Vrije Universiteit Amsterdam

Thesis

Outlier detection in datasets with mixed-attributes

Author:
Milou Meltzer

Supervisor:
Johan ten Houten
Evert Haasdijk

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

September 2015

"At any given moment, there is a certain percentage of the population that's up to no good."

J.E. Hoover

Vrije Universiteit Amsterdam

Abstract

Faculty of Sciences
Business Analytics

Master of Science

Outlier detection in datasets with mixed-attributes

by Milou Meltzer

Committing fraud is a financial burden for a company. Detecting fraud in an early stage can reduce financial and reputational losses. Some fraudulent behavior (e.g. credit card fraud) are assumed to be outliers in the dataset. This research proposes a new technique to detect outliers in mixed-attribute datasets.

The proposed outlier detection technique uses outlier detection for categorical and continuous dimensions separately. Infreqeunt pattern analysis is used for categorical datasets and k-medians clustering for outlier detection in continuous datasets. These two technique combined form the mixed-outlier detection algorithm, which was applied on the multiple subsets created by attribute bagging.

The proposed algorithm is validated based on the NSL_KDD dataset, which contains intrusions in a network trafic, and the Wisconsin Breast Cancer dataset. The different components of the algorithm were validated separately and combined for the NSL_KDD dataset. Based on the results we concluded that for this dataset the infrequent pattern analysis was most accurate in detecting outliers. Attribute bagging added value for the k-medians clustering. Infrequent pattern analysis and k-medians clustering were compared by validating these algorithms with the Wisconsin Breast Cancer dataset. These two algorithms show similar results.

Acknowledgements

This research is a final stage for obtaining my master's degree in Business Analytics at the Vrije Universiteit of Amsterdam. Deloitte Risk Services offered me the possibility of writing my thesis with their collaboration. Writing my thesis with the help of qualified employees of Deloitte Risk Services was an experience wherefore I am very grateful. Therefore, I want to thank Deloitte Risk Services and all the specialists that helped me with this research.

Special thanks to Johan ten Houten and Evert Haasdijk who guided me during this research and helped me with every bump in the road. Together we managed to finalize this research and were able to propose a new technique in the field of outlier detection.

Also, I want to thank Ruby Pouwels and George van Thoor for their support and help in finalizing this paper.

Chapter 1

Introduction

Committed fraud today is considered a serious financial burden to a company. Depending on the industry an organization is required to conduct an investigation, pay substantial fines, or is subject to other types of damages (e.g. reputation damage). Research shows that the cost of detected fraud on average is 5 percent of a company's annual revenue [1]. However, fraud is often never detected or under "lucky" circumstances a few years after the first event.

Almost every organization implements a form of an anti-fraud program in the attempt of fraud control and prevention. Anti-fraud programs in general consist of fraud policies, procedures and communication of these policies and procedures to employees [2]. Next to these business controls, fraud prevention also includes the implementation of checks built into software systems called IT-controls (e.g. enforced credit limits or technical implemented segregation of duties). Although, in practice the currently implemented controls by organizations show their weaknesses which still enable perpetrators in finding new methods to commit fraud. Because fraud prevention alone does not eliminate fraudulent behavior in a company, a company should also focus on fraud detection.

Detecting fraud is known as a time consuming effort. Early detection of fraud reduces the aggregation of unnecessary involved financial and reputational losses. To date most committed frauds were reported by whistleblowers, and afterwards detected by using conventional research methods. Therefore the use of data analytics in fraud detection is relevant and necessary.

Nowadays most companies are using Enterprise Resource Planning (ERP) systems to support, register and manage their day-to-day business activities. As relevant transactional and other data within these systems is being stored and logged the question arises if fraud detection is possible using data analytics in ERP systems.

Black's Law Dictionary defines fraud as "a knowing misrepresentation of the truth or concealment of a material fact to induce another to act to his or her detriment" [3]. The definition already explains that fraud is a comprehensive concept with different characteristics and therefore different analytical methods.

Fraudulent behavior is assumed to be a not normal operation. For example, transactions usually have similar characteristics, where patterns, like amount, bank-account, type of transaction, etc., can be grouped together. Fraudsters deviate from this pattern and are therefore not part of any group, and so outlying.

Outlier detection could be an innovative approach in detecting the existence of fraudulent records, because we expect that a fraudulent case or record is an outlier in the transactional dataset and that the "normal" cases are often similar to each other. Outlier detection can be both a supervised and unsupervised learning technique. This research focuses on unsupervised outlier detection, because in almost all real dataset fraud is an unknown case or record in the dataset.

Not all types of fraudulent behavior can be detected with outlier detection techniques. Due to seemingly normal behavior clustered in the dataset, it is possible that the actual outlier is not detected by the outlier detection method. For example, if the perpetrator processes a non-remarkable transaction amount to a non-blacklisted bank-account within a usual timeframe, the transaction will not be identified as irregular and therefore never marked as outlier. Fraudulent behavior which can be detected with outlier detection are: Credit Card fraud and intrusion in a network [4].

This research investigates an outlier detection technique, which can be useful in fraud detection and tries to find an answer regarding the following research-question:

What outlier detection technique can be used to detect outliers in mixed-attribute datasets?

This research-question is supported by a number of sub-questions:

- What are datasets with mixed-attributes?
- What is an appropriate technique for outlier detection in a continuous dataset?
- How are outliers detected in categorical datasets?
- What is an appropriate method for a high dimensional dataset with mixed attributes?

This thesis first discusses related work of other researchers and the application of these methods in our approach. Next, we introduce a new technique to detect outliers in

categorical attributes. Subsequently, it will combine related work and our new technique to detect outliers in mixed-attribute datasets. Finally, we will present our experimental setup and results.

Chapter 2

Related work

2.1 General

Outlier detection has been studied by several researchers. Researchers studied two types of outlier detection algorithms: supervised and unsupervised learning algorithms. Supervised learning algorithms detect outliers using labeled data, which means that records are classified as "normal" or "outlier". Whereas unsupervised learning algorithms use unlabeled data, which means that outliers (and normals) are unknown.

Supervised learning algorithms use examples and rules in order to determine what characterizes an outlier and what characterizes normal behavior. New observations are assigned to one of the two classes. Supervised learning algorithms are very popular in fraud detection [8, 9]. However, supervised learning algorithms require examples of both classes ("normal" and "outlier") and can merely detect type of outliers or frauds that exist in the training dataset [10]. Therefore, a new type of fraud (or outlier) is not likely to be recognized as an outlier. Additionally as real data might not be labeled, an investigation is necessary to classify the outlier and normal records to create labeled data for learning and validation purposes. Also, the identification of classes needs to be accurate, which is often not the case due to human interpretations. Besides, unbalanced ratio between the "normal" and "outlier" classes (e.g. scarcity of outliers) can cause misclassification [8].

Unsupervised learning algorithms are based on how the data is distributed instead of rules or examples. The distribution of data and the unsupervised algorithms identify outliers in datasets based on variation of this distribution. Therefore, this algorithm does not require labeled examples, which provides a more realistic process of fraud detection since this information is not available. However, Lazarevic and Kumar state

that using unsupervised learning the false positive rate could possibly increase because new (normal) data is recognized as outlier [11]. Unsupervised learning is based on two assumptions:

- the number of "normal" records is considerably higher than the number of "outlier" records;
- Outlying behavior can be separated from normal behavior [12].

According to K. Yamanishi et. al. unsupervised learning is technically more difficult [9]. This research focuses on unsupervised learning techniques. As stated, in fraud detection the fraudulent behavior is unknown and it takes time and effort to investigate these outlying examples that can be used in supervised learning. Besides, in comparison to supervised learning unsupervised learning has the benefit of detecting new outlier records which have not been discovered before.

Lazarevic and Kumar categorized four groups of unsupervised outlier detection techniques [11]:

- *Statistical approaches* assume that the underlying distribution of the data is known. Based on deviation of the distribution outliers are recognized. However, this type of technique has limitations on high-dimensional data, because finding the underlying distributions of each dimension is a complex exercise [4];
- *Distance based approaches* compute distances between points to detect outliers. This technique has less limitations than the statistical approaches [11]. Distance based models are a common technique in researching outlier detection of high dimension data [13]. However, other literature states that distances in high-dimensional data suffer from the "curse of dimensionality", where data points become equidistant and similar [14];
- *Profiling methods*: These techniques create profiles of normal-behavior using heuristics. Outliers are detected as deviations of the normal-behavior profiles [15]. Bolton and Hand investigated detection of credit card fraud using unsupervised profiling methods. They introduce an analysis where a peer group is created by measuring spending profiles [8].
- *Model-based approaches*: This category is characterized by predictive models to detect anomaly behavior. Examples of predictive models are replicator neural networks or support vector machines [11].

In this research statistical approaches are not considered, because the underlying distribution is unknown and hard to investigate for multiple dimensions. The profiling approaches are also not preferred, because there is no profile for true outliers. This is why we use a combination of the distance and model based approach. As mentioned, distance based approaches for high dimensional-data suffer from the "curse of dimensionality". Section 2.3 discusses a technique, called attribute bagging, which handles this problem and distances can be used for outlier detection in high-dimensional data.

2.2 Clustering techniques

Clustering techniques are unsupervised learning algorithms that classifies patterns in the data and groups patterns which form "clusters". Clustering algorithms are investigated by many researchers in different disciplines. Clustering techniques are also used in outlier detection. Often outlier detection algorithms use k-Means clustering [6, 7]. An alternative is k-median clustering.

The objective of a clustering technique is to find clusters in a dataset and although it may detect outliers, it is not its primary purpose. However, if a distance between a point and its centroid increases the probability of being an outlier increases. Therefore clustering can help detecting outliers in datasets.

2.2.1 k-Means clustering

k-Means clustering starts by randomly initializing the k centroids prior to the first iteration. During every following iteration all data points are assigned to their closest centroid followed by a calculation of a new mean of the cluster which becomes the new centroid. This process will be repeated until the centroid of the last iteration equals the outcome of the current iteration which declares the model as stable. The algorithm converges to a local minimum.

In general the algorithm has linear time complexity when the size of data increases. However, k-means clustering is not resistant to and highly sensitive for outliers. Based on these characteristics the algorithm cannot be perceived as robust [7, 16].

2.2.2 k-Median clustering

Whereas k-means clustering selects the mean of a cluster as centroid k-median selects the median. Therefore we can conclude that k-median clustering is a more robust

technique for outlier detection [17]. k-Median clustering minimizes the 1-norm distance of every point to its assigned cluster centroid [17]. The 1-norm distance is also called the Manhattan distance. It calculates the sum of differences between two vectors in a dimensional space [18]. The algorithm converges to a local minimum of the Manhattans distance between the centroid and its assigned points [19].

Definition 2.1. *Manhattan distance* calculates the sum of differences between two vectors u and v in dimension n [18].

$$MD(u, v) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.1}$$

In k-median clustering the number of clusters, k, is specified by the user. Many researchers have investigated how to select the appropriate number of clusters [20]. A method to set an appropriate k is based on the heuristic of minimizing the intra-cluster (within cluster) distance and simultaneously maximizing the inter-cluster distance (between cluster) [21]. A technique using this heuristic is the silhouette-index [22].

Definition 2.2. *Silhouette-index* maximizes the inter-cluster and minimizes the intra-cluster distances [21].

$$S(i) = \frac{b_i - a_i}{max(a_i, b_i)} \tag{2.2}$$

Where a_i is the average distance of point i to the points in its cluster and b_i is the average distance of point i to each point of the nearest cluster.

The range of the silhouette-index is $[-1, 1]$. If the Silhouette-index is close to 1, point i is assigned to a "correct" cluster. Point i is misclassified when the Silhouette index is near -1. A high average Silhouette-index value of all data points means that the data is clustered appropriately.

In this research we first tried means-clustering, however with extreme values in the outliers the model was really sensitive. Therefore, we use k-median clustering algorithm for outlier detection in continuous datasets, because in contrast to k-means this clustering method is more robust for outliers. This is an example of a distance based approach.

2.3 Attribute Bagging

In recent years the amount of data is increasing exponentially. The inherent consequence of this growth is an increased number of attributes. In order to reduce dimensionality researchers investigated methods to carefully select attributes and remove the irrelevant

or redundant attributes. In contrast to these regular approaches the research of Lazarevic and Kumar shows a method to handle the challenges of irrelevant or redundant attributes, called attribute bagging [11].

This principle of attribute bagging is derived from "bootstrapping". Bootstrapping is a statistical technique to measure accuracy. The bootstrapping technique creates many random samples of the dataset with replacement. The estimators of all random samples result in a confidence interval for the statistical estimator.

Attribute bagging has similarities with bootstrapping. In contrast to bootstrapping, attribute bagging is a technique used for sampling of the attributes without replacement. Researchers Lazarevic and Kumar prove that when a dataset has an increased number of irrelevant or redundant attributes it is difficult to detect true outliers, due to these noisy attributes [11]. Algorithm 1 shows the technique of attribute bagging.

Algorithm 1 Attribute Bagging algorithm

procedure ATTRIBUTE BAGGING(D, T, b)

subsets $\leftarrow NULL$

for t in 1,..,T **do**
selected_attributes = sample(1:numberOfAttributes, b)
subsets $\leftarrow$ Add subset of selected_attributes of D
end for

return *subsets*

end procedure

The attribute bagging algorithm creates many subsets by iteration of the following two steps [11]:

- Take a random sample of the attributes without replacement. A random sampled dataset D_t is created with all N records and a selection of $d \in b$ attributes in the tth iteration (b = maximum number of attributes in attribute bag).
- Use dataset D_t as input for the outlier algorithm, which creates a vector with outlier scores for all records based on the attribute subset D_t.

Lazarevic and Kumar discuss two techniques to combine the results of attribute bagging:

- *Breadth-First* is a simple method where the AS_t vector with the results for iteration $t \in T$ are ranked. The first element of the AS vectors, with the highest probability for that algorithm of being an outlier, are set on the $\{1, ..., t\}$-place of AS_{FINAL}

- *Cumulative sum* computes the sum of the outlier scores of record i of all algorithms and ranks these sums to create the AS_{FINAL}

This research proposes an outlier detection technique which calculates multiple outlier scores per attribute bagging iteration. The scores indicate the probability for an observation of being an outlier. In this research we use the Cumulative sum method to combine the results of the different iterations. The research of J. Gao et al shows that cumulative sum has more accuracy than Breadth-First, so in this research Breadth-First is not considered [23]. in chapter 5 we will further discuss the attribute bagging used in this research.

Chapter 3

Infrequent Pattern Analysis

In practice almost every dataset contains both continuous and categorical attributes, also called mixed attribute dataset. Outlier detection methods merely focus on continuous or categorical attributes. In order to create a dataset with a single attribute type, sometimes all categorical attributes are transformed into multiple binary attributes [24]. This implies that the data becomes high dimensional because every categorical attribute-value becomes a binary attribute, in particular when data contains categorical attributes having many different attribute-values. This method is not preferred because in high dimensional spaces the data points become equidistant and similar, which leads to the "curse of dimensionality" [14].

This section focuses on outlier detection in categorical attributes. The research of Otey et al. [25] and Koufakou [26] demonstrate a technique that searches for infrequent patterns in datasets, which is called infrequent pattern analysis. Infrequent patterns are combinations of categorical attributes that are irregular or outlying. Both researchers defined a categorical score for outliers based on the frequency or support. However, these researchers use every possible pattern combination and therefore, the amount of patterns grows exponentially with the number of attribute-values causing complexity and extensive processing time [27].

Important to the analysis of infrequent patterns is the frequency or support of an occurring pattern. First we introduce the definition of support.

Definition 3.1. *Support* is the frequency of an occurring pattern d. The function for support is defined as:

$$support(d) = \sum_{i=0}^{n} 1|x_i(D) = d \tag{3.1}$$

Where n is the number of records in the dataset, $d \in D$ is a categorical pattern where attributes occur only once and $x_i(D)$ is the categorical values of point x_i for attributes

D. If the categorical values of this point are equal to d, the support increases with factor 1.

The definition of support has the following properties:

(a) If an attribute-value is added to the pattern, the support cannot increase. For example, if pattern *a,b* has a frequency of 4, pattern *a,b,c* cannot occur more than 4 times. $\forall P : \forall Q \supseteq P : support(Q) \leq support(P)$ where P and Q are patterns [27];

(b) Supersets (e.g. I) of an infrequent pattern cannot be frequent. This follows from property (a) ;

(c) All subsets of a frequent pattern are frequent. This also follows from property (a).

Data points have an infrequent pattern in their categorical attributes when a co-occurrence of attribute-values is below a user specified threshold, σ. When the support of a pattern is below σ percent of n (number of records in entire dataset) the pattern is infrequent. The threshold is not changed during the infrequent pattern analysis. The research of Otey et al [25] and Koufakou [26] uses $\sigma = 5\%$. In the example in this section the used threshold is also $\sigma = 5\%$.

With infrequent pattern analysis x_i receives an outlier score based on the support of the pattern in the categorical attributes. If a pattern is infrequent, each superset of this pattern is also infrequent, see property (b). The research of Otey et al [25] and Koufakou [26] investigate all possible combination of attribute-values, which is time consuming and complex. Therefore, this research considered an alternative for the infrequent pattern analysis and proposes a new outlier detection technique for datasets with categorical attributes, *postponed outlier detection*. This technique is based on property (b), which makes it easier to process and is less time consuming.

3.1 Postponed outlier detection

Postponed outlier detection investigates infrequent patterns in the categorical attributes of the data. The postponed outlier detection algorithm creates an attribute tree. This tree is created with an iterative process, starting with a selection of first nodes where the attribute is least discriminative, see definition 3.2.

Definition 3.2. the *least discriminative attribute (LDA)* is the attribute that contains the maximum number of different attribute-values, where the minimum support of all

attribute-values is frequent.

$$LDA = \max_{0<a\leq|A|} L_a \tag{3.2}$$

Where L_a is the number of unique attribute values for attribute a and $|A|$ is the number of remaining categorical attributes in the dataset.

Due to this definition the outliers are detected in a deeper level of the tree, hence we call this postponed outlier detection.

With the definition of the least discriminative attribute the tree can be build. After finding the first least discriminative attribute, the next can be calculated. In this following step the remaining attributes are combined with the first Least Discriminative Attribute. Patterns are formed whereafter the support of every pattern is calculated and the next LDA is established based on the new patterns and their support. This process continues until the last LDA is added to the tree, see figure 3.1 for an example.

If an infrequent pattern presents itself in a tree, the superset of that infrequent pattern is also infrequent. Therefore, that branch of the tree will not be further investigated and the node is knotted from that point on. Figure 3.1 shows the knotted nodes with red labels of support. This is a pruning technique, which reduces the size of a tree and the run time of our algorithm.

3.1.1 Score

Until now, we have defined support, postponed outlier tree and infrequent patterns. With support, the (in)frequent patterns and the depth of the tree the proposed infrequent pattern analysis creates the outlier score for each record in the dataset. This research suggests the following outlier score for categorical dataset:

$$Score_1(x_i) = \sum_{|d|\geq 0} \frac{1}{support(d) \cdot |d|} \tag{3.3}$$

Where $d \in D$ is a categorical pattern where attributes are not duplicated and $|d|$ is the depth of the tree. d represents the minimum infrequent pattern following the LDA-tree. The score uses the minimum infrequent pattern, because the supersets of d are also infrequent and therefore not further investigated. The range of $Score_1$ is $[0, 1]$. If the score of x_i is close to 1, x_i has a high probability of being an outlier.

As already mentioned, the research of Otey et al [25] also calculates an outlier categorical score. However, they suggest that only infrequent patterns get a score > 0 and frequent patterns a score equal to 0. This research calculates a $Score_1$ for every record, regardless

FIGURE 3.1: set up of categorical attributes in a tree to find infrequent patterns. Attribute D is the first LDA, after combining attribute D with C, C is the LDA. The number in each node is equal to the support of the pattern, when this is marked red, it is an infrequent pattern. Remark that after the number is red, the tree is knotted because every superset is an infrequent pattern. In this figure $\sigma = 5\%$

of a record has frequent or infrequent patterns. This technique is applied in order to prevent all records from receiving a frequent pattern score 0, when the postponed outlier detection does not detect infrequent patterns. This will be further explained in chapter 4.

Algorithm 2 shows the postponed outlier detection technique in combination with the categorical scoring.

Algorithm 2 Postponed Outlier Detection

procedure POSTPONED OUTLIER DETECTION($D, sigma$)

$m_c \leftarrow numberOfCategoricalAttributes(D)$
$threshold = sigma \cdot length(D)$
$Score_1(X) = 0$ ▷ X: vector containing all records
$LDA \leftarrow NULL$
$patterns(x) \leftarrow NULL$
$X_{freq} = X$

for a in 1,..,m_c **do**
LDA = max(L_a)
patterns(X_{freq}) = patterns(X_{freq}) + x_{LDA}
support(X_{freq}) = support(patterns(X_{freq}))

if support(X_{freq}) $\leq$ threshold **then**
X_{infreq} = support($X_{freq} \leq$ threshold)
$Score_1(X_{freq}) = 1/(\text{support}(X_{freq}) \cdot a)$
$X_{freq} = X_{freq} - X_{infreq}$
end if

end for

$Score_1(X_{freq}) = 1/\text{support}(X_{freq} \cdot m_c)$

return $Score_1(x)$

end procedure

3.1.2 Example

Consider two points x_j and x_k and their categorical attributes m_{cj} and m_{ck} respectively. This examples uses a user specified threshold for infrequent patterns, $\sigma = 5\%$, the number of records, $n = 186$ and figure 3.1 is the LDA-tree. Point x_j has $\{D_3, C_2, B_2, A_3\}$ as categorical attributes and point x_k has $\{D_4, C_1, B_2, A_3\}$ as categorical attributes. If we look at the tree we see that $\{D_3, C_2\}$ is an infrequent pattern, see figure 3.1 node "Attribute $D3$" followed by node "Attribute $C2$" with a support of 5. Therefore, we can conclude that point x_j has an infrequent pattern. When looking at the path of record

x_k in the LDA-tree, we notice that x_k has only frequent patterns, see figure 3.1 node "Attribute D_4" followed by node "Attribute C_1, B_2 and A_3" with a support of 14. This support is above the threshold σ, which makes the pattern frequent. The outlier score for x_j is $\frac{1}{5 \cdot 2} = 0.1$ and for x_k is $\frac{1}{14 \cdot 4} = 0.018$. In this example, x_k has the lowest score and is considered more an outlier than x_j

Earlier research investigated infrequent pattern analysis for categorical attributes. This research uses the knowledge from the previous investigated infrequent pattern analysis, but proposes a new technique, called *Postponed Outlier Detection*. This new technique introduces a LDA-tree, which searches for infrequent patterns. An infrequent pattern occurs when the support is below a user-specified threshold, σ. Finally, an outlier score defines the probability of being an outlier for every record.

Chapter 4

Mixed-attribute outlier detection

Previous chapters discussed techniques that can be used for outlier detection in different datasets. Almost all datasets have both continuous and categorical attributes, also called mixed-attribute datasets. With infrequent pattern analysis we avoid the transformation of categorical attributes to binary attributes. In practice, continuous attributes can also transform to categorical ranges, but this will inevitable lead to loss of information. Therefore, different methods are applied on a the different subsets of the dataset.

Until now, this research discussed infrequent pattern analysis for datasets with categorical attributes and k-medians cluster for outlier detection in datasets with continuous attributes, chapter 2. But how can we combine the infrequent pattern analysis with the k-median clustering algorithm to create a mixed-attribute outlier detection technique? To answer this question this section proposes a mixed-attribute outlier detection method. In addition to the mixed-attribute outlier detection method, we also use the attribute bagging technique, discussed in chapter 2.

Many researchers investigated possible outlier detection techniques for categorical or continuous datasets only, but few detected outliers in mixed-attribute datasets. Researchers Otey et al [25] and Koufakou [26] have performed mixed-attribute outlier detection and suggest to calculate two different scores: one for categorical data (categorical score) and one for continuous data (continuous score).

4.1 Components of the mixed attribute outlier detection technique

Although both researches [25, 26] show satisfying results, this research suggests a different method, which will be explained below. The technique differs in three areas, which

are the three pillars or components of the proposed mixed-attribute outlier detection technique:

1. *Attribute bagging*
2. *Categorical Score*
3. *Continuous Score*

4.1.1 Attribute bagging

A way to handle the challenges of irrelevant or redundant attributes is using the attribute bagging technique. By using this technique, we do not carefully select attributes, but randomly create multiple subsets of the data (section 2.3). Attribute bagging is useful in (high-dimensional) datasets with many redundant attributes. This technique is not considered in the research of Otey et al and Koufakou. This research investigates the use of attribute bagging in combination with mixed-attribute outlier detection.

The attribute bagging algorithm, see algorithm 1, has the following parameters:

- T: total number of iterations;
- b: maximum number of attributes in a subset or so called "bag".

4.1.2 Categorical Score

The proposed categorical score of both researchers [25, 26] is a variant of the infrequent pattern analysis used in this research, chapter 3. The Postponed Outlier Detection technique calculates a categorical score. This technique is proposed because it takes the properties of the support in consideration, which generates a faster technique that detect outliers in datasets with categorical attributes.

The research of Otey et al [25] and Koufakou [26] calculate the categorical outlier score for a record with the support of infrequent patterns in the attribute-values. If a record does not have infrequent patterns, the categorical score is equal to 0. We decided to calculate a categorical score ($Score_1$) for every record, because we also use attribute bagging, which means that the results are handled a little different.

With attribute bagging the subsets can contain a various number of attributes up to b (maximum number in subset). In the situation that the attribute bagging algorithm

selects only few number of categorical attributes chances are that most patterns are frequent. When all records have frequent patterns, still the support of frequent patterns deviates. Therefore, we want to distinct a frequent pattern with a very large support from those with a support that is just above the σ-threshold. When the support is increasing, the $Score_1$ will approach 0.

All categorical scores are ranked in decreasing order, which will be discussed in section 4.2.

4.1.3 Continuous Score

Apart from categorical score, the records get also a continuous score for the dataset with continuous attributes. This continuous score is calculated with the use of k-Medians clustering technique, see chapter 2.

Earlier research [25, 26] calculated the continuous score based on the continuous similarities between records with similar categorical attribute-values. So, the researchers assume that records with similar categorical attribute values share similar continuous attribute values. We decided not to use this technique, because we don't want to make this assumption. Instead we use k-median clustering to calculate the continuous score, where the continuous records showing similar behavior are already clustered without considering categorical attribute-values.

When using k-Median clustering in outlier detection, a record with a large distance to its closest cluster is considered to be more outlying than a datapoint located perfectly within a cluster. Therefore, $Score_2(x_i)$ is defined as the Manhattan distance from a record to the closest centroid, see equation 4.1.

$$Score_2(x_i) = MD(x_i, c_k) = \| x_i - c_k \|_1 = \sum_{j=1}^{|m_q|} |x_{ij} - c_{kj}| \tag{4.1}$$

Where c_k is the closest centroid of point x_i and $|m_q|$ is the number of continuous attributes. $Score_2(x_i) \geq 0$. If $Score_2(x_i)$ increases, the closest centroid for record x_i is distant. This means that x_i has an increased probability of being marked as outlier.

4.2 Ranking

In order to create a final ranking, the above described components are combined in the outlier detection algorithms, called mixed-attribute outlier detection. The two scores,

$Score_1$ and $Score_2$, have both different ranges and cannot be compared. This is why we rank the scores, both in decreasing order. This results in two rankings per record: a rank for categorical attributes, and one for continuous attributes. Eventually, the final rank is equal to the average of the two rankings.

If datapoints have similar scores, called ties, the rank is calculated differently. For example, when two records are both ranked as 1, so actually they have rank 1 and 2 the rank is equal to $\frac{2+1}{2} = 1.5$.

Chapter 5

Experimental setup

5.1 Introduction

We implemented our algorithm using the *R* language. This is open source statistical language. We ran the experiment on a computer with 2.50 GHz processor and 16,0 GB of RAM.

Validating unsupervised learning algorithms is a challenge, because true-positives and false-negatives are unknown. However, thanks to two datasets with labeled data we are able to validate our proposed algorithm. These two datasets are the NSL_KDD dataset and the Breast Cancer dataset both from the UCI Machine Learning Repository [29].

These datasets are also used in other outlier detection research. The NSL_KDD dataset contains intrusions, where we assume that these intrusions are different (so outlying) from normal entrance. In the Breast Cancer dataset the malignant cases are supposed to be outliers.

The algorithm proposed in this research has multiple components to detect outliers in the dataset. It uses infrequent pattern analysis to detect outliers in categorical dimensions and k-median clustering technique in continuous dimensions. These two techniques are combined resulting in a mixed-attributes outlier detection algorithm. Ultimately and additionally mixed-attributes outlier detection is applied on all subsets resulting from the attribute bagging method. Each component of the algorithm is validated, namely:

- Infrequent pattern analysis with all categorical attributes;
- k-median clustering with all continuous attributes;

- mixed-attributes outlier detection with all attributes: infrequent pattern analysis and k-median clustering;
- attribute bagging with categorical attributes and infrequent pattern analysis;
- attribute bagging with continuous attributes and k-median clustering;
- full model: attributes bagging combined with mixed-attributes outlier detection.

5.2 Validation methods

5.2.1 Kendall's rank correlation

In order to validate our proposed algorithm we use the Kendall-rank correlation test. The Kendall-rank correlation test computes a coefficient, τ, which explains the correlation between two rankings.

Definition 5.1. *The Kendall τ coefficient* explains correlation between two rankings and is defined as [31]:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)} \tag{5.1}$$

where n is the number of records, C is the number of concordant pairs and D is the number of discordant pairs. The reach of the coefficient is $-1 \leq \tau \leq 1$. When the coefficient is close to 1 the two rankings are similar. When it nears -1 the rankings are perfectly disagreeing. When it is close to 0 the rankings are independent. Please note that this coefficient looks at the order of the list, not the exact number of the ranking.

The ranking of our algorithm is compared to a ranking of true-outliers, where the outliers are ranked as 1^{st} and the normals as 2^{nd}. This means that there are ties (values with a similar ranking) in the rankings. Kendall uses a different formula when ties exist in the rankings, this is called τ_b, [31, 32].

Definition 5.2. *The Kendall τ_b coefficient* explains correlation between two rankings with ties and is defined as [31]:

$$\tau_b = \frac{C - D}{\sqrt{(\frac{1}{2}(n(n-1)) - T)(\frac{1}{2}(n(n-1)) - U)}} \tag{5.2}$$

Where T is the number of ties in one of the rankings and U is the number of ties in the other ranking.

5.3 NSL_KDD dataset

The dataset from the third international Knowledge Discovery and Data mining tool competition of 1999 (KDD cup 1999 [28, 29]) is often used to validate outlier detection algorithms. The training dataset covers 7 weeks of network traffic. Multiple intrusions or attacks are simulated and added to the dataset. Depending if the activity is normal or a type of intrusion, each record is labeled accordingly.

The dataset contains 5 million records, 41 attributes and 1 attribute with the classification of the records (intrusion/attack or normal). The dataset allows us to validate our outlier algorithm. There are many critics regarding this dataset. Researchers state that the dataset suffers from redundant, duplicated records and other shortcomings. They created a new dataset, called NSL_KDD dataset [30]. In our research we use this renewed dataset to validate our algorithm.

This dataset has approximately 150,000 records and the same number of attributes as the KDD cup 1999 dataset. The dataset spotted 39 different types of attack, see appendix A for all types of attacks and their frequency. 52% of the dataset records are identified as normal activity, which means that 48% of the dataset are types of intrusions. This is insufficient for validation of an outlier detection algorithm, because intrusions are not rare in the dataset. Besides, 31% of the dataset records are identified as a "Neptune" attack. Therefore, the Neptune attack records do not qualify as outlier.

In order to validate the algorithm we selected 18 types of attacks with a low level of support, see table 5.1. The outlier algorithm is applied to 18 different datasets, where every dataset contains a type of intrusion combined with normal activity records. We decided to use 18 datasets so we can compare the results of our algorithm applied on 18 different sets and to make sure we have a thorough validation. Please refer to chapter 6 for the results.

5.4 Breast Cancer Wisconsin

For our second validation we used the Wisconsin breast cancer dataset [29, 33]. Many researchers use the Breast Cancer Wisconsin dataset as input for the validation of their outlier detection algorithm [34–37]. The dataset has 699 records with 458 (66%) labeled as benign and 241 (34%) as malignant. There are 10 different attributes available in the dataset. All attributes are categorical attributes: one attribute is the id number of a patient and the remaining attributes have a range between $1 - 10$ and are categorical indicators, see table 5.2.

TABLE 5.1: attacks and their occurrences

intrusion	**frequency**
spy	2
sqlattack	2
udpstorm	2
worm	2
xsnoop	4
perl	5
phf	6
xlock	9
ftp_write	11
loadmodule	11
imap	12
xterm	13
sendmail	14
ps	15
named	17
land	22
rootkit	23
multihop	25

TABLE 5.2: Attributes of Wisconsin breast cancer dataset

Attribute	**Values**
Sample code number	id number
Clump Thickness	1 - 10
Uniformity of Cell Size	1 - 10
Uniformity of Cell Shape	1 - 10
Marginal Adhesion	1 - 10
Single Epithelial Cell Size	1 - 10
Bare Nuclei	1 - 10
Bland Chromatin	1 - 10
Normal Nucleoli	1 - 10
Mitoses	1 - 10
Class	benign or malignant

5.4.1 Validate categorical outliers

Because of the categorical attributes, this dataset is suitable for validation of our postponed outlier detection algorithm based on infrequent pattern analysis. However, the 34% of the dataset is identified as malignant, it is likely that the malignant cases do not qualify as outlier. Therefore we take a sample of 458 benign and 24 (4.9% of the entire dataset) randomly picked malignant cases.

5.4.2 Validate continuous outliers

Although the attributes are categorized, the categories have numeric characteristics. For example, the clump thickness category 10 is larger than clump thickness category 1, so the distance between categories is explanatory. To compare the results of the infrequent pattern analysis with the k-median clustering algorithm, we also perform the k-median clustering algorithm on the Wisconsin breast cancer dataset.

5.5 Model validation

In order to validate whether our algorithm is robust, we use K-fold cross validation. In K-fold cross validation the data is split into K random samples of approximately the same length. The outlier detection technique is applied K times on $K-1$ random data samples. In every iteration a different sample is excluded. This results in K rankings of outliers where one value is missing, because each sample is excluded once in the K-fold cross validation. Figure 5.1 shows a representation of the K-fold cross validation. The figure shows that in every i^{th} iteration, where $i \in \{1, .., k\}$, the $k-1$ samples are used as training set and that the i^{th} sample is excluded.

Normally, k-fold cross validation use the training set to train the model. The excluded sample is used to validate the trained model. However, this research does not train a model on a dataset but simply detects outliers in the dataset. Therefore we use K-fold cross validation to validate whether a recognized outlier in sample A is also a present outlier in sample B.

With the Kendall correlation the K rankings are validated for correlation. If they are correlated, the algorithm is robust and an outlier is identified in different samples.

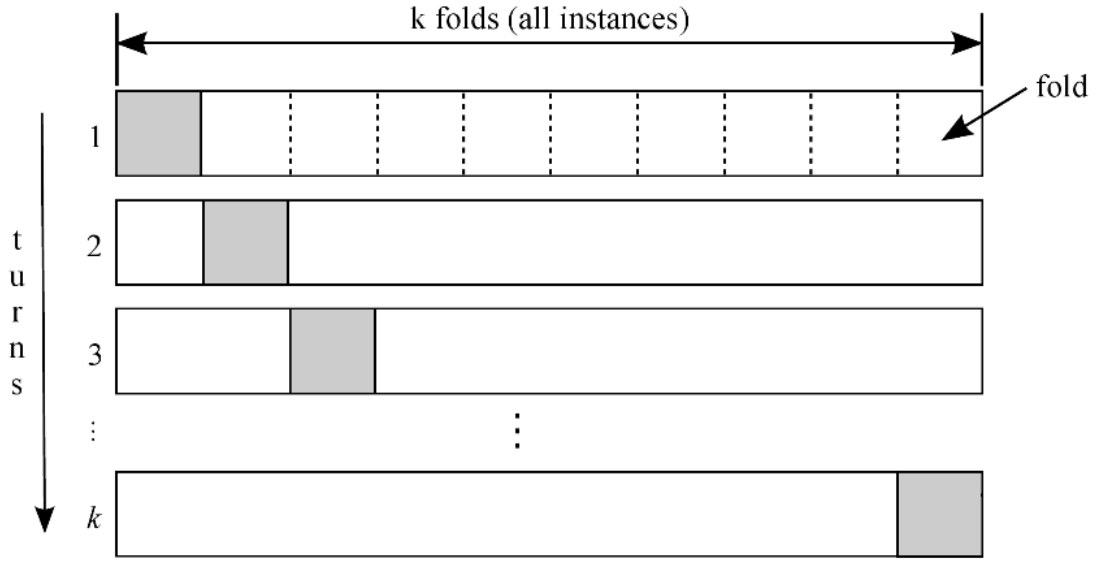

FIGURE 5.1: K-fold cross validation: each turn $k-1$ samples are used as input for the algorithm for validation.

Chapter 6

Results

6.1 NSL_KDD

As described in the experimental setup, our algorithm has 3 three different component: Attribute Bagging, Infrequent Pattern Analysis and k-median clustering. Every single component and possible combination of components is applied on the 18 subsets of the NSL_KDD dataset. Table 6.1 shows all results. The table has 7 columns:

- **Attack**: type of intrusion in the dataset (type of outlier);
- **Cluster**: performance of the k-median clustering algorithm;
- **IFP**: performance of the Infrequent Pattern Analysis;
- **Mixed**: performance of the mixed-attribute outlier detection technique, which is a combination of k-median clustering an Infrequent Pattern Analysis;
- **Cluster AB**: performance of the k-median clustering technique in combination with attribute bagging;
- **IFP AB**: performance of the Infrequent Pattern Analysis in combination with attribute bagging;
- **Mixed AB**: performance of the mixed-attribute outlier detection in combination with attribute bagging.

TABLE 6.1: Kendall correlation for every component of the algorithm

Attack	**Cluster**	**IFP**	**Mixed**	**Cluster AB**	**IFP AB**	**Mixed AB**
spy	0.002	0.008	0.006	0.007	0.007	0.007
sqlattack	0.001	0.008	0.007	0.006	0.007	0.007
udpstorm	-0.001	0.007	0.005	0.008	0.008	0.005
worm	0.006	0.008	0.007	0.006	0.006	0.004
xsnoop	0.006	0.012	0.010	0.012	0.012	0.008
perl	0.006	0.012	0.010	0.009	0.009	0.008
phf	0.007	-0.004	0.004	-0.004	-0.004	0.005
xlock	0.011	0.018	0.014	0.014	0.016	0.013
ftp_write	0.001	0.016	0.011	0.015	0.013	0.010
loadmodule	0.005	0.016	0.013	0.011	0.011	0.008
imap	0.008	0.021	0.015	0.019	0.020	0.017
xterm	0.014	0.018	0.016	0.012	0.016	0.016
sendmail	0.008	0.017	0.013	0.010	0.012	0.012
ps	0.002	0.020	0.013	0.016	0.016	0.015
named	0.009	0.022	0.018	0.023	0.022	0.015
land	0.000	0.027	0.019	0.029	0.029	0.024
rootkit	0.002	0.023	0.016	0.021	0.020	0.013
multihop	0.005	0.025	0.019	0.026	0.025	0.015

The results show that the Kendall correlation between the detected ranking and the true-rankings is relatively small for every component. This means that the two rankings are not correlated and that the outlier detection technique is not effective on these datasets.

However, comparing the clustering algorithm to the infrequent pattern analysis, the rank correlation of infrequent pattern analysis is higher. Therefore we can conclude that the infrequent pattern analysis is more accurate and has a better performance.

When the k-median clustering algorithm is combined with infrequent pattern analysis, the rank correlation is lower than applying the single infrequent pattern analysis in most datasets. This can indicate that the infrequent pattern analysis performs better individually.

When looking at the influence of the attribute bagging technique, we see that the rank correlation increases when the combination of attribute bagging and k-median clustering is applied. Based on this finding we conclude that the attribute bagging technique is adding value to the k-median clustering technique. Still, it does not extremely influence the infrequent pattern analysis.

Because of the differences between the infrequent pattern analysis and the k-medians clustering, we also performed a Kendall correlation test on the rankings of these two techniques. The result is that these rankings are not correlated, which means that infrequent pattern analysis detects different outliers than the k-median clustering algorithm. This seems logical, because the different type of techniques are applied on different datasets (categorical or continuous).

6.2 Breast Cancer Wisconsin

The Wisconsin Breast Cancer dataset is used to compare the Infrequent Pattern Analysis with k-median clustering, as explained in the experimental setup. The Wisconsin Breast Cancer dataset contains 10 categorical attributes. But in order to compare infrequent pattern analysis with k-median clustering we transformed the dataset to a dataset with continuous attributes.

6.2.1 Results infrequent pattern analysis

When we apply the infrequent pattern analysis on the categorical dataset, we discovered that the best performance (Kendall's rank correlation) of the outlier detection algorithm is 0.303 with parameter $\sigma = 0.80$, 40 attribute bagging iterations and a maximum of 2 attributes per attribute-bag.

Figure 6.1 visualizes the rank correlation for various number of attribute bagging iterations, T. Where the remaining parameters are set on $\sigma = 0.05$ and $b = 2$. It shows that when the number of iterations increases, the rank correlation increases. However, 20 iterations already result in a stabilized rank correlation. Which means that using 20 attribute bagging iterations is optimal, because each additional iteration increases the runtime of the algorithm. We prefer to minimize this.

Figure 6.2 shows the rank correlation of a various maximum number of attributes in the attribute bags, b, where $\sigma = 0.05$ and $T = 40$. We can see in this figure that the maximum number of attributes in an attribute bag equals 1 for the combination of attribute bagging with infrequent pattern analysis.

Figure 6.3 visualizes the rank correlation for various sigma-values. The figure shows that a diversification of sigma does not influence the correlation. This is actually very logical because frequent patterns also receive $Score_1$ and therefore only the depth of the tree influences the outlier score for different sigma's.

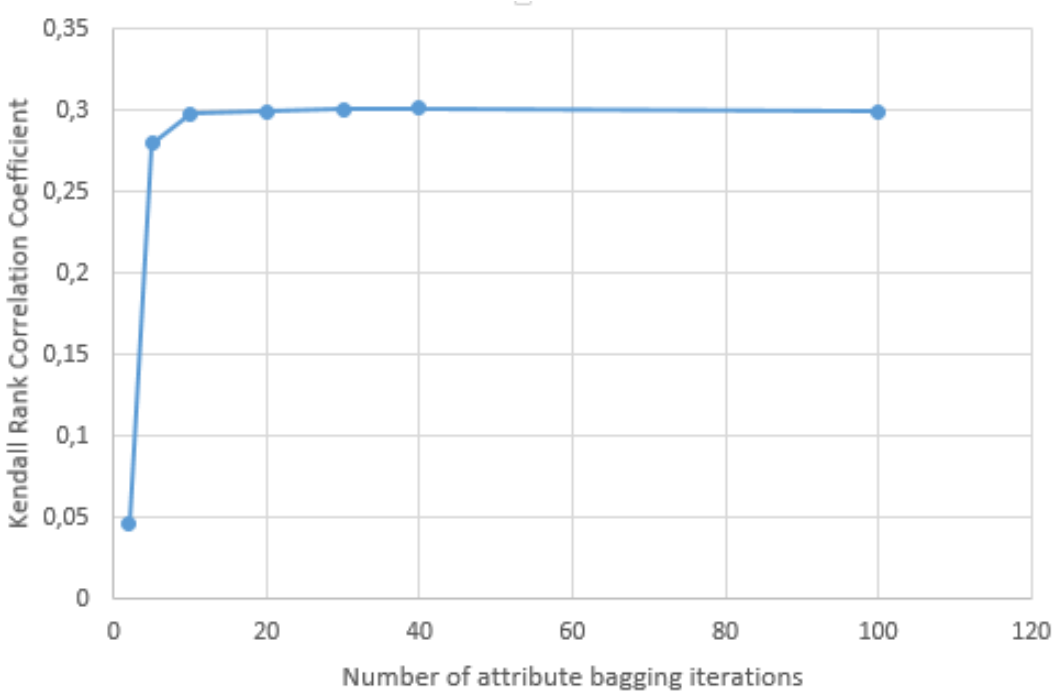

FIGURE 6.1: Rank correlation for different number of attribute bagging iterations. $\sigma = 0.05$ and maximum number of attributes in bag $= 2$

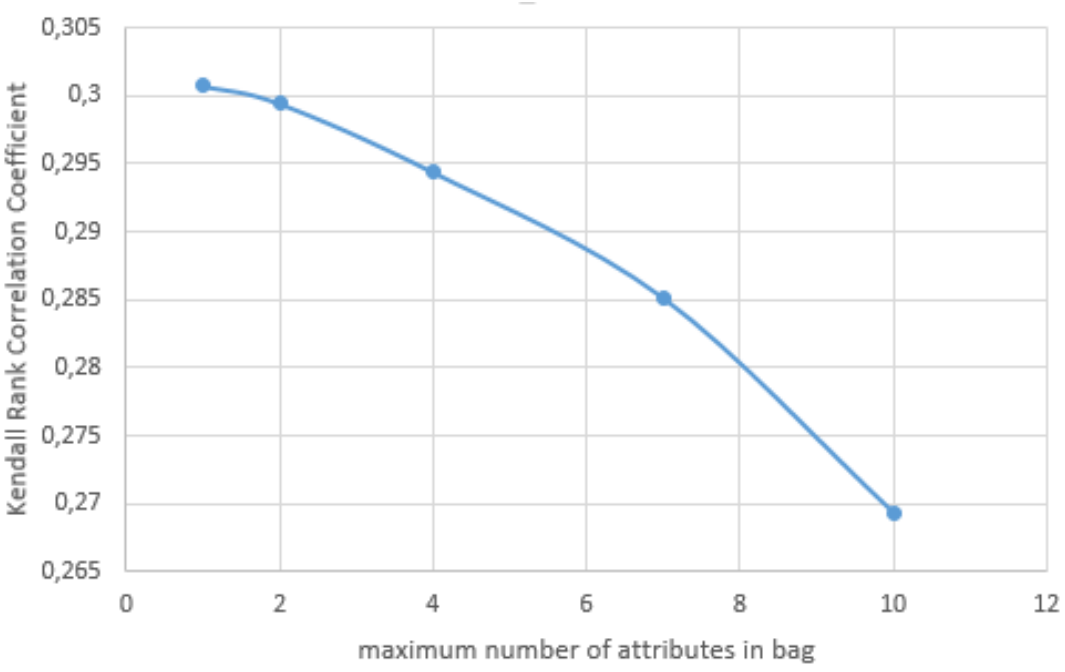

FIGURE 6.2: Rank correlation for different maximum number of attributes in the attribute bags. $\sigma = 0.05$ and number of attribute bagging iterations equal to 40

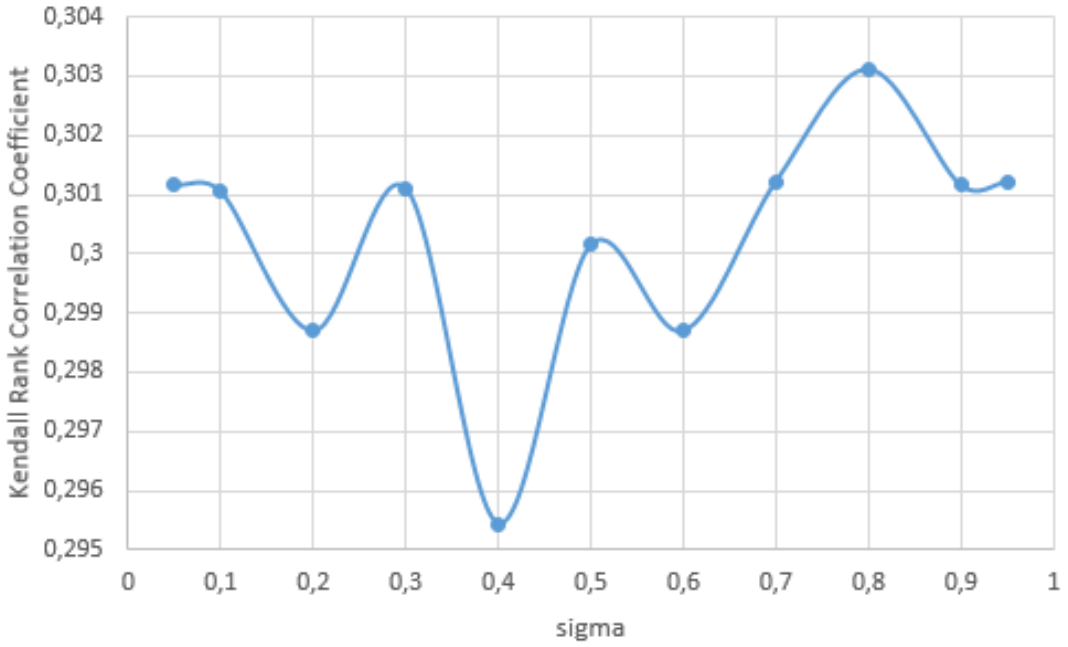

FIGURE 6.3: Rank correlation for different sigma's. With number of attribute bagging iterations equal to 40 and maximum number of attributes in an attribute bag equal to 2

6.2.2 Results k-medians clustering

To compare the previous results with the k-median clustering algorithm, we transform the categorical attributes of the Wisconsin Breast Cancer dataset to continuous attributes. After applying the algorithm on the continuous dataset, we discovered that the best performance of the algorithm is 0.304 using 2 maximum attributes in an attribute bag and 40 attribute iterations as parameters. Note that these are the only parameters we need, because k-median clustering is used instead of infrequent pattern analysis.

Figure 6.4 visualizes the rank correlation of various number of attributes in an attribute bag, b, with $T = 100$ and maximum number of clusters is equal to 8. The figure shows a peak in the correlation when the maximum number of attributes in an attribute bag equals 2. With infrequent pattern analysis we found an optimum with a maximum number of 1 attribute in the bags. So both techniques, infrequent pattern analysis and k-medians clustering, prefer a low b.

Figure 6.5 shows the rank correlation of various number of attribute bagging iterations, T, with $b = 2$ and maximum number of clusters is equal to 8. The rank correlation stabilizes after 40 iterations. Comparing this to attribute bagging in combination with

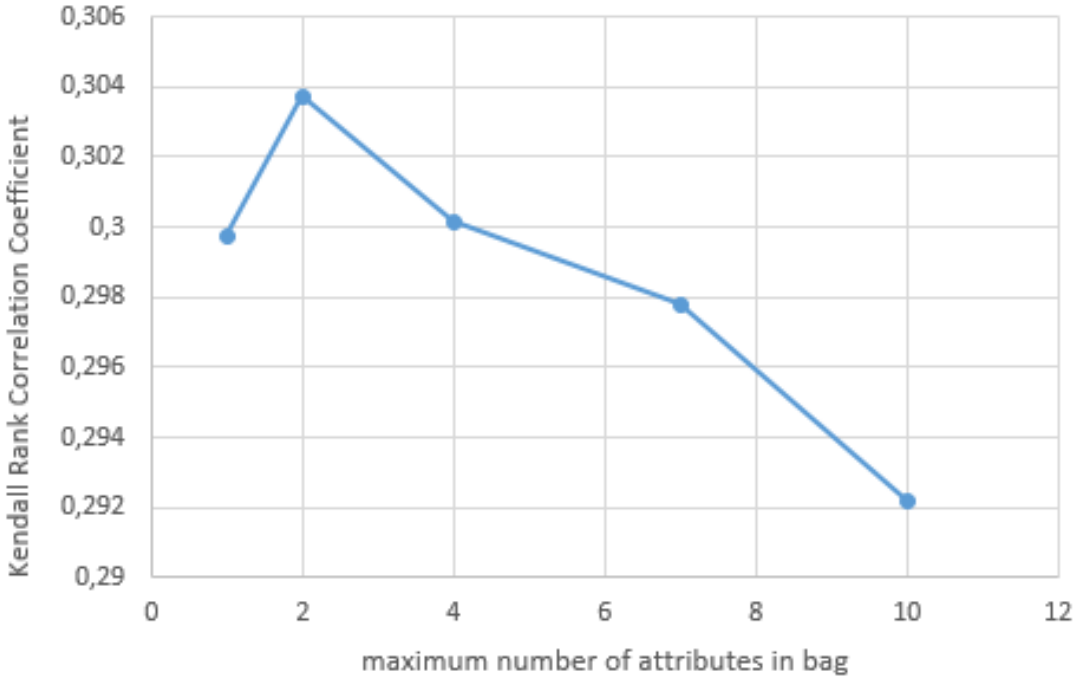

FIGURE 6.4: Rank correlation for different maximum number of attributes in an attribute bag, with 100 attribute bagging iterations and the maximum number of clusters is 8.

infrequent pattern analysis, where the rank correlation stabilizes at 20 iterations. Therefore, we can conclude that k-medians clustering needs more iterations for an optimal rank correlation.

Both continuous and categorical attribute bagging models have similar rank correlations, which concludes that in case of the Wisconsin Breast Cancer dataset both techniques find the same number of outliers. The rank correlation between the results of the maximum infrequent pattern analysis and the maximum k-medians clustering is equal to 1. This concludes that both algorithms have similar rankings, so we can conclude that it marks the same records as outliers.

6.3 Model validation

The robustness of the model is validated using the technique of 5-fold cross validation. The rankings of the outlier detection algorithm on the 5 different samples are similar, correlation is 1. This means that the outlier detection algorithm is robust.

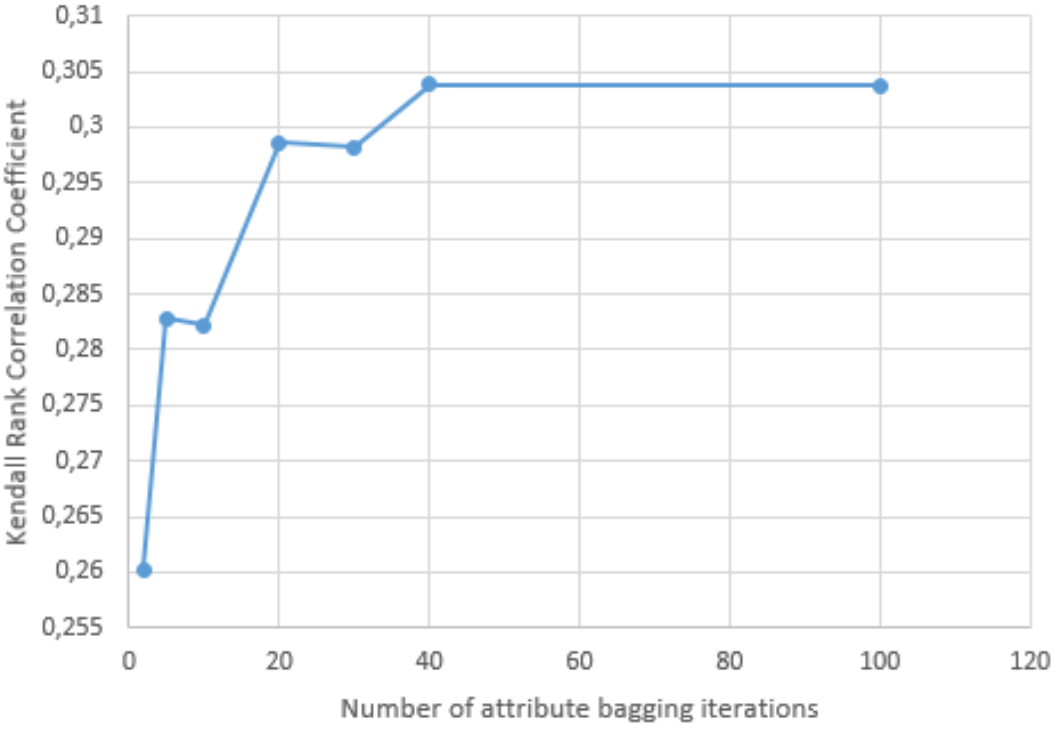

FIGURE 6.5: Rank correlation for different number of attribute bagging iterations, with 2 maximum attributes in an attribute bag and the maximum number of clusters is 8.

Chapter 7

Conclusion

The objective of this research was to investigate an outlier technique that can be used for datasets with mixed-attributes. The reason of creating an outlier detection model is to detect fraudulent behavior in large datasets, assuming that fraudulent behavior is outlying (e.g. credit card fraud or intrusion in a network). Due to the unlabeled data in most investigations, we studied unsupervised learning techniques.

Nowadays, almost every real dataset contains both categorical and continuous attributes, which is called mixed-attribute datasets. Most researchers that investigate outlier detection merely use datasets containing categorical or continuous attributes. Only few detected outliers in mixed-attribute datasets. This research found a technique that can detect outliers in mixed-attribute datasets. First, we split the problem into two separate problems: outlier detection in a dataset with continuous attributes and outlier detection in a dataset with categorical attributes.

An appropriate and well investigated technique for datasets with continuous attributes is k-median clustering. This technique is officially not used for outlier detection, but is able to detect outliers by clustering similar records based on their continuous behavior. A record with an increased distance to the closest cluster has a higher probability of being an outlier.

Infrequent pattern analysis is an outlier detection technique that can be used for datasets with categorical attributes. Infrequent pattern analysis investigates the occurrence of patterns. If a pattern is below a user-specified threshold, the pattern is considered infrequent. This research proposes a new technique of infrequent pattern analysis, Postponed Outlier Detection.

In recent years, the amount of data grows exponentially. Therefore, many datasets contain many (redundant) attributes, which is called high-dimensional data. When datasets

are high-dimensional they suffer from the "curse of dimensionality" where all datapoints become equidistant and similar. This makes it more complex to detect outliers. This research investigated attribute bagging, which is a technique that can deal with high-dimensional data.

The proposed outlier detection technique for detection of outliers in mixed attributes uses a combination of continuous and categorical outlier detection. A variant of infrequent pattern analysis, called postponed outlier detection, is used for detecting outlier in datasets with categorical attributes. Postponed outlier detection calculates a score for all records, which indicates the probability of a record being an outlier. The proposed technique uses k-Median clustering technique to detect outliers in a dataset with continuous attributes. The Manhattan distance from a point to its centroid defines the outlier score for this, continuous, subset. The outlier scores from both categorical and continuous data algorithm results in a ranking with the most feasible outlier top-ranked. The proposed technique uses attribute bagging as a possible attribute selection method.

Three different components drive the outlier detection algorithm: infrequent pattern analysis; k-median clustering; and attribute bagging. These different algorithms are validated separately and combined on two different datasets, NSL_KDD and Wisconsin Breast Cancer dataset. The separated components and different combinations of components are validated applying the different algorithms on the NSL_KDD dataset. Infrequent pattern analysis and k-median clustering is compared applying both algorithms on the Wisconsin Breast Cancer dataset.

Based on the results, we conclude that for the NSL_KDD dataset the infrequent pattern analysis is more successful than the k-median clustering. However, the correlation is very low between the true-outliers and the outliers detected by the algorithm. So the algorithm is not accurately detecting the outliers in the NSL_KDD dataset.

The k-medians clustering detected the same outliers as the infrequent pattern analysis in the Wisconsin Breast Cancer dataset. However, the correlation is low between the detected outliers and true-outliers.

The proposed outlier detection technique is not able to detect all outliers for the NSL_KDD and Wisconsin Breast Cancer dataset. Therefore, to validate whether the proposed outlier detection technique is accurate in detecting outliers more research is necessary.

This research investigated an outlier detection technique that can detect outliers in datasets, without knowledge about the attributes. Attribute bagging makes it possible to randomly select attributes by the data scientist. Without this possibility selection attributes would be very time consuming.

The limitation of this research is that when the data is widely spread it is harder to detect outliers, because of non-similarities. A possible way to avoid this problem is to add additional explanatory attributes to the dataset.

It is difficult to validate the outlier detection algorithm on a dataset with supposed fraudulent behavior. Detecting fraudulent behavior in datasets is the official background of this thesis. Thorough investigation is necessary to validate the algorithm for these datasets. These investigations require special qualified teams.

Further research could focus on validating the dataset for different datasets, preferable datasets with supposed fraudulent behavior. Also, researchers could investigate a different approach to set up the attribute tree for infrequent patterns. Perhaps this changes the scores for categorical datasets.

Another possible future research suggestion is a different technique of selecting attributes. Perhaps if attributes are not randomly selected, the outlier scores are more accurate.

This research investigated a complex problem with a complex solution. We used an alternative approach, which needs to be further investigated. This research adds value in the area of unsupervised outlier detection techniques. This field is not fully discovered yet.

Appendix A

Attacks in KDD cup 1999 dataset

Type of attacks	
spy	2
sqlattack	2
udpstorm	2
worm	2
xsnoop	4
perl	5
phf	6
xlock	9
ftp_write	11
loadmodule	11
imap	12
xterm	13
sendmail	14
ps	15
named	17
land	22
rootkit	23
multihop	25
$\text{buffer}_o\text{verflow}$	50
httptunnel	133
snmpgetattack	178
pod	221
mailbomb	293
saint	319
snmpguess	331
processtable	685
apache2	737
warezclient	890
teardrop	901
warezmaster	964
mscan	996
$\text{guess}_p\text{asswd}$	1284
back	1300
nmap	1566
portsweep	3070
smurf	3108
ipsweep	3643
satan	4360
neptune	45716
normal	76967

Bibliography

[1] Association Of Certified Fraud Examiners. Report to the nation on occupational fraud and abuse.

[2] Z. Rezaee. *Financial statement fraud: prevention and detection*. 2002.

[3] B. A. Garner. *Black's Law Dictionary*. 8th edition edition.

[4] I. Ben-Gal. Outlier detection. 2005.

[5] M. Markou and S. Singh. Novelty detection: a review –part 1: statistical approaches. *Signal Processsing*, Volume 83(Issue 12):2481 – 24979, December 2003.

[6] R. Pamula, J. Kumar, S. Nandi. An outlier detection method based on clustering, 2011.

[7] Sanjay Chawla, Aristides Gionis. k-means: A unified approach to clustering and outlier detection, 2013.

[8] R.J. Bolton and D.J. Hand. Unsupervised profiling methods for fraud detection, .

[9] K. Yamanishi, J. Takeuchi,G. Williams. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. 2000.

[10] R.J. Bolton and D.J. Hand. Statistical fraud detection: a review, .

[11] A. Lazarevic, V. Kumar. Feature bagging for outlier detection.

[12] M.I. Petrovskiy. Outlier detection algorithms in data mining systems. *Programming and Computer Software*, Volume 29(Issue 4):228 – 237, February 2003.

[13] J. Zhang and H. Wang. Detecting outlying subspaces for high-dimensinal data: the new task, algorithms and performance.

[14] H.P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data.

[15] Z. Ferdousi and A. Maeda. Anomaly detection using unsupervised profiling method in time series data. 2006.

[16] Lior Rokach and oded Maimon. *Clustering Methods*.

[17] Benjamin J. Anderson, Deborah S. Gross, David R. Musicant, Anna M. Ritz, Thomas G. Smith, Leah E. Steinberg. Adapting k-medians to generate normalized cluster centers, 2006.

[18] A. Hasnat, S. Halders, A. Hoque, D. Bhattacharjee, M. Nasipuri. A fast epga based on architecture for measuring the distance between two color images using manhattan distance metric. *International journal of electronics and communication engineering and technology*, Volume 4(Issue 3):pp. 01–10, May - June 2013.

[19] D. Pelleg, A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters.

[20] D.T. Pham, S.S. Dimov, and C.D. Nguyen. Selection of k in k-means clustering, 2004.

[21] S. Ray and R.H. Turi. Determination of number of clusters in k-means clustering and application in color image segmentation.

[22] Tomas Borovicka, Marcel Jirina, Pavel Kordik, and Marcel Jirina. *Selecting Representative Data Sets*.

[23] J. Gao and P. Tan. Converting output scores from outlier detection algorithms into probability estimates, 2006.

[24] C. Aggarwal. *Outlier Analysis*.

[25] Matthew Eric Otey, Amol Ghoting and Srinivasan Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets, June 2005. URL http://web.cse.ohio-state.edu/dmrl/papers/TR42.pdf.

[26] Anna Koufakou. Outlier detection for large distributed mixed-attribute data: Odmad. In *scalable and efficient outlier detection in large distributed data sets with mixed-type attributes*, 2009.

[27] C. Borgelt. Frequent pattern mining.

[28] Kdd cup 1999. URL http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[29] M. Lichman. {UCI} machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

[30] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. 2009.

[31] M.G. Kendall. *Rank correlation methods.* 4th edition edition.

[32] E. Szmidt, J. Kacprzyk. The spearman and kendall rank correlation coefficients between intuistionistic fuzzy sets, July 2011.

[33] W.H. Wolberg. Breast cancer (original) wisconsin.

[34] J. Lengt. A novel subspace outlier detection approach in high dimensional data sets, 2010.

[35] A. L. S. Reddy, B. R. Babu, A. Govardhan. Outlier analysis of categorical data using navf. *Informatica Economică*, Volume 17(Issue 1):5 – 13, January 2013.

[36] V. Kumar, D. Kumar, R.K. Singh. Outlier mining in medical databases: an application of data mining in health care management to detect abnormal values presented in medical databases. *International Journal of Computer Science and Network Security*, Volume 8(Issue 8):272 – 277, August 2008.

[37] L. Duan. *Density-Based Clustering and Anomaly Detection.*