Opleiding Informatica

Automated feature selection for unsupervised anomaly detection with internal evaluation strategies

Samuel Smulders

Supervisors:
-Matthijs van Leeuwen; Associate professor/Director of Education (Leiden University)
-Zhong Li; PhD candidate (Leiden University), daily supervisor.

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

1/7/2023

Abstract

The advancements in technology from the past decades and the widespread use of internet has led to a significant increase in the generation and availability of information. Consequently, large datasets are available to individuals, as well as businesses. Proper analysis of these datasets is of paramount importance when used for real-life decision making. Anomaly detection is one of these analysis tasks. Its task is to identify instances that deviate from an established pattern. In practice it is used for fraud detection, health monitoring, cyber security, preventive maintenance and many other areas. It is therefore indispensable for today's data analysis.

The curse of dimensionality that large datasets have brought along has made unsupervised anomaly detection increasingly difficult. It is essential that feature selection is applied to reduce this high-dimensionality, in order to build comprehensible predictive models with good performance. However, it has come to our attention that automatic feature selection is an understudied part in unsupervised anomaly detection, as most hyperparameters tuning, such as the setting of the percentage of selected features, is done by humans with domain knowledge.

Hence, the aim of this research is to take the most important hyperparameter - the percentage of selected features - as an example to explore automatic feature selection for unsupervised anomaly detection tasks. Therefore, we create a pipeline in which automatic feature selection for unsupervised anomaly detection is performed. Given an input dataset, this automated ML model selects the best subset of features for an unsupervised anomaly detection task.

Research on 5 different datasets with 10 different hyperparameter configurations based on 2 different feature selection algorithms and 2 different anomaly detection algorithms shows that 1 out of 6 internal evaluation strategies is the most accurate. As a result, the use of this model could be a first steps in automatic feature selection for unsupervised anomaly detection.

Contents

1 Introduction

The availability of large amounts of data has given individuals, organisations and businesses numerous possibilities. An example would be the customer-centric approach, which makes use of *associate rule mining*. In this approach the business uses data to drive its actions. An organisation can personalise messaging to its prospects and customers to create a more effective way of approaching them. Or a simple *classification* problem, such as predicting whether an email is spam and should be delivered to the junk folder, based on previously acquired data. Further applications are fraud detection, health monitoring, cyber security and preventive maintenance, where *anomaly detection* is essential. All these analysis tasks are based on large amounts of data a specific entity has access to.

Anomaly detection is one of the tasks that has become increasingly important, considering the fact that anomalies in data often translate to critical and actionable information in a wide variety of application domains such as fraud detection, health monitoring, cyber security and preventive maintenance [1]. In preventive maintenance for example, more accurate anomaly detection ensures lower maintenance costs and high system availability [2]. For that reason anomaly detection will be further studied in this thesis.

Anomaly detection refers to 'finding patterns in data that do not conform to expected behaviour' [3]. The datasets on which anomaly detection is performed are becoming increasingly large, due to increased storage of information. When looking at the previously mentioned preventive maintenance, we can think of these features as a machine's downtime, runtime, power usage, temperature, and so on. The number of these features is referred to as the dimensionality of the dataset. An issue that comes along with the increase of the dimensionality, is that many traditional anomaly detection methods become ineffective as they fail to retain sufficient accuracy. This phenomenon is referred to as the *Curse of dimensionality* [4]. Therefore using dimensionality reduction methods including feature selection on the dataset before applying anomaly detection algorithms would tackle the curse of dimensionality and thus improve efficiency and effectiveness of the anomaly detectors [4]. That being the case shows the importance of choosing the right set of features for anomaly detection.

In recent work the effects of feature selection on anomaly detection algorithms have been analyzed [5]. This study has systematically investigated the impacts of feature selection on different anomaly detection algorithms. The results show that, in general, it is beneficial to leverage feature selection in terms of efficiency, while there is no significant difference in terms of accuracy. However, in this prior work the number of features selected has been kept constant. In other works hyperparameter tuning is done by humans with domain knowledge, which limits automation and efficiency in research related to feature selection. Which is a problem that we aim to tackle. This gave rise to the following question considering feature selection:

- Is there any effective automatic strategy to specify the number of selected features?

Furthermore, reference [6] has performed a systematic review of internal model evaluation strategies for *unsupervised anomaly detection*, including stand-alone strategies and consensus-based strategies. In that paper, they attempt to study the following problem: Given an unsupervised anomaly detection task and a pool of anomaly detection algorithms, how to select an anomaly detection algorithm and its associated hyperparameters?

The questions that arose from the research in reference [5] and the internal evaluation strategies for unsupervised outlier model selection from research [6] has led to conducting the following thesis research: Taking the most important hyperparameter- the number of selected features- as an example, how to create a machine learning pipeline that can automatically select the best hyperparameter values of a feature selection method for unsupervised anomaly detection.

This thesis touches upon a, to our knowledge, not yet studied part for automatic hyperparameter tuning for unsupervised anomaly detection tasks. Due to this reason, we only use one hyperparameter, *the percentage of features selected*. This study can easily be extended to a selection of different types of hyperparameters in feature selection algorithms.

The contributions of this thesis are twofold. First, to explore the feasibility of performing automatic feature selection for unsupervised anomaly detection, we have carefully devised a pipeline and conducted experiments, which we briefly descibe as follows.

For the automatic feature selection, we have created a pipeline that starts with a complete high-dimensional *unlabelled* dataset. This dataset is then processed into 10 different lower-dimensional sub-datasets. Lowering the dimensions is done by a feature selection algorithm. Each one of the lower-dimensional datasets corresponds to a hyperparameter configuration for a feature selection algorithm. Therefore, a hyperparameter(HP) configuration 0.X (HP-configuration 0.X) means: the sub-dataset has 0.X dimensions of the total number of dimensions from the complete dataset. These different HP-configuration are then processed by an anomaly detection algorithm, leading to different sets of anomaly scores for every HP-configuration. These anomaly scores are then analysed by *internal evaluation strategies*. The internal evaluation strategies are necessary as this thesis focuses on unsupervised outlier detection, meaning that there are no true labels for outliers in our model. Each internal evaluation strategy automatically ranks the HP-configurations' performances based on their own metric. These rankings show the best HP-configuration for a dataset, given a feature selection algorithm and anomaly detection algorithm based on different internal evaluation strategies. Further explanation of the pipeline and the different variables will be given in Section 3.

The initial findings and conclusions of this new pipeline are considered the second contribution of this thesis, where by means of Spearman's correlation coefficient we check whether the automatically chosen set of features is in line with the best set of features based on the ground truth. This part will be discussed in Section 4. Tests on five different datasets using two different feature selection algorithms and two different anomaly detection algorithms show that the UDR method (Section 3.5.1) has done a significantly better job in automatic feature selection for anomaly detection, given a unsupervised outlier detection task. This will be further discussed in section 5.

2 Preliminaries and related work

2.1 Definitions

Some technical terms in this thesis that are used interchangeable. These terms are explained in this section to facilitate better understanding and add clarity.

• **Feature** is referred to as a column in a tabular data-set.
• **Feature selection** is the process of selecting a subset of features in a data-set. In this thesis interchangeable with *Hyperparameter configuration (HPconfiguration)* [7].
• **Anomaly** is a datapoint that deviates significantly from the expected/typical pattern of a dataset. The datapoint is different from other datapoints in terms of value. In this thesis interchangeable with the term *outlier* [3].
• **Outlier score / Anomaly score** is a numerical value that quantifies to what degree a datapoint can be considered as an oulier in a dataset. A high outlier score means a greater deviation. Suggesting a higher likelihood of being an outlier. Conversely, lower outlier score would suggest a more typical datapoint. In this thesis outlier scores are calculated using anomaly detection algorithms [8].
• **Anomaly detection** is the process of identifying abnormal observations in data that deviate from the norm [3].
• **Internal evaluation strategies**
In this thesis anomaly detection is performed on unlabelled datasets. In other words, unsupervised anomaly detection. This means that for our automated model we do not have true labels that say if an instance is an outlier. Therefore we use internal evaluation strategies to determine the outliers. *UDR, HITS and Cluster validation* (Section 3.5) are the three different internal evaluation strategies used in this thesis [6].

2.2 Related Work

Research on the effect of feature selection on unsupervised anomaly detection [5] gave rise to initial questions on automated feature selection. A taxonomy of unsupervised feature selection methods and a description of the main ideas behind them has been provided by reference[7]. Moreover, reference [9] has elaborated on the feature selection algorithms used in the experiments, namely *Laplacian score* and *Spectral feature selection*. Further explanation will be done in Section 3.3.

A broad explanation of anomaly detection has been given by reference [3]. This reference provided us with the necessary information for the anomaly detection algorithms *One-class support vector machine* and *K-nearest neighbour*. These algorithms will be covered in Section 4. Adding to this, reference [10] has provided us with the metrics for the evaluations of unsupervised outlier detection, namely using ROC AUC to use as ground truth. Another essential element for anomaly detection has been the development of the package PyOD by [11], which provided us with a set of 25 commonly used anomaly detection algorithms.

In addition, the review and evaluation of internal evaluation strategies [6] provided us with different internal evaluation strategies such as UDR, HITS, and Cluster Validation metrics. For the computation of the Cluster validation we used four different types that have been used in reference [12]. These are the Davies-Bouldin index, Xie-Beni index, Calinski-Harabasz index and Silhouette score.

3 Design of Pipeline and Experiments

3.1 Pipeline

In order to construct an automated machine learning method the code adheres a pipeline structure in which different points determine key processes with the data. At these points different variables have been chosen. Point one, starting point, is the data-set. Followed by point two where a feature selection algorithm is used to create a new, lower-dimensional, data-set. In this research we set 10 different values for the remaining number of features for the original data-set. These are depicted by HP-configuration. After this step comes, point 3, the anomaly detector. The anomaly detector results in a list with different outlier scores for each HP-configuration. At the final point, point 4, an internal evaluation strategy is used to rank the HP-configurations from best (1) to worse (10). Figure 1 gives a simple visualisation of this pipeline.

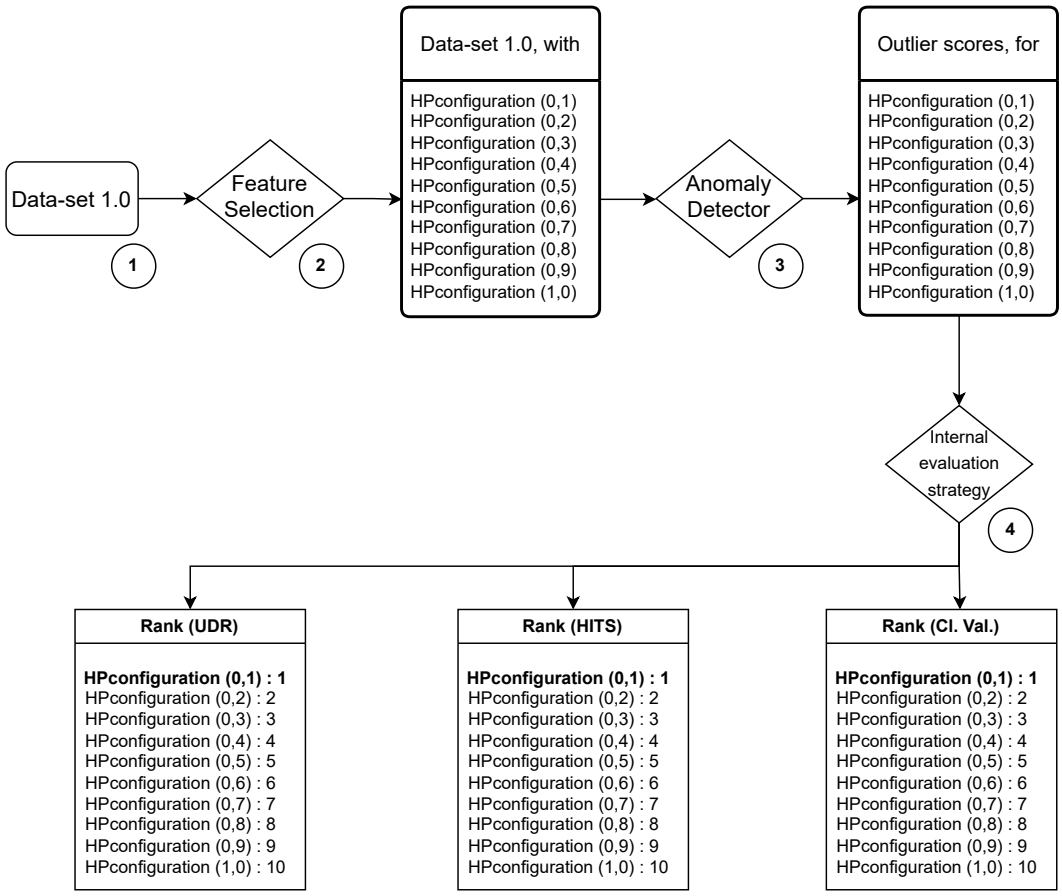

Figure 1: This figure shows the pipeline for automatic feature selection for unsupervised anomaly detection. The pipeline starts at the top left side. *Data-set 1.0* is split into 10 smaller datasets/HPconfigurations. The remaining features are selected by a feature selection algorithm. For every HP-configuration an anomaly detection algorithm determines the outlier scores. Based on these outlier score the three different internal evaluation strategies (UDR, HITS and Cl.Val.) produce a ranking, showing the best HPconfiguration for *Data-set 1.0*.

3.2 Data sets

In the field of machine learning, datasets can be categorized into two main types: labeled and unlabeled. On the one hand we have labeled dataset consisting of data points or examples that have been explicitly assigned corresponding labels or target values. These labels are typically provided by crowd-sourced content such as the 'ImageNet database' or human annotators who possess domain knowledge and expertise. Labeled datasets are essential for supervised learning algorithms, where the goal is to train a model to make predictions based on the given inputs and their associated labels [13].

On the other hand, we have unlabeled datasets which lack explicit labels or target values. They contain raw data points without any predefined class or category information. Unlabeled datasets are commonly used in unsupervised learning tasks, where the objective is to discover patterns, structures, or relationships within the data without prior knowledge. Unsupervised model selection is therefore complicated by the fact that there is labeled data. This has made hyperparameter selection an understudied and difficult topic in unsupervised outlier detection.

These unlabeled datasets have grown to be incredibly large, containing vast amounts of information. However, this exponential increase in dataset size has led to a phenomenon known as the Curse of dimensionality. Dimensionality refers to the number of features or attributes used to describe each data point. The available data points become more isolated and spread out in the feature space. This sparsity and scattering make it challenging to apply traditional techniques or analysis methods that rely on close proximity or density of data points. Feature selection then becomes increasingly important when using unsupervised outlier detection algorithms [14].

For this research we use five different datasets from, the in total, twenty benchmark datasets that are available in *GitLab - LIACS* [15]. These five have been chosen for their relatively small number of datapoints. This property has made runtime smaller, which provided us with the opportunity to focus on the correctness of the pipeline. The five datasets will go through the complete pipeline. Further details of the datasets are described in Table 1.

Table 1: Benchmark datasets used to run through the pipeline

Dataset	**Domain**	**Data size**	**Dimensionality**	**Outliers**
1.WDBC	Healthcare	367	30	10
2.Arrhythmia	Healthcare	452	274	66
3.Ionosphere	Science	351	32	126
4.Letter Recognition	Images	1600	32	100
5.KDDcup99, reduced	Computer Traffic	4811	40	54

3.3 Feature selection

Feature selection is an essential step in unsupervised machine learning when working with unlabeled datasets. It involves identifying the most relevant features that contribute to the underlying patterns and structures within the data. Given a feature selection method we can alter the hyperparameters, also called the HP-configuration. In this thesis we will look at 10 different HP-configurations for each feature selection algorithm. These different HP-configurations are the following: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0.

The HP-configuration of 0.X means that the dimensionality of the original data-set is reduced to 0.X of the total number of features. Therefore a data-set, WDBC for instance, which has a dimensionality of 30 will be reduced to a dimensionality of 15 at HP-configuration (0.5). The set of features depends on the feature selection algorithm being used. For the preliminary results of this thesis we will stick to the Laplacian score and Spectral feature selection, as explained in the next subsections. The limited selection of feature selection algorithms is due to computation time constraints. However, this section can easily be extended by using more feature selection algorithms, present in reference [5]. The created pipeline is build in such a way that these variables can be changed manually.

3.3.1 Laplacian score

Laplacian Score [16] is a method used to assess the significance of features based on their ability to preserve the local relationships within a dataset. It achieves this by considering each data point as a node and creating a graph based on the nearest neighbors. If a node is one of the k nearest neighbors to another node, an edge is formed between them. Using this information, a Laplacian Matrix is constructed, assigning higher weights to features that effectively maintain the structure of the graph. The Laplacian Matrix is examined to identify the features that contribute most to preserving the underlying graph's characteristics [16].

Computation of the laplacian score as feature selector is done through the *lapscore* function that is present in the scikit library [17]. This results in 10 different configurations of the original dataset.

3.3.2 Spectral feature Selection

Spectral Feature Selection (SPEC) [18] utilises pairwise similarity between data points to select features. Specifically, SPEC first constructs a graph to represent the pairwise similarities, applying graph theory to similarity graph. Then, if a feature is consistent with the graph structure, this feature is able to separate the data points better than a feature that is inconsistent with the graph structure. Therefore, this feature is considered relevant and assigned a high weight [18].

3.4 Anomaly detection

Unsupervised anomaly detection is a valuable technique used in various fields to identify unusual patterns or outliers within a dataset without relying on labeled examples. This approach aims to discover deviations from the expected norm or behaviors that do not conform to the established patterns, making it particularly useful. The process typically involves extracting meaningful features from the data and employing statistical methods, clustering algorithms, or density estimation techniques to detect abnormalities. By learning the inherent structure and distribution of the dataset, unsupervised anomaly detection algorithms can effectively identify anomalous instances.

In anomaly detection, the anomaly scores are calculated by an anomaly detection algorithm. The process involves calculating a score for each data point, reflecting its degree of abnormality compared to the rest of the dataset. These outlier scores are derived from various machine learning techniques, such as distance-based measures, density estimation, or model-based approaches [3]. The idea is to assign higher scores to instances that deviate more from the expected patterns. By setting a threshold on the outlier scores, data points surpassing this threshold are classified as outliers. This approach allows for flexible detection of anomalies without relying on predefined labels, making it suitable for unsupervised outlier detection tasks. Therefore, the outlier scores provide a quantitative measure of abnormality, enabling us to 'pick the odd one out'.

So, for every HP-configuration we get an outlier score list. The values in these outlier score list are different for all HP-configurations as they are based on different configurations of the original data-set. Therefore we get a set of ten different outlier score lists, based on one anomaly detection algorithm. In this thesis the preliminary results are based on OCSVM and KNN as the anomaly detection algorithms. The limited selection of anomaly detection algorithms is due to computation time constraints. However, this section can easily be extended by using more anomaly detection algorithms, present in reference [5]. The created pipeline has been build in such a way that these variables can be changed manually.

3.4.1 One-Class Support Vector Machine

The One-Class Support Vector Machine (OCSVM) [19] is a classification method that specifically deals with scenarios where there is only one desired class. It aims to identify anomalies by considering data objects that do not belong to the target class. OCSVM achieves this by creating a hyperplane with the largest possible margin (d in Figure 2) to separate the target class from the rest of the data. The hyperplane is adjusted to align with the distribution of the data, pulling it closer to the data points [20]. The OCSVM algorithm has been used to compute the outliers scores for the ten different HPconfigurations. This provides us with a set of ten different outlier score lists. In the resulting lists with outlier scores, high values for a data-point indicate more probable outliers.

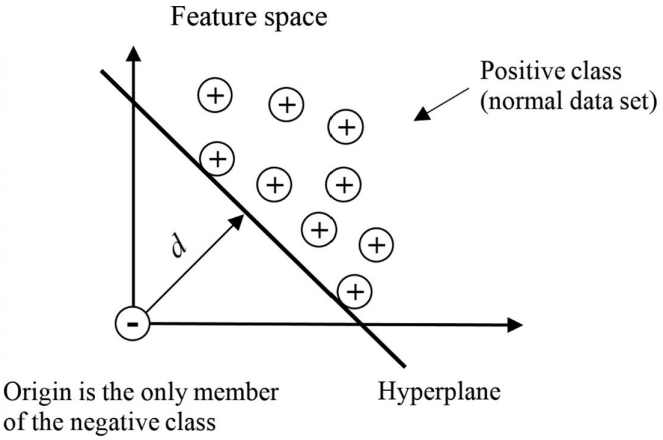

Figure 2: Shows the basic principle of the OCSVM algorithm used. In this approach, the training points are considered as members of the positive class or *inlier class*, while the origin is treated as the negative or *outlier class*.[19].

3.4.2 K-Nearest Neighbour

K-Nearest Neighbors (KNN) computes similarity with the neighborhood data points and considers the proximity of data points to identify anomalies. By calculating the distances between a *query point* and its K nearest neighbors, the algorithm determines the local density of the data point. Anomalies are often characterized by their significant deviation from the neighboring points, resulting in a larger distance to the K nearest neighbors. By setting a threshold for the distance, KNN can classify data points as either normal or anomalous. This distance-based approach leverages the concept that anomalies tend to exhibit dissimilarity in terms of distance compared to the majority of the data points, making KNN an effective method for anomaly detection [1]. For that reason KNN has been chosen besides OCSVM to run in the initial tests of the created pipeline. After using this algorithm on the ten different HP-configurations, we are left with ten different outlier score lists. High outlier scores indicate that a datapoint is more probable of being an outlier. Where *Algorithm 1* is provided by reference [1].

Algorithm 1 KNN Algorithm

Input: the training set D, test object x, category label set C

Output: the category c_x of test object x, c_x belongs to set C

0: **'Start'procedure** KNN(D, x, C)

1: **for** each $y \in D$ **do**

2: Calculate the distance $D(y, x)$ between y and x.

3: **end for**

4: Select the subset N from the dataset D, where N contains k training samples which are the k nearest neighbors of the test sample x .

5: Calculate the category of x:

$$c_x = \arg\max_{c \in C} \sum_{y \in N} I(c_{\text{class}}(y) = c)$$

6: **'End'procedure**

3.5 Internal evaluation strategies

In this study, internal evaluation strategies are employed to automatically selected the best HP-configuration without accessing the ground-truth labels. The selection process utilizes internal information, which is limited to two key elements: the internal evaluation strategies along with the outlier scores per HP-configuration [6]. The underlying principle shared by all internal evaluation strategies in this research is the utilization of estimated heuristic to asses their performance. This means that, among the various models using the internal measure, the model with the highest value with respect to their chosen measure is the best pest performing automated feature selection model. This research focuses on 3 different types of internal evaluation strategies. These are UDR, HITS and Cluster validation metrics. They have been selected to cover the each different *type* and *strategy* for internal evaluation [6]. Their properties are described in Table 2.

Table 2: Overview of internal evaluation strategies used in the experiments.

Method	**Type**	**Based on**	**Strategy**
UDR	Consensus	Outlier scores	One-shot
HITS	Consensus	Outlier scores	Iterative
Cluster Validation Metrics	Stand-alone	Outlier scores	Cluster quality

3.5.1 UDR

The Unsupervised Disentanglement Rank (UDR) is adopted from deep learning and refers to a process used to automatically choose or identify the most suitable model in the context of unsupervised learning [6]. Hence the use of this concept for automatic feature selection. In this internal evaluation strategy Tolstoy's theorem is applied. To cite Tolstoy, "All happy families are alike; each unhappy family is unhappy in its own way." Therefore a model that is alike most others will be considered a good model as its outlier scores are more similar. The idea of leveraging an agreement between the different HP-configurations used in this research is referred to as a consensus-based type of internal evaluation strategy.

In this context we use the Kendall Tau coefficient as similarity metric for the outlier score lists per HP-configuration. Therefore we construct a Kendall Tau matix for the outlier scores for at each HP-configuration. Thus HP-configuration(0.1) is compared to HP-configuration(0.2), (0.3), up until (1.0). An example of such a matrix can be seen in Table 3. Such a table is constructed for every combination of datasets, feature selection algorithm and anomaly detector. The Kendall-Tau function used implements the Kendall tau-b as default. The mathematical notation for the computation of Kendal tau-b.

$$t_b = \frac{P - Q}{\sqrt{(P + Q + X_0)(P + Q + Y_0)}}$$

where P is the number of concordant pairs, the points are concordant if they are in the same order with respect to each variable. Q is the number of discordant pairs, the points are discordant if values are arranged in opposite directions. X_0 is the number of pairs tied only on the X variable and Y_0 is the number of pairs tied only on the Y variable [21].

Table 3: Kendall-Tau coefficients on WDBC dataset for different HPconfiguration with Laplacian score, using OCSVM.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	1	0.825	0.723	0.623	0.578	0.533	0.504	0.477	0.458	0.410
0.2	0.825	1	0.815	0.710	0.653	0.601	0.573	0.551	0.528	0.488
0.3	0.723	0.815	1	0.853	0.776	0.713	0.683	0.658	0.634	0.589
0.4	0.623	0.710	0.853	1	0.907	0.834	0.799	0.756	0.731	0.677
0.5	0.578	0.653	0.776	0.907	1	0.915	0.876	0.827	0.794	0.730
0.6	0.533	0.601	0.713	0.834	0.915	1	0.928	0.870	0.825	0.750
0.7	0.504	0.573	0.683	0.799	0.876	0.928	1	0.927	0.877	0.798
0.8	0.477	0.551	0.658	0.756	0.827	0.870	0.927	1	0.936	0.856
0.9	0.458	0.528	0.634	0.731	0.794	0.825	0.877	0.936	1	0.906
1	0.410	0.488	0.589	0.677	0.730	0.750	0.798	0.856	0.906	1

A sum of the scores in in every row of this table, *Sum of each row*, is the metric used to show similarity. See Table 4. Therefore a higher sum value means better model. We compare the output ranking based on Kendall-Tau similarity with the ranking based on ROC AUC (ground truth) in Section 4. This has been done for the combinations of every dataset, feature selection algorithm and anomaly detection algorithm that were mentioned in Sections 3.2, 3.3 and 3.4

Table 4: HP-configuration ranking for UDR, based on the results of Table 2.

HPconfiguration	**Sum of Each Row**	**Rank**
0.1	6.131	10
0.2	6.746	9
0.3	7.445	7
0.4	7.892	4
0.5	8.056	1
0.6	7.969	2
0.7	7.966	3
0.8	7.859	5
0.9	7.689	6
1.0	7.203	8

3.5.2 HITS

We can build on the idea of *ModelCentrality* through computing centrality in a network setting. Centrality in a network works recursively, meaning that a node has higher centrality if it points to nodes that are pointed to by other nodes with high centrality [6].

To adopt this idea to our benfit we can create a bipartite network. The basis for this network are the outlier scores per HP-configuration. For every outlier score list we select *the top n values* as data points, where *top n* refers to the n highest outlier values. For the

sake of feasibility we set the threshold value as the value of the *n-th outlier score*, where n is the number of outlier that a data set has. We assume the number of outliers is known, for simplicity.

To illustrate this with an simple example; a dataset with 100 datapoints has 10 outliers. Therefore we select the 10th value in the sorted list of outlier scores as threshold value for that specific HP-configuration.

After iteration over all HP-configuration using the process described above we can construct the bipartite network. With on the left side the different HP-configurations and on the right side the datapoints with outlier scores above a threshold value. For each HP-configuration, an edge is constructed if this datapoint is above the threshold value (thus considered an outlier). The centrality score for a HP-configuration is the edge count for all the corresponding datapoints in the bipartite graph.

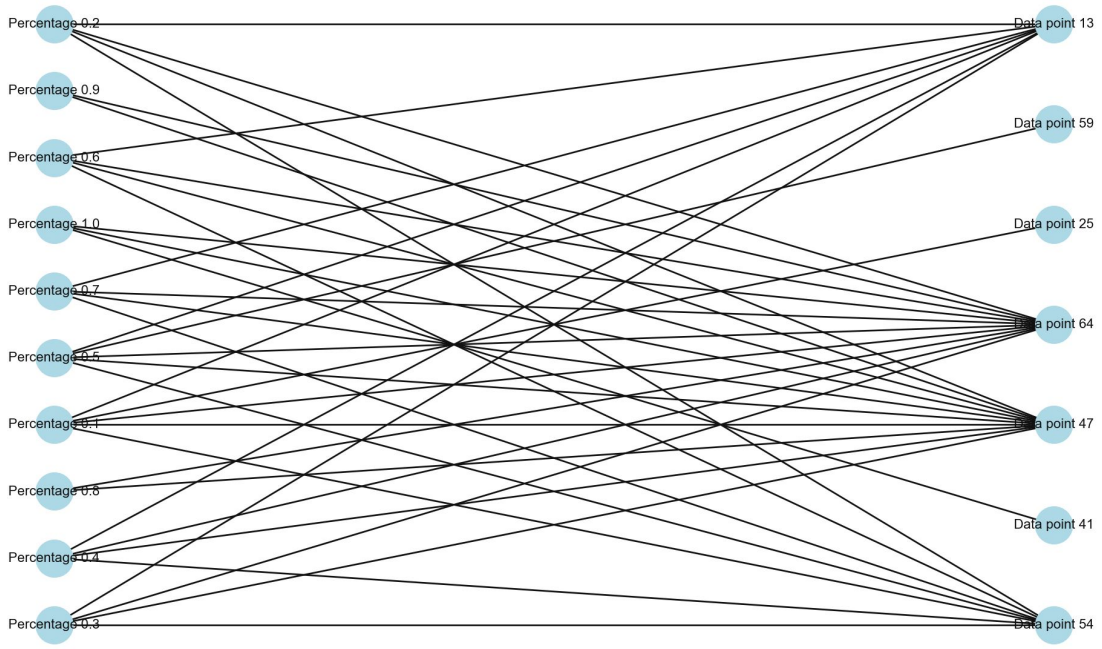

Figure 3: The bipartite network for a test dataset with HP-configuration (left) and data points (right).

Take percentage 0.2 for instance. The data points corresponding to this HP-configuration are: 13, 64, 47 and 54. Their respective degrees (number of edges connected) are 7, 10, 10 and 7. The sum of these values give the Centrality Score HP(0.2) = 34. If we apply the same reasoning on data point 0.8, we can see a Centrality Score HP(0.8) = 20. This means that based on the outlier scores for each HP-configuration the *top n* points (the data points above the threshold value/outliers) for HP(0.2) have been selected 'better' than for HP(0.8). A computation of the centrality score based on the bipartite network in Figure 3 leads to a HP-configuration ranking of the centrality score in Table 5. The bipartite graph and the resulting ranking table are computed for the combinations of every dataset, feature selection, algorithm and anomaly detection algorithm that were mentioned in Sections 3.2, 3.3 and 3.4

Table 5: HP-configuration ranking for HITS, based on the results from Figure 3.

HPconfiguration	**Centrality Score**	**Rank**
0.1	35	1
0.2	34	3
0.3	34	3
0.4	34	3
0.5	35	1
0.6	34	3
0.7	34	3
0.8	20	9
0.9	21	8
1.0	20	9

3.5.3 Cluster validation metrics

The basis for this internal evaluation strategy is the outlier score for a *specific* HP-configuration. This means that the outlier scores list for one HP-configuration is used in cluster validation. This method does not look for consensus in a pool of results, for that reason it is considered a *stand-alone type*. Since UDR and HITS are both consensus-based methods, they both use combinations of the outlier scores for all HP-configurations to get to one answer. The cluster validation metrics only looks at the outlier score per HP-configuration. Therefore this method is intuitively easier.

The intuition is that the cluster validation metric looks at the cluster quality of the chosen clusters. These clusters are based on the outlier scores produced by an anomaly detection algorithms, given a HP-configuration. The sorted outlier scores are divided into the *outlier-cluster* and the *inlier-cluster*. The number of data points in the outlier-cluster is equal to the number of data-points above the top n-th value, where n is the number anomalies in the dataset. For this threshold setting the same intuition is used as for the HITS model. The remainder of the data points correspond to the inlier-cluster.

Clustering aims to group similar objects together, while keeping objects from different clusters distinct. Internal clustering validation measures primarily focus on two criteria: compactness

and separation. Compactness evaluates the extent of similarity or dissimilarity among samples within the same cluster. Separation assesses how distinct or well-separated a cluster is from other clusters [22]. Hence, we use clustering validation metrics to evaluate the anomaly detection based on different HP-configurations.

The clustering validation metrics that will be tested for the evaluation of the goodness of anomaly detection are Davies-Bouldin index, Xie-Beni index, Calinski-Harabasz index and Silhoutte score [22]. These names will from now on be referred to as DB, XB, CH and S. For DB and XB the smaller values present better clustering. Conversely, for CH and S larger values mean that the clusters obtained are better.

Table 6: Notation and Meanings

Notation	Meaning
D	Data set
n	Number of objects in data set
c	Center of D
NC	Number of clusters
C_i	The ith Cluster
n_i	Number of objects in C_i
c_i	Center of C_i
$d(x, y)$	The distance between x and y

Davies-Boulding index (DB)
Is the average of cluster's similarities. The similarity of each cluster is defined as the maximum value of its similarities to other clusters. where DB is defined as follows.

$$DB = \frac{1}{NC}\sum_{i}\max_{j,j\neq i}\left\{\left[\frac{1}{n_i}\sum_{x\in C_i} d\left(x, c_i\right) + \frac{1}{n_j}\sum_{x\in C_j} d\left(x, c_j\right)\right] \Big/ d\left(c_i, c_j\right)\right\}$$

Xie-Beni index (XB)
Uses the minimum square distance between cluster centers as intercluster separation and the mean square distance between each data object and its cluster center as the intracluster compactness. This index is defined as the ratio of the compactness to the separation. The XB index adheres the following notation.

$$XB = \frac{\sum_{i=1}^{NC}\sum_{x\in C_i} d^2\left(x, c_i\right) \Big/ n}{\min_{i,j} d^2\left(c_i, c_j\right)}.$$

Calinski-Harabasz index (CH)
Evaluates the cluster solution based on the average between- and within-cluster sum of squares. This index is defined as follows.

$$CH = \left[\sum_{i=1}^{NC} n_i d^2\left(c_i, c\right) / \left(NC - 1\right)\right] \Big/ \left[\sum_{i=1}^{NC}\sum_{x\in C_i} d^2\left(x, c_i\right) / \left(n - NC\right)\right]$$

Silhouette (S)
Measures clustering partition based on the dissimilarity of each instance to its cluster's instances, and to its neigbour cluster's instance [22]. The silhouette index is presented using the following notation.

$$S = \frac{1}{NC}\sum_{i=1}^{NC}\left[\frac{1}{n_i}\sum\nolimits_{x\in C_i}\frac{b\left(x\right)-a\left(x\right)}{\max\left[b\left(x\right),a\left(x\right)\right]}\right],$$
$$a\left(x\right) = \frac{1}{n_i-1}\sum\nolimits_{y\in C_i, y\neq x} d\left(x,y\right),$$
$$b\left(x\right) = \min_{j,j\neq i}\left[\frac{1}{n_j}\sum\nolimits_{y\in C_j} d\left(x,y\right)\right].$$

Computation of these cluster validation metrics enables us to rank the HP-configuration based on DB, XB, CH and S. Table 7 a,b,c and d show these rankings when applied on the WDBC data-set, using Laplacian score as feature selector and OCSVM as anomaly detection algorithm. In the complete experiment this computation has been done for all combinations of datasets, feature selection algorithms and anomaly detection algorithms.

Table 7: HP Configuration rankings for cluster validation metrics, using Laplacian score as feature selector and OCSVM as anomaly detector on the WDBC dataset.

(a) HP-configuration ranking based on DB Score

HP Configuration	DB Score	Rank
0.1	1.606405248	10
0.2	0.529354753	9
0.3	0.45999627	4
0.4	0.47671772	5
0.5	0.484045825	6
0.6	0.527275923	8
0.7	0.504435676	7
0.8	0.452316164	3
0.9	0.425052191	2
1.0	0.414971229	1

(b) HP-configuration ranking based on XB index

HP Configuration	XB Index	Rank
0.1	0.011855899	10
0.2	0.004112498	9
0.3	0.003484921	1
0.4	0.003506163	2
0.5	0.003628876	3
0.6	0.00406733	8
0.7	0.003845724	7
0.8	0.003735534	4
0.9	0.003790209	6
1.0	0.003789395	5

(c) HP-configuration ranking based on CH Score

HP Configuration	CH Score	Rank
0.1	670.1478863	10
0.2	1931.965826	9
0.3	2279.881137	1
0.4	2266.068663	2
0.5	2189.439548	3
0.6	1953.420189	8
0.7	2065.984211	7
0.8	2126.926498	4
0.9	2096.244801	6
1.0	2096.694989	5

(d) HP-configuration ranking based on Silhouette

HP Configuration	Silhouette	Rank
0.1	-0.152281273	10
0.2	0.063106043	1
0.3	0.033847521	3
0.4	-0.048447818	6
0.5	0.029413262	5
0.6	0.056544477	2
0.7	0.032875458	4
0.8	-0.060192376	8
0.9	-0.148665823	9
1.0	-0.048828244	7

4 Results and Analysis

The Spearman correlation coefficient is a statistical measure that is used to assess the relationship between two sets of rankings [23]. It is perfectly applicable when we are dealing with ordinal data where the actual values between ranks do not give proper insights. The coefficient is calculated by assigning ranks to the observations in each set and then computing the correlation between these ranks. The Spearman correlation ranges from -1 to 1, with 1 indicating a perfect positive relationship between the rankings, -1 indicating a perfect negative relationship, and 0 suggesting no monotonic relationship. This coefficient allows us to determine whether there is a consistent pattern in the rankings based on the different internal evaluation strategies and the rankings based on ROC AUC score (considered the ground truth).

To calculate this correlation we use the formula for Spearman's correlation for data with tied ranks, where i = paired score, defined as [23].

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Tables 8,9,10 and 11 show the Spearman's rank-order correlation between 'rankings of the HP-configurations for 6 different internal evaluation strategies' and the 'HPconfiguration ranking based on ROC AUC score'. A correlation coefficient approaching a value of 1 is needed to verify that the chosen (best) hyperparameter setting is also correct based on the true outliers.

To analyse whether or not the results are only bound to a combination of a Feature selection algorithm and an Anomaly detection algorithm, we have set different combinations of Feature selection algorithms and Anomaly detection algorithms. Tables 8 and 9 show the results for the Laplacian score as feature selection algorithm, where Table 8 shows the OCSVM as an anomaly detector and Table 9 shows the K-Nearest Neighbour as an anomaly detector. Tables 10 and 11 show the results with SPEC as feature selection algorithm, where Table 10 shows the OCSVM as an anomaly detector and table 11 shows the K-Nearest Neighbour as an anomaly detector. In these tables the scores in bold show best ranking performance based on internal evaluation strategies. In other words, they show the scores for the best performing automatic feature selection method. We have the following main observation regarding the results in Tables 8-11.

First, Table 8 shows the correlation coefficients based on *Laplacian score* as feature selection algorithm and *OCSVM* as the anomaly detection algorithm. This combination has been used on all 5 benchmark sets. Results for the best automatic feature selection are divergent, all three internal evaluation strategies have at least once received the highest correlation score. UDR, however, outperformed other methods twice, with one score approaching a perfect correlation with the ground truth. Conclusively, there is not one automatic feature selection method that appears to be consistently better than others, given the 5 data with *laplacian score* and *OCSVM*.

Second, Table 9 shows the correlation coefficients based on *Laplacian score* as feature selection algorithm and *KNN* as the anomaly detection algorithm. The results in this table conspicuously show that the UDR method outperforms the other models. On all 5 datasets UDR

has the highest, or shared highest, correlation coefficient. Therefore it appears that given the combination of *Laplacian score* and *KNN*, the UDR method is a proper way to automatically select the best subset of features.

Third, Table 10 shows the correlation coefficients based on *SPEC* as feature selection algorithm and *OCSVM* as the anomaly detection algorithm. For this combination results vary a bit more, in comparison to Table 8. UDR results is the best method in 3 out of 5 datasets, whereas in the other 2 datasets cluster validation provide the best automatic feature selection. Therefore, not one specific method can be considered the best methood for autmatic feature selection for the combination of SPEC and OCSVM.

Fourth, Table 11 shows correlation coefficients based on *SPEC* as feature selection algorithm and *KNN* as the anomaly detection algorithm. The results in this table, again, show that UDR outperforms the other models (in 4 out of 5 times). Additionally, the coefficient scores achieved are relatively close to 1. For that reason we can say that in for the combination SPEC and KNN the UDR method would be the designated automatic feature selection method.

In summary, complete analysis of the Spearman's rank-order correlation coefficients for the different combinations of feature selection algorithms and anomaly detection algorithms, has led to an important takeaway. Namely, *the UDR model is the best performing automated feature selection model for unsupervised anomaly detection out of the three models that have been tested in this research.* This claim is based on results shown in Tables 8,9,10 and 11, where for the four different combinations of feature selectors and anomaly detectors, UDR has a significant number of times where it outperforms other models. Namely, 14 out 20. Furthermore its average coefficient value lays much higher than for the other models. Namely, 0.457 compared to 0.160, 0.044, 0.117, 0.406 and 0.041.

Table 8: Correlation coefficient between HP-configuration rankings based on AUC and the rankings of HP-configurations for internal evaluation strategies using *Laplacian score* and *OCSVM* on different datasets.

Dataset	**UDR**	**HITS**	**DB**	**XB**	**CH**	**S**
WDBC	0.406	-0.618	**0.600**	0.091	0.091	-0.091
Arrhythmia	-0.697	-0.593	-0.430	**0.733**	**0.733**	-0.492
Ionosphere	**0.345**	0.285	-0.188	-0.224	0.212	0.297
Letter Recognition	0.321	**0.333**	-0.261	-0.261	-0.261	0.103
KDDcup99, reduced	**0.988**	0.800	-0.079	-0.673	-0.673	0.430

Table 9: Correlation coefficient between HP-configuration rankings based on AUC and the rankings of HP-configurations for internal evaluation strategies using *Laplacian score* and *KNN* on different datasets.

Dataset	**UDR**	**HITS**	**DB**	**XB**	**CH**	**S**
WDBC	**0.764**	-0.215	-0.764	-0.764	-0.764	-0.067
Arrhythmia	**0.609**	0.262	0.505	**0.609**	0.596	-0.170
Ionosphere	**0.927**	0.309	**0.927**	**0.927**	**0.927**	0.139
Letter Recognition	**0.333**	-0.036	-0.176	-0.273	-0.273	-0.042
KDDcup99, reduced	**0.838**	**0.838**	0.130	0.093	0.093	0.235

Table 10: Correlation coefficient between HP-configuration rankings based on AUC and the rankings of HP-configurations for internal evaluation strategies using *SPEC* and *OCSVM* on different datasets.

Dataset	**UDR**	**HITS**	**DB**	**XB**	**CH**	**S**
WDBC	**0.652**	0.385	-0.152	-0.152	-0.152	-0.512
Arrhythmia	-0.669	-0.777	-0.097	**0.827**	**0.827**	-0.018
Ionosphere	0.188	-0.146	0.612	**0.758**	**0.758**	-0.042
Letter Recognition	**0.612**	0.394	0.067	0.103	0.103	0.115
KDDcup99, reduced	**0.787**	0.170	-0.097	0.778	0.778	0.043

Table 11: Correlation coefficient between HP-configuration rankings based on AUC and the rankings of HP-configurations for internal evaluation strategies using *SPEC* and *KNN* on different datasets.

Dataset	**UDR**	**HITS**	**DB**	**XB**	**CH**	**S**
WDBC	**0.802**	0.304	0.498	0.359	0.359	0.359
Arrhythmia	0.758	**0.782**	-0.418	-0.394	-0.394	-0.188
Ionosphere	**0.842**	0.176	-0.006	0.055	0.055	0.564
Letter Recognition	**0.648**	0.442	0.067	-0.055	-0.055	0.467
KDDcup99, reduced	**0.681**	0.107	0.146	-0.201	-0.201	-0.298

5 Conclusions and Further Research

This research has focused on a gap in the field of unsupervised anomaly detection. The missing piece in the existing body of knowledge on this topic that we refer to is; *automated feature selection for unsupervised anomaly detection.* We have created a pipeline that, given unlabelled dataset, returns the best feature selection for an anomaly detection task, based on different internal evaluation strategies.

For the preliminary results of this pipeline we have combined Laplacian score and SPEC (feature selectors) with OCSVM and KNN (anomaly detectors) on different data sets, for which we gathered the data and got results in the form of Spearman's rank order correlation coefficient. The results show that using UDR as the internal evaluation strategy in automatic feature selection yields the best output.

Be that as it may, the claim is made on the foundation of a limited number of experiments. This ties in with a critical note on this research, which is the number of experiments. A total of 600 (5x10x3x4) experiments were conducted in this research. These experiments involved 5 different datasets, 10 distinct HP configurations, 6 internal evaluation strategies, and 4 feature selection and anomaly detection algorithms. Therefore the conclusions that have been drawn may only count for this specific set of data. The reason for the limited experiments is that the complete construction of the pipeline using advanced internal evaluation strategies has never been done before, in automatic feature selection for unsupervised anomaly detection. Therefore, future research should focus on incrementing the number of datasets and increase the number of feature selection algorithms and anomaly detection algorithms. This in order to further prove, or disprove, that UDR model is a proper method for automatic feature selection.

Another significant improvement can be made in parts of the coding. Current output leads to automatic ranking of features for the different internal evaluation strategies. They therefore clearly show the best subset of features for a dataset with the chosen feature selection and anomaly detection algorithm. However, computation of the Spearman's rank-order has not yet been fully automated. Meaning that for every set of ranking outputs it had to manually be put into another part of the code to calculate the correlation between ranking output based on internal evaluation strategies and the ground truth. Improvements in this part would significantly reduce workload.

The strengths of this research lay in the well designed pipelines created for the UDR, HITS and Cluster validation metric models. This pipeline is the cornerstone in our research for automated feature selection. It enables us to systematically add datasets, feature selection algorithms and anomaly detection algorithms in future research. This is necessary to improve generalizability of the models that we used.

6 Acknowledgement

I would like to express my gratitude to Zhong Li for his dedication and support throughout the duration of this bachelor thesis. I am truly thankful for his daily supervision, as well as his commitment to taking the time for weekly meetings, which greatly contributed to this research. Zhong Li's guidance, from the initial stages to the very end, has been invaluable. His insightful advice and ability to propose effective solutions have been instrumental in shaping the directions for this thesis on autmatic feature selection.

I would also like to extend my thanks to Matthijs van Leeuwen for providing me with the opportunity to undertake this bachelor thesis in the area of data science under the supervision of Zhong Li.

References

[1] Akanksha Toshniwal, Kavi Mahesh, and R. Jayashree. Overview of anomaly detection techniques in machine learning. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 808–815, 2020.

[2] Digital Twin. Project 4, technology health management. https://www.digital-twin-research.nl/research/research-project-4/. 4.1 Feature and data subset selection for contextual anomaly detection using hybrid models.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[4] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7:1–30, 2020.

[5] Milan Luijken. Pyod: "the effects of feature selection on anomaly detection". *Leiden University*, 2022.

[6] Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies. *ACM SIGKDD Explorations Newsletter*, 25(1), 2023.

[7] Saúl Solorio-Fernández, J Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948, 2020.

[8] Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 13–24. SIAM, 2011.

[9] Anala A Pandit, Bhakti Pimpale, and Shiksha Dubey. A comprehensive review on unsupervised feature selection algorithms. In *International Conference on Intelligent Computing and Smart Communication 2019: Proceedings of ICSC 2019*, pages 255–266. Springer, 2019.

[10] Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30:891–927, 2016.

[11] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*, 2019.

[12] Towards Data Science. 7 evaluation metrics for clustering algorithms. https://towardsdatascience.com/7-evaluation-metrics-for-clustering-algorithms-bdc537ff54d2. [Online; accessed on 18th June 2023].

[13] IBM. Data labeling. https://www.ibm.com/topics/data-labeling. labeled data vs. unlabeled data.

[14] Emre Hancer, Bing Xue, and Mengjie Zhang. A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 53(8):4519–4545, 2020.

[15] Z. Li. Benchmark datasets. https://git.liacs.nl/liz16/f-son-ad-datasets-copy.git. 5 different datasets used.

[16] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Advances in neural information processing systems*, 18, 2005.

[17] scikit-learn. Scikit-learn: Machine learning in python. Available at https://scikit-learn.org/stable/, 2021. lap-score, Version 0.24.2.

[18] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157, 2007.

[19] Zhengwei Hu, Zhangli Hu, and Xiaoping Du. One-class support vector machines with a bias constraint and its application in system reliability prediction. *AI EDAM, Cambridge University Press*, 33(3):346–358, 2019.

[20] Abdenour Bounsiar and Michael G. Madden. One-class support vector machines revisited. In *2014 International Conference on Information Science & Applications (ICISA)*, pages 1–4. IEEE, 2014.

[21] Stat 509. Design and analysis trial. https://online.stat.psu.edu/stat509/lesson/18/18.3.

[22] Thanh Trung Nguyen, Uy Quang Nguyen, et al. An evaluation method for unsupervised anomaly detection algorithms. *Journal of Computer Science and Cybernetics*, 32(3):259–272, 2016.

[23] Laerd Statistics. Spearman's rank-order correlation. https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide-2.php. coefficient correlation.