einen Algorithmus vor, der die Kontaminationrate, also den Anteil der Datenpunkte, die Anomalien sind, schätzt.

Der Algorithmus nimmt an, dass die Testdaten eine Mischung aus normalen Datenpunkten und Anomalien sind. Der Algorithmus nimmt verschiedene Kontaminationsraten an und berechnet eine dazugehörige Mischung aus der Verteilung von den normalen Daten und den Anomalien. Diese gemischte Verteilung wird dann mit den Testdaten verglichen. Die Kontaminationsrate, bei der diese Verteilung am ähnlichsten zu den Testdaten ist, ist die geschätzte Kontaminationsrate. Es werden verschiedene Versionen des Algorithmus benutzt: Eine nutzt direkt die zuvor genannte Variante: Eine andere berechnet erst die Anomalie-Scores der Datenpunkte und führt den Algorithmus auf den Anomalie-Scores aus.

Die Tests des Algorithmus zeigen, dass der Algorithmus selbst dann gut funktioniert, wenn die Art der Verteilung falsch identifiziert wurde. Die Genauigkeit ist sehr stark abhängig von der Anzahl der Datenpunkte. Das inverse des durchschnittlichen absoluten Fehlers ist proportional zu der Datengröße hoch 0,77 bei der multivariaten Variante. Bei der Variante mit den Anomalie-Scores ist die Potenz 0,59. Die Genauigkeit hängt auch von der Kontaminationsrate selbst ab: Die multivariate Variante liefert bessere Ergebnisse bei großen Kontaminationsraten, während die Version mit den Anomalie-Scores besser bei kleinen Kontaminationsraten abschneidet. Bei den Tests auf dem ADBench-Datensatz liefert die Version mit Anomalie-Scores gute Ergebnisse. Die multivariate Variante schneidet dagegen nicht so gut ab.

Abstract

Standard anomaly detection algorithms usually provide anomaly scores for data points relative to the other data points in the data set. However, the absolute probability of a data point being an anomaly or the number of anomalies is often unknown. This thesis proposes an algorithm that estimates the contamination rate ν, the share of data points which are anomalies.

The algorithm assumes that the test data set is a mixture of normal data points and anomalies which each follow their own distribution. The algorithm hypothesizes different contamination rates and calculates a distribution of the data points for each contamination rate. This distribution is compared to the test data. The contamination rate at which the distribution closest matches the test data is the resulting estimate. Two versions of this algorithm are proposed: One version directly estimates the contamination rate using the method just described. The second version first generates anomaly scores on which this algorithm is performed.

The experiments conducted show that the algorithm is potentially robust against misidentification of the distribution type of the anomalies. They show that the inverse of the mean absolute error is proportional to the size of the data set to the power of 0.77 for the multivariate version. This power is 0.59 for the version using anomaly scores. The proposed versions perform differently depending on the contamination rate: The multivariate version performs better at high contamination rates while the version using anomaly scores performs better at low contamination rates. The version with anomaly scores performs well on the ADBench data set. The multivariate versions performance is not as good.

Contents

List of Figures

1 Introduction

Anomaly detection, sometimes referred to as outlier detection, is an important problem in data analysis. Its significance is underscored by its wide range of practical applications, including fraud detection in financial transactions [HGY22], fault detection in industrial systems [Mil11], network security [XJJL+22], and medical diagnostics [FGD+21]. In these domains, the ability to accurately identify rare or unusual events can have a substantial real-world impact, such as preventing financial loss, ensuring system reliability or diagnosing diseases.

However, one challenge continues to exist: Most methods provide only relative anomaly scores, lacking a direct estimation of the absolute probability that a given data point is anomalous. This limitation complicates the interpretation of the anomaly scores. There are some methods which convert anomaly scores into probabilities [GT06]. However, there is no standard approach to evaluate these probabilities [RMCZ24].

This thesis provides an algorithm that estimates the contamination rate (ν), the proportion of the anomalies in a data set. This enhances the interpretability and practical utility of the anomaly detection scores. The proposed approach hypothesizes different contamination rates and models the data as a mixture of normal and anomalous distributions and selects the rate that best matches the observed data distribution.

Firstly, this thesis gives background information about anomaly detection and the methods used in this thesis. Then the algorithm is presented. Afterwards, the performance of the algorithm is evaluated: The influence of different parameters on the performance is measured on synthetic data sets. In the next step, the algorithm is evaluated on the real-world benchmark data set ADBench [HHH+22]. Finally, a conclusion and an outlook is given.

2 Background and fundamentals

2.1 What is anomaly detection?

2.1.1 What are anomalies and why are they important?

In order to effectively discuss anomaly detection a definition of anomalies is paramount. One such definition is given in 'Anomaly Detection: A Survey' [CBK09]: 'Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains.'

The gap to the expected behavior is often due to a different origin of the anomalies. For example, in a medical context the normal data could represent the clinical data of healthy individuals. Anomalies can represent the data of someone with a disease [WMSS93]. One of those examples is shown in Figure 1. A subset of the data is shown where 2 out of 30 dimensions are chosen. The data distribution shows a clear difference between the normal data and the anomalies.

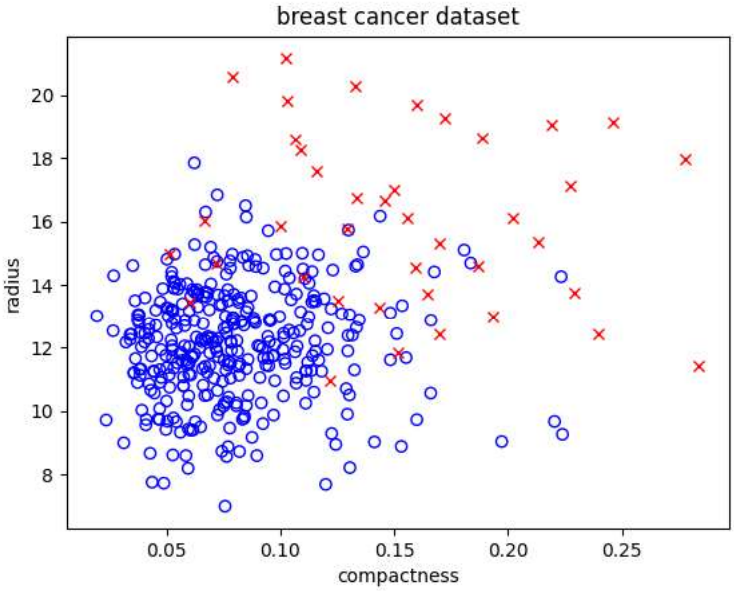


Figure 1: Two dimensions of a subset of a breast cancer data set [WMSS93], blue circles are normal data, red crosses are anomalies

This shows the importance of detecting anomalies as they often correspond to rare events where immediate action is necessary. Other such cases include:

- fraudulent financial transactions [HGY22]
- network intrusion [XJJL+22]
- system failure [Mil11]

This demonstrates the demand for anomaly detection as it can help to uncover these events and lead to a quicker response where the anomalies might otherwise continue to be unnoticed.

2.1.2 Detection methods

This section provides examples of approaches to detect anomalies. A lot of common anomaly detection methods are distance based anomaly detection methods. As the name suggests, these methods use the distance between data points to detect anomalies. They include:

- k-nearest neighbors [RRS00]:

  The k-nearest neighbor algorithm can be used for anomaly detection. For each data point the k nearest neighbors are calculated where k is a hyperparameter which is chosen beforehand. There are multiple ways to calculate an anomaly score from these distances: For example, the distance of the k-th nearest neighbor can be chosen or the average distance of the k nearest data points is taken.

- local outlier factor [BKNS00]:

  The local outlier factor (LOF) compares how close a data point is to its neighbors compared to how close these neighbors are to their neighbors. Firstly, the reachability-distance is calculated: $reachability - distance_k(A, B) = max(k - distance(B), d(A, B))$ with d being the distance and the k-distance being the distance to the k-th nearest neighbor. This is used to calculate the local reachability density $lrd_k(A) = \frac{|N_k(a)|}{\Sigma_{B \in N_k(A)} reachability - distance_k(A,B)}$. This is the inverse of the average reachability distance of A from its neighbors. Finally, the local outlier factor is calculated: $LOF_k = \frac{1}{|N_k(a)|} \Sigma_{B \in N_k(A)} \frac{lrd_k(B)}{lrd_k(A)}$. A value higher than 1 signifies a data point with a lower density than its neighbors. Therefore, data points with a high LOF are more likely to be anomalies.

2.2 Isolation Forest

This section focuses on the isolation forest [LTZ08] in detail as it is one of the most widely used and fastest anomaly detection algorithms and is used prominently in this thesis. Isolation forests can also have approximation qualities of the underlying probability distribution [BHM20].

The basic idea of isolation forests is that anomalies are rare and different and therefore are more likely to be isolated. Isolated in this case means separated from the majority of the data set.

The algorithm does this by creating an ensemble of isolation trees which are constructed in the following way:

1. A split dimension and a split value of that dimension are randomly selected.
2. The data points are split into a left and a right subtree depending on whether the value in the split dimension is lower or higher than the split value.
3. This process is recursively repeated with both subtrees unless at least one of the following three conditions is met: i) the isolation tree reaches its height limit, ii) the subtree contains only one data point or iii) all data points have the same value in every dimension.

The result is a tree where all data points have their own leaf unless there is a data point with all the same values or the leaf is at the height limit. This gives each data point the height of its leaf as a value. Since anomalies are more likely to be isolated, they are more likely to split of from the majority of the data points earlier and are therefore more likely to have a lower height. The anomaly score s of a data point x is then calculated with the following formula:

$$s(x, n) = 2^{-\frac{\mathbb{E}(h(x))}{c(n)}} \tag{1}$$

n is the number of internal nodes of the tree, h(x) is the height of the tree and $c(n) = 2H(n-1) - (\frac{2(n-1)}{n})$ with H(x) being the harmonic number.

2.3 Kolgomorov-Smirnov test

The Kolgomorov-Smirnov test (KS test) [AE11], which is used in this thesis, is a statistical test which can be used to estimate whether two data sets originate from the same distribution. It can also be used to estimate the probability of a set of data points originating from a specified distribution.

The KS test compares the empirical cumulative density functions of the data sets or the empirical cumulative density function of the test data with the cumulative density function of the distribution. The test statistic D is the maximum difference between these functions. This statistic is used to calculate the probability that the two samples originate from the same distribution (or that the sample matches the distribution).

2.4 Maximum likelihood estimation

Maximum likelihood is another method used in this thesis. The goal of maximum likelihood estimation [Myu03] is to estimate the parameter of a distribution which has the highest likelihood to produce a data set. This is usually done by estimating the parameter with the highest log-likelihood function. The logarithm is strictly monotonically increasing and therefore does not change the result. However, it does convert the product of the likelihood of each data point into a sum, making it easier to use. The parameter θ where $l(\theta; x) = \Sigma ln(f(x_i|\theta))$ is maximized is the parameter with the highest likelihood.

2.5 Related works

2.5.1 Tukey fences

One criterion to determine whether a data point is an anomaly are Tukey fences [Hoa03]: Any data points lower than Q1-k*IQR or higher than Q3+k*IQR are considered to be outliers. Q1 and Q3 are the quartiles and IQR is the interquartile range, the difference between the two quartiles. k is a contstant which is typically set to 1.5. This is a good and broad measure when nothing is known about a data set, however, it has a few limitations: It is only applicable on 1-dimensional data. It does not take the shape of the data into account. This may lead to scenarios where 20% of a data set is outside 1.5 times the IQR and considered to be anomalous even though all data points outside of the range are all close to each other.

2.5.2 Threshholding using the training data

One common approach used to decide whether a data point is an anomaly is to consider any data point an anomaly, if it has a higher anomaly score than a predefined fraction of the training data. This is often a default in one of the most common anomaly detection libraries [ZNL19]. The library was also used for this thesis. This method is simple to implement. However, it has drawbacks: The share of data points being labeled anomalies will nearly always be higher or equal to the predefined fraction. For example, if there are no anomalies in the test data set, the algorithm will label a share equal to the predefined fraction as anomalies. If the fraction is set too low, the algorithm will miss anomalies.

2.5.3 Estimating the Contamination Factor's Distribution in Unsupervised Anomaly [PBK23]

One paper which estimates the contamination rate is 'Estimating the Contamination Factor's Distribution in Unsupervised Anomaly' [PBK23]. In the paper, the authors use different anomaly detectors to map the data points into a M-dimensional space where M is the number of detectors. The data points are modeled with the use of different mixture components and ordered by these components . For each component, the probability of its anomalousness is calculated by using the probabilty that all components, which are more anomalous, are anomalies and the conditional probability that this component is anomalous, if all other more anomalous components are also anomalous. These are used to calculate the posterior distribution.

3 Methology

3.1 Idea of the algorithm

A data set used for anomaly detection consists of two types of data points: the normal data and the anomalies. As a consequence, the distribution of the data set p_t can be modeled as a mixture of the distribution of the normal data p_n and the distribution of the anomalies p_a. This mixture has the form $p_t = (1 - \nu) * p_n + \nu * p_a$.

This results in the mean having a similar equation:

$$\mu_t = \upsilon\mu_a + (1 - \upsilon)\mu_n \tag{2}$$

The covariance of the total data can be calculated by using a combination of the means, the covariance of the normal data and anomalies and the contamination rate. Some of these variables can be estimated from the training and test data:

- the mean μ_n and covariance Σ_n of the normal data can be estimated from the training data
- the mean μ_t and covariance Σ_t of the total data can be estimated from the test data
- only the mean μ_a and covariance Σ_a of the anomalies are completely unknown

The relationship between the contamination rate and the mean and covariance of the anomalies results in the knowledge of the contamination rate directly leading to the mean and covariance of the anomalies.

The idea of the algorithm is to use this in the following four step process:

1. Hypothesize a candidate contamination rate ν_{guess}
2. Calculate the mean μ_a and covariance Σ_a corresponding to ν_{guess}
3. Predict the combined distribution using these variables
4. Compare the predicted distribution to the test data

For each candidate contamination rate, this process produces a comparison score. These scores are then compared themselves. The candidate contamination rate corresponding to the best comparison score is the final estimated contamination rate.

3.2 Hypothesizing the candidate contamination rate

The first step of the process is the hypothesis of the contamination rate.
The possible range of the contamination rate is (0,1). Grid search, testing values within the range in an interval, is a simple method for this task and the one chosen in the implementation of the algorithm for this thesis. Other methods like the usage of evolutionary algorithms can also be used.

3.3 Calculating mean μ_a and covariance Σ_a

The second step is the calculation of the mean μ_a and covariance Σ_a of the anomalies.

$$\mu_t = v\mu_a + (1-v)\mu_n \tag{3}$$

$$v\mu_a = \mu_t - (1-v)\mu_n \tag{4}$$

$$\mu_a = \frac{\mu_t}{v} - \frac{(1-v)\mu_n}{v} \tag{5}$$

The formula for the indices of the covariance matrix is more complex. Firstly, the definition of the covariance is used and the expected value replaced with its definition and the simple mean:

$$\begin{aligned} Cov(X_t, Y_t) &= \mathbb{E}((X_t - \mathbb{E}(X_t))(Y - \mathbb{E}(Y_t))) \\ &= \frac{1}{N}\Sigma_t((X_i - \mu_{Xt})(Y_i - \mu_{Yt})) \end{aligned} \tag{6}$$

The sum is split into the part with the anomalies and the part with the normal data:

$$= \frac{1}{N}(\Sigma_a((X_i - \mu_{Xt})(Y_i - \mu_{Yt})) + \Sigma_n((X_i - \mu_{Xt})(Y_i - \mu_{Yt}))) \tag{7}$$

The summed up terms are multiplied out:

$$= \frac{1}{N}\Sigma_a(X_iY_i - X_i\mu_{Yt} - Y_i\mu_{Xt} + \mu_{Xt}\mu_{Yt}) + \frac{1}{N}\Sigma_n(X_iY_i - X_i\mu_{Yt} - Y_i\mu_{Xt} + \mu_{Xt}\mu_{Yt}) \tag{8}$$

The summed up parts are modified in order to get a definition of the covariance:

$$= v\frac{1}{Nv}\Sigma_a(X_iY_i - \mu_{Ya}\mu_{Xa} + \mu_{Ya}\mu_{Xa} - X_i\mu_{Yt} - Y_i\mu_{Xt} + \mu_{Xt}\mu_{Yt}) \tag{9}$$

$$\begin{aligned} &+(1-v)\frac{1}{N(1-v)}\Sigma_n(X_iY_i - \mu_{Yn}\mu_{Xn} + \mu_{Yn}\mu_{Xn} - X_i\mu_{Yt} - Y_i\mu_{Xt} + \mu_{Xt}\mu_{Yt}) \\ &= v(Cov(X_a, Y_a) + \mu_{Xa}\mu Ya - \mu_{Xa}\mu_{Yt} - \mu_{Ya}\mu_{Xt} + \mu_{Xt}\mu_{Yt}) \\ &+(1-v)(Cov(X_n, Y_n) + \mu_{Xn}\mu Yn - \mu_{Xn}\mu_{Yt} - \mu_{Yn}\mu_{Xt} + \mu_{Xt}\mu_{Yt}) \end{aligned} \tag{10}$$

The means are simplified and ν is multiplied out:

$$= v(Cov(X_a, Y_a) + (\mu_{Xt} - \mu_{Xa})(\mu_{Yt} - \mu_{Ya})) + (1-v)(Cov(X_n, Y_n) + (\mu_{Xt} - \mu_{Xn})(\mu_{Yt} - \mu_{Yn})) \tag{11}$$
$$= vCov(X_a, Y_a) + v(\mu_{Xt} - \mu_{Xa})(\mu_{Yt} - \mu_{Ya}) + (1-v)(Cov(X_n, Y_n) + (\mu_{Xt} - \mu_{Xn})(\mu_{Yt} - \mu_{Yn}))$$

Finally, the covariance of the anomalies can be isolated:

$$vCov(X_a, Y_a) = Cov(X_t, Y_t) - v(\mu_{Xt} - \mu_{Xa})(\mu_{Yt} - \mu_{Ya}) \tag{12}$$
$$-(1-v)(Cov(X_n, Y_n) + (\mu_{Xt} - \mu_{Xn})(\mu_{Yt} - \mu_{Yn}))$$

$$Cov(X_a, Y_a) = \frac{Cov(X_t, Y_t)}{v} - (\mu_{Xt} - \mu_{Xa})(\mu_{Yt} - \mu_{Ya}) \tag{13}$$
$$-\frac{(1-v)(Cov(X_n, Y_n) + (\mu_{Xt} - \mu_{Xn})(\mu_{Yt} - \mu_{Yn}))}{v}$$

In the univariate case, this formula simplifies to:

$$\sigma_a^2 = \frac{\sigma_t^2}{v} - (\mu_t - \mu_a)^2 - \frac{(1-v)(\sigma_n + (\mu_t - \mu_n)^2)}{v} \tag{14}$$

If the hypothesized contamination rate is 0 or 1 the formula is not applicable. However a contamination rate of 1 would make μ_a and Σ_a equal to μ_t and Σ_t and for a distribution with contamination rate 0 μ_a and Σ_a are not needed. It is worth pointing out that the formulas can result in negative variances and covariance matrices which are not positive semidefinite. The reason for this is the fact that only candidate contamination rates are used and not necessarily the actual contamination rate. If the actual contamination rate is used, the variance will be positive and the covariance matrix positive semidefinite. Therefore, any negative variances or not positive semidefinite covariance matrices must be the result of an inaccurate candidate contamination rate and can be discarded.

3.4 Predicting the distribution

The third step of the algorithm is the prediction of the combined distribution. The normal data and the anomalies are predicted individually and are consequently combined. While estimates for each mean and covariance matrix exists, the type of distribution is unclear. In the practical case, the distribution type of the normal data can often be known. However, the distribution type of the anomalies is rarely known.

One version, called the multivariate version in this thesis, assumes that both distributions are Gaussian distributions as it is one of the most common distribution type. Even if the actual distribution is not Gaussian, the result of the algorithm can still be the actual contamination rate, as it would be the only one with the correct mean and covariance.

In the case of univariate data, the distribution of the normal data can be estimated by the empirical cumulative distribution function. To do so for the multivariate data, the dimensionality of the data can be reduced. In this thesis, it is achieved by using anomaly scores. This can have the additional benefit of changing the distribution of the anomalies in a way which is closer to a Gaussian distribution. This version will be referred to as the version using anomaly scores.

3.5 Comparing the prediction to the test data

Finally, the predicted distribution of each candidate contamination rate is compared to the test data. For the version using anomaly scores, the comparison is done with the Kolgomorov-Smirnov test. The test calculates the highest difference between the empirical distribution function of the test data and the predicted distribution. The candidate contmination rate corresponding to the distribution with the lowest difference to the test data is the final estimate.

Multivariate data can not be compared using the Kolgomorov-Smirnov test. Therefore, a different test is needed. The test chosen for the multivariate version is the maximum likelihood estimation. For each predicted distribution, the log-likelihood of the test data is calculated. The candidate contamination rate corresponding to the distribution with the highest log-likelihood is the final estimate.

4 Evaluation of the algorithm

4.1 Introduction to the evaluation

The objective of this chapter is to evaluate the performance of the proposed algorithm in various scenarios. The goal is to quantify the algorithms accuracy depending on the influence of key parameters such as the size of the data sets, the contamination rate and the distribution of the data sets.

The evaluation will be conducted on two kinds of data sets:

1. Synthetic Datasets: These data sets are generated in a controlled environment. This allows for manipulation and exact knowledge of the parameters.
2. Real-World Datasets: Data sets collected from real data [HHH+22]. These data sets are useful to evaluate the algorithm in a realistic environment.

The algorithm is primarily evaluated by the mean absolute error. Another metric used is the bias as it captures general tendencies of the algorithm.

4.2 Datasets

4.2.1 Synthetic datasets

Synthetic data sets are generated data sets. This offers a few advantages. They have clear characteristics [SB21] and are used to provide a controlled environment for the evaluation of algorithms. Another advantage of using generated data sets is the ability to manipulate the different parameters and the precise knowledge of those. This is a key tool in order to assess the impact of different parameters.

The synthetic data sets used in this thesis are constructed in the following way:

1. Definition of the parameters:
   - the distribution types
   - mean and covariance matrix of the distributions
   - the contamination rate
   - the size of the data set
2. Sample size calculation:
   $N_{anomaly} = v * N_{test}$
   $N_{normal} = N_{training} + (1 - v) * N_{test}$
3. Data Generation: $N_{anomaly}$ and N_{normal} data points of the chosen types with the respective chosen mean and covariance matrix are generated. The training data set is made up of $N_{training}$ normal data while the test data set contains the other $(1 - v) * N_{test}$ normal data and $v * N_{test}$ anomalies.

This thesis will use the following values, if the experiments do not define the parameters in another way:

- the data will follow a Gaussian distribution
- mean and covariance matrix of the normal data are:
$$\mu_n = \begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}, \Sigma_n = \begin{bmatrix} 1 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix},$$
- mean and covariance matrix of the anomalies are:
$$\mu_a = \begin{bmatrix} -1 \\ -2 \\ 0 \end{bmatrix}, \Sigma_a = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix},$$
- contamination rate is 0.1
- the data set size is 10 000 for the multivariate version and 5 000 for the version with anomaly scores

4.2.2 Real world datasets

The algorithm is also tested on real-world data sets. These data sets are often more complex and more diverse compared to the synthetic data sets. This is necessary in order to judge the algorithms practical performance.

The ADbench [HHH+22] data set was chosen for this purpose. It is a benchmark data set which was deliberately designed to evaluate the performance of anomaly detection algorithms. The data set consists of 57 real-world data sets from various topics. Each of those was chosen specifically for its suitability as part of a benchmark data set.

Each data set has the following structure:

- a training data set which contains only normal data
- a test data set where half of the data points are anomalies and half are normal data points
- a label of each data point of the test data set signaling whether a data point is normal or an anomaly

4.3 Performance metrics

In order to evaluate the performance of the algorithm, specific metrics are required. Two metrics are used in this thesis:

1. Mean absolute error:

   The mean absolute error measures how close the estimated and the actual contamination rate are to each other. It is the most important of the metrics as the primary goal is to produce an estimate that is as close as possible to the actual contamination rate. It is calculated in the following way:

   $$MAE = \Sigma|v_{estimate} - v_{actual}| \tag{15}$$

2. Bias (average difference): The bias measures the tendency of the algorithm to over- or underestimate the contamination rate. The formula for its calculation is the following:

   $$Bias = \Sigma v_{estimate} - v_{actual} \tag{16}$$

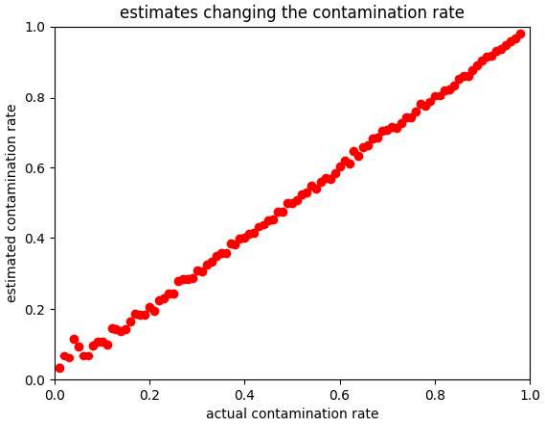
4.4 Testing on synthetic datasets

4.4.1 Impact of the size of the contamination rate

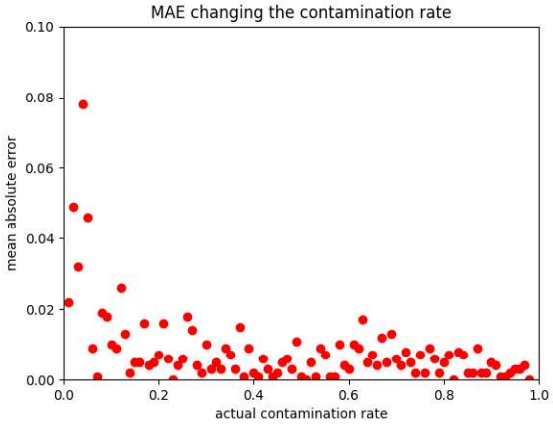
The contamination rate was the first parameter to be investigated. Since the contamination rate is the parameter on which the desired output depends, the needed change in output determines whether the algorithm has the flexibility necessary to be useful across a variety of contamination rates. In order to make this determination for the multivariate variation, the contamination rate was increased in increments of 0.01.

The results (Figure 2) have the following features:

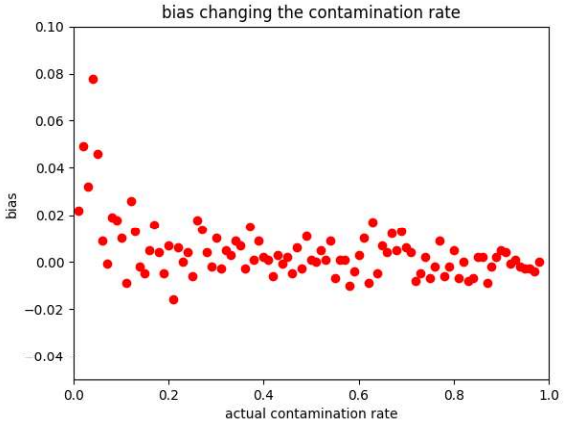
- The estimated contamination rates matches the actual contamination rate almost perfectly for $\nu > 0.2$ with mean absolute errors between 0 and 0.02
- for $\nu > 0.2$ the bias hovers around 0
- for $\nu < 0.2$ the mean absolute error and the bias increase



(a) estimates



(b) mean absolute error

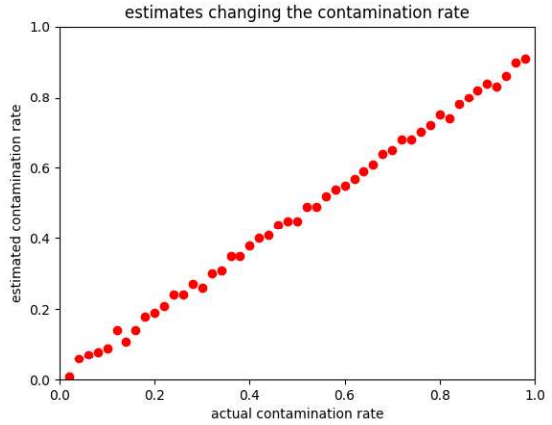


(c) bias

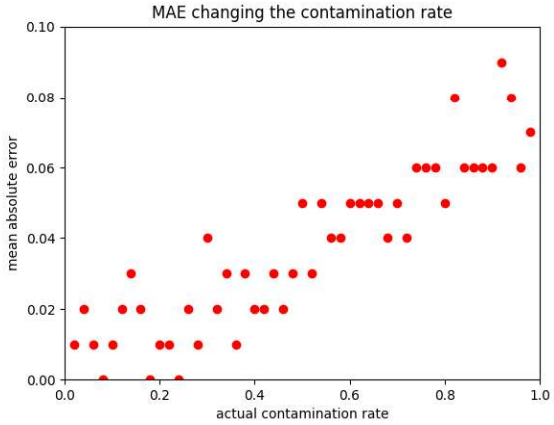
Figure 2: Estimates, mean absolute error and bias of the multivariate version at different contamination rates

The version using anomaly scores reacts slightly different to a changing contamination rate (Figure 3):

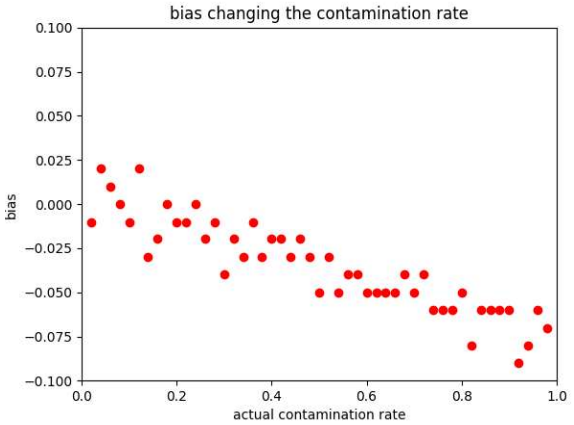
- the estimated contamination rate generally matches the actual contamination rate
- the mean absolute error seems to be increasing in a linear way with the contamination rate
- the bias is about zero for low contamination rates and decreases linearly



(a) estimates



(b) mean absolute error



(c) bias

Figure 3: Estimates, mean absolute error and bias of the version with anomaly scores at different contamination rates

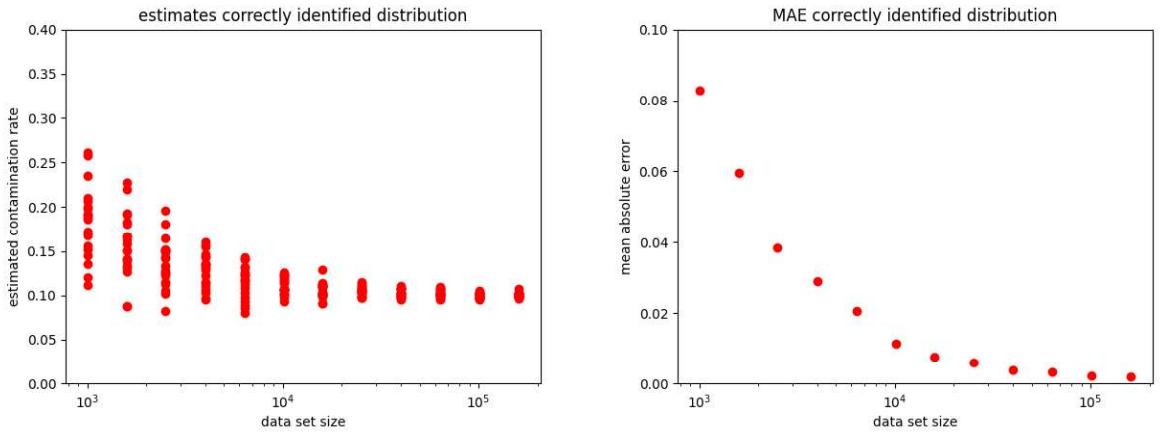
4.4.2 Impact of distribution misidentification

A critical assumption for the algorithm is the need to specify the type of distribution of the anomalies. This is rarely known in practical scenarios. Therefore, the performance of the algorithm in the case of misidentification is of great importance. This was tested on a generated data set where the normal data are correctly assumed to be Gaussian. However, the anomalies are assumed to be Gaussian when they are actually uniformly distributed.

The results (Figures 4 and 5) show that the error is significantly higher in the case that the distribution was misidentified. However, both errors are showing a similar pattern which converges towards 0 at large sample sizes and the error is just larger by a factor of about two.

One reason why the algorithm still works, even if the distribution type of the anomalies is misidentified is the following: At the correct contamination rate the mean and variance of the anomalies would still be correct. If the contamination rate would be estimated too high, the mean would be closer to the normal data. A low estimate would lead to an estimated mean of the anomalies farther away from the normal data. Usually the correct mean and variance lead to a more similar distribution even if the distribution is not the exactly the same .

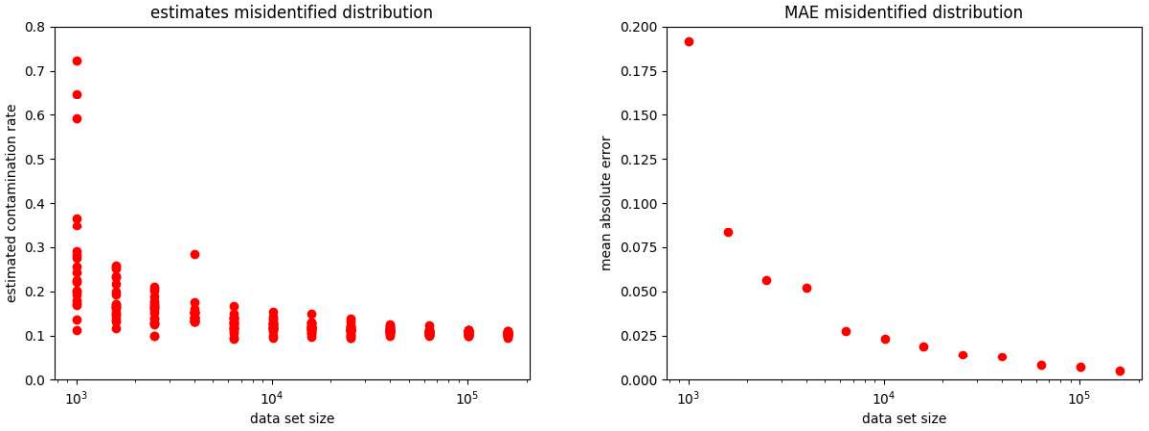
The fact that the algorithm performs well, even if the distribution type of the anomalies is misidentified is important for the practical use as the distribution type is rarely known. It shows that a correct identification helps the algorithm to be more precise, yet a misidentification is not detrimental to the performance.



(a) estimates

(b) mean absolute error

Figure 4: Estimates and mean absolute error of the algorithm when the distribution type is correctly identified



(a) estimates

(b) mean absolute error

Figure 5: Estimates and mean absolute error of the algorithm when the distribution type is misidentified as a Gaussian distribution instead of a uniform distribution

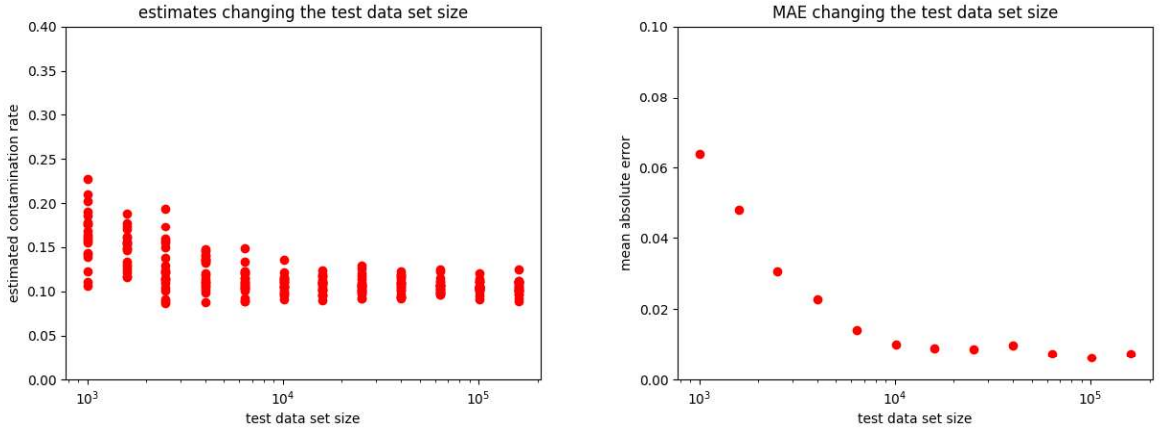
4.4.3 Impact of the size of the training and test data individually

This section investigates how the sizes of the training and test data sets affect the accuracy of the algorithm. To isolate the influence of each size, one was held constant while the other, and all other parameters, remained unchanged.

The results (Figures 6 and 7) show a similar pattern in both cases: When the variable data set size is less than the constant data set size, an increase in the variable size leads to a significant reduction in the mean absolute error. Once the variable size exceeds the constant data set size, further increases in size do not yield a significant reduction of the error any more.

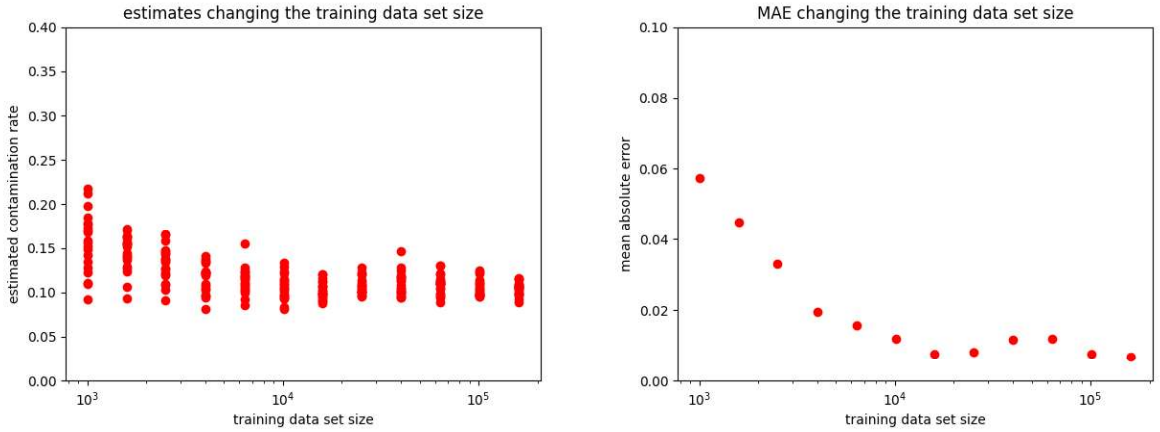
The reason for this phenomenon could be the way the error comes about. If the size of the training data is smaller than the size of the test data, the estimated mean and covariance of the normal data is likely more inaccurate than the estimated mean and covariance of the total data. If the size of the training data is larger, the estimated mean and covariance of the total data is more likely to be inaccurate.

These inaccuracies through this estimation are likely to combine in an additive way. The inaccuracies themselves on the other hand are likely to decrease slower at larger sizes. The result of this is that any increase in size of the smaller data set has a significant influence on the error while an increase of the larger one can be negligible.



(a) estimates (b) mean absolute error

Figure 6: Estimates and mean absolute error of the algorithm when only the size of the test data is changed

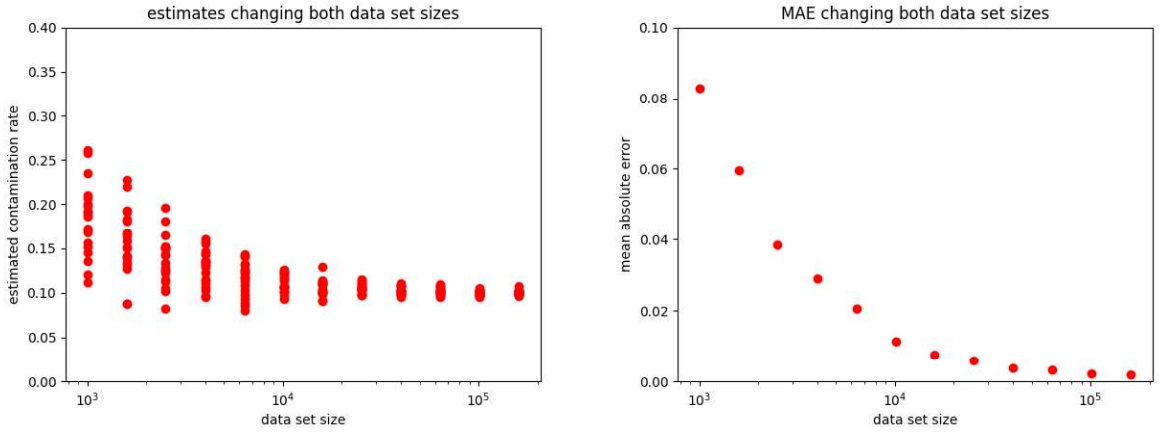


(a) estimates (b) mean absolute error

Figure 7: Estimates and mean absolute error of the algorithm when only the size of the training data is changed

4.4.4 Combined impact of the size of the training and test data

The previous experiment showed that increasing only one of the training or the test data sizes leaves the accuracy restricted by the other. In order to measure their impact without this restriction, both the training and the test data size were increased simultaneously. The results (Figure 8) show a monotone decrease in the mean absolute error. They show a smooth curve which can indicate a clear relationship between the size of the data sets and the mean absolute error.



(a) estimates

(b) mean absolute error

Figure 8: Estimates and mean absolute error of the algorithm when both the size of the training data and the size of the test data are changed simultaneously (same data as in Figure 4)

To further investigate the relationship between the data size and the mean absolute error, the logarithm of the data size and the inverted mean absolute error is taken. This makes it possible to perform linear regression to get the exponent. The result is an exponent of 1.3. Figure 9 shows that a plot of $ln(datasize)$ to $ln(\frac{1}{MAE^{1.3}})$ results in a good fit with a linear function with gradient 1. This suggest that the inverse of the mean absolute error, which can be seen as a measure of accuracy, is proportional to $N^{0.77}$. Fractional exponents are unusual, however, they are also found in the comparison of data set size and performance in transformer language models [KSMTHC+20]. This is also different from the expected power of 2 as the idea about this algorithm came during discussions about it [KM25].

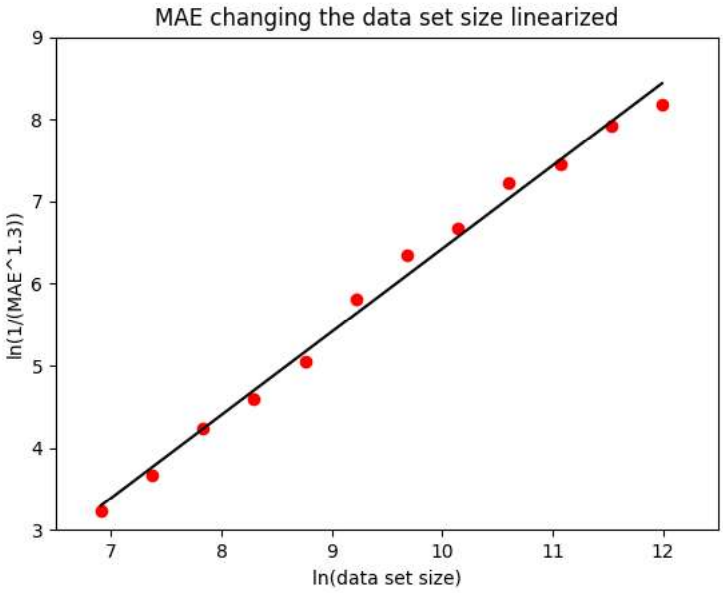
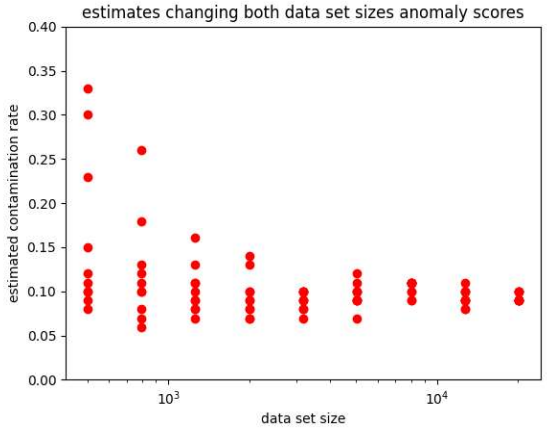
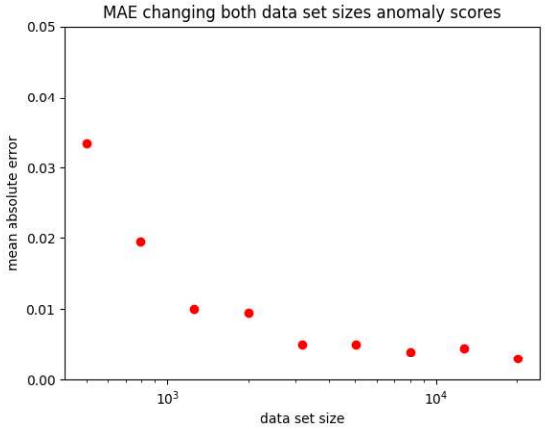


Figure 9: Linearized version of the mean absolute error when the training and test data are changed simultaneously

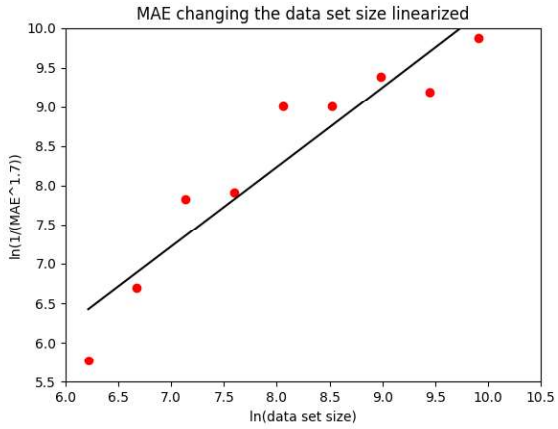
When the version with anomaly scores was used the results (Figure 10) showed a similar behavior with an exponent of 0.59. However, the fit was not as good. This can be due to the lower sample size used to test this version. The lower exponent as well as possibly the worse fit could be due to the isolation forest having an additional limiting factor.



(a) estimates



(b) mean absolute error



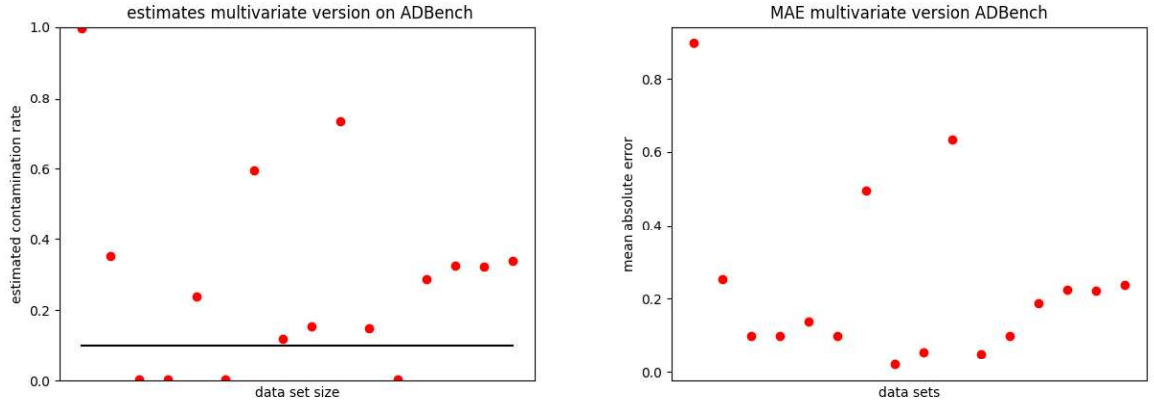
(c) linearized version

Figure 10: Estimates, mean absolute error and the linearized version of the version with anomaly scores when both the size of the training data and the size of the test data are changed simultaneously

4.5 Testing on real world datasets

The last test was conducted on the ADBench data set to test its performance on a realistic benchmark dataset. Since the algorithm needs enough data points to function properly, 16 data sets with at least 1000 training and test data points were used. Only the amount of anomalies needed for a contamination rate of 0.1 were use.

The estimates of the multivariate version were not very precise (Figure 10). This can most likely be attributed to the fact that the type of the distribution of the normal data was unknown and a Gaussian distribution was assumed.



(a) estimates

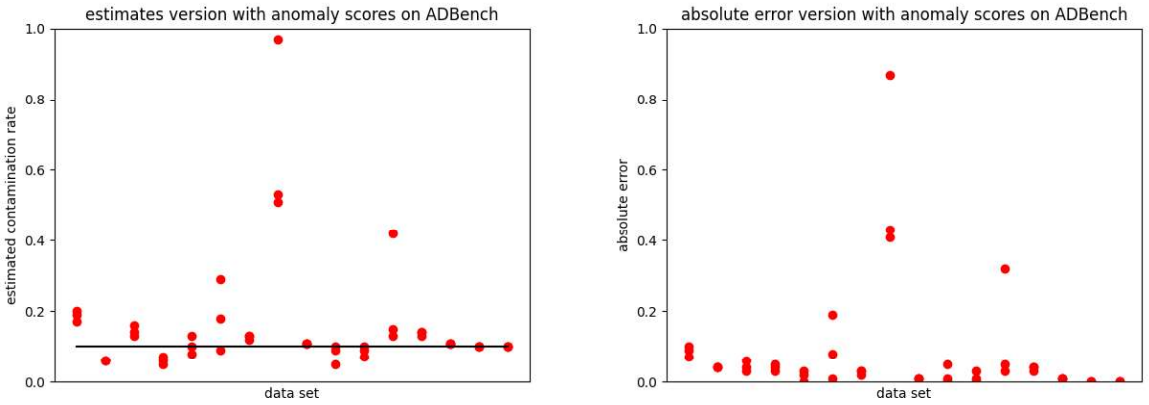
(b) mean absolute error

Figure 11: Estimates and mean absolute error of the multivariate version on different data sets from ADBench

The version with anomaly scores is far more accurate. This is likely due to the lack of need to guess the distribution type of the normal data. Most estimates are approximately at the correct contamination rate of 0.1. However, the algorithm consistently clearly overestimated the contamination rate of one data set (Adonor).

Interestingly, the estimations of each data set stayed consistent. This suggests that the accuracy is not much influenced by the changes through different anomaly scores by a new isolation forest. There can be multiple reasons how the errors can happen:

1. The anomaly scores of the anomalies were not distributed by a Gaussian
2. The data did not match the underlying distribution close enough due to the randomness in their creation
3. The training data still contained some anomalies leading to a lower estimation
4. Test data sets were mislabeled, leading to a higher estimation than 0.1 if anomalies were labeled as normal data and a lower estimation if normal data were labeled as anomalies



(a) estimates

(b) absolute error

Figure 12: Estimates and absolute error of the version with anomaly scores on different data sets from ADBench

5 Conclusions and outlook

The challenge addressed by this thesis is the inability of standard anomaly detection algorithms to estimate the contamination rate of a data set, hindering the interpretation of the anomaly scores usually produced by these algorithms. This was resolved through an algorithm which estimates the contamination rate of a data set. The algorithm works by hypothesizing a contamination rate and calculating a distribution associated with this contamination rate using the estimated distributions of the normal data and the anomalies. The contamination rate at which this distribution is closest to the test data is the final estimate.

The tests of this algorithm provided the following results:

- robustness to misidentification of the distribution type of the anomalies:

  Even if the the type of distribution of the anomalies is misidentified, the mean absolute error remains low.

- relationship of data set size and the mean absolute error:

  The mean absolute error is highly dependent on the size of the data set. In the multivariate version its inverse is proportional to the size of the data set to the power of 0.77. This power is 0.59 for the version using anomaly scores.

- accuracy depending on the contamination rate:

  For the two versions tested, the relationship to the size of the contamination rate is different: The multivariate version performs better at higher contamination rates and overestimates the contamination rate at lower rate. The version using anomaly scores performs better at lower contamination rates and underestimates the contamination rate at higher rates.

- results on real-world data sets:

  The version using anomaly scores shows a high accuracy on the ADBench data set. The multivariate version is not as accurate on the ADBench data set.

Overall, the results can be judged positively as the algorithm shows good results on synthetic data sets. The version using anomaly scores also shows good results on the ADBench data set while the multivariate version does not perform as good.

In the future, the algorithm can be improved by replacing the grid search used with a more efficient technique like an evolutionary algorithm. The possibility of using this algorithm for threshholding can be investigated.

References

[AE11] ARNOLD, Taylor B. ; EMERSON, John W.: *Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions*. 2011

[BHM20] BUSCHJÄGER, Sebastian ; HONYSZ, Philipp-Jan ; MORIK, Katharina: *Randomized outlier detection with trees*. 2020

[BKNS00] BREUNIG, Markus M. ; KRIEGEL, Hans-Peter ; NG, Raymond T. ; SANDER, Jörg: *LOF: Identifying Density-Based Local Outliers*. 2000

[CBK09] CHANDOLA, Varun ; BANERJEE, Arindam ; KUMAR, Vipin: *Anomaly Detection : A Survey*. 2009

[FGD+21] FERNANDO, Tharindu ; GAMMULLE, Harshala ; DENMAN, Simon ; SRIDHARAN, Sridha ; FOOKES, Clinton: Deep Learning for Medical Anomaly Detection – A Survey. In: *ACM Comput. Surv.* 54 (2021), Juli, Nr. 7. http://dx.doi.org/10.1145/3464423. – DOI 10.1145/3464423. – ISSN 0360–0300

[GT06] GAO, Jing ; TAN, Pang-ning: Converting Output Scores from Outlier Detection Algorithms into Probability Estimates. In: *Sixth International Conference on Data Mining (ICDM'06)*, 2006, S. 212–221

[HGY22] HILAL, Waleed ; GADSDEN, S. A. ; YAWNEY, John: Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. In: *Expert Systems with Applications* 193 (2022), 116429. http://dx.doi.org/https://doi.org/10.1016/j.eswa.2021.116429. – DOI https://doi.org/10.1016/j.eswa.2021.116429. – ISSN 0957–4174

[HHH+22] HAN, Songqiao ; HU, Xiyang ; HUANG, Hailiang ; JIANG, Minqi ; ZHAO, Yue: *ADBench: Anomaly Detection Benchmark*. 2022

[Hoa03] HOAGLIN, David C.: *John W. Tukey and Data Analysis*. 2003

[KM25] KLÜTTERMANN, Simon ; MÜLLER, Emmanuel: *Rare anomalies require large datasets: About proving the existence of anomalies*. https://arxiv.org/abs/2508.09894. Version: 2025

[KSMTHC+20] KAPLAN, Jared ; SAM MCCANDLISH TOM HENIGHAN, Tom B. B. ; CHESS, Benjamin ; CHILD, Rewon ; GRAY, Scott ; RADFORD, Alec ; WU, Jeffrey ; AMODE, Dario: *Scaling Laws for Neural Language Models*. 2020

[LTZ08] LIU, Fei T. ; TING, Kai M. ; ZHOU, Zhi-Hua: Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*, 2008, S. 413–422

[Mil11] MILJKOVIĆ, Dubravko: Fault detection methods: A literature survey. In: *2011 Proceedings of the 34th International Convention MIPRO*, 2011, S. 750–755

[Myu03] MYUNG, In J.: *Tutorial on maximum likelihood estimation*. 2003

[PBK23] PERINI, Lorenzo ; BÜRKNER, Paul-Christian ; KLAMI, Arto: *Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection*. 2023

[RMCZ24] RÖCHNER, Philipp ; MARQUES, Henrique O. ; CAMPELLO, Ricardo J. G. B. ; ZIMEK, Arthur: *Evaluating outlier probabilities: assessing sharpness, refinement, and calibration using stratified and weighted measures*. 2024

[RRS00] RAMASMAMY, Sridhar ; RASTOGI, Rajeev ; SHIM, Kyuseok: *Efficient Algorithms for Mining Outliers from Large Data Sets*. 2000

[SB21] STEINBUSS, Georg ; BÖHM, Klemens: *Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data*. 2021

[WMSS93] WOLBERG, William ; MANGASARIAN, Olvi ; STREET, Nick ; STREET, W: *Breast Cancer Wisconsin (Diagnostic)*. UCI Machine Learning Repository, 1993. – DOI: https://doi.org/10.24432/C5DW2B

[XJJL+22] XU, Wen ; JANG-JACCARD, Julian ; LIU, Tong ; SABRINA, Fariza ; KWAK, Jin: *Improved bidirectional gan-based approach for network intrusion detection using one-class classifier*. 2022

[ZNL19] ZHAO, Yue ; NASRULLAH, Zain ; LI, Zheng: PyOD: A Python Toolbox for Scalable Outlier Detection. In: *Journal of Machine Learning Research* 20 (2019), Nr. 96, 1-7. `http://jmlr.org/papers/v20/19-011.html`

Eidesstattliche Versicherung

(Affidavit)

Meinschien,Finn

Name, Vorname
(surname, first name)

235399

Matrikelnummer
(student ID number)

☒ Bachelorarbeit
(Bachelor's thesis)

☐ Masterarbeit
(Master's thesis)

Titel
(Title)

Estimating the Contaminationrate in unsupervised Anomaly Detection

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Dortmund, 16.10.2025

Ort, Datum
(place, date)

F. Meinschien

Unterschrift
(signature)

Belehrung:
Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin") zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:
Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Dortmund, 16.10.2025

Ort, Datum
(place, date)

F. Meinschien

Unterschrift
(signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**