Study and Evaluation of Unsupervised Algorithms used in Network Anomaly Detection

Juliette Dromard, Philippe Owezarski

Study and Evaluation of Unsupervised Algorithms used in Network Anomaly Detection

Juliette Dromard and Philippe Owezarski

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
owe@laas.fr

Abstract. Network anomalies are unusual traffic mainly induced by network attacks or network failures. Therefore it is important for network operators as end users to detect and diagnose them to protect their network. However, these anomalies keep changing in time, it is therefore important to propose detectors which can learn from the traffic and spot anomalies without relying on any previous knowledge. Unsupervised network anomaly detectors reach this goal by taking advantage of machine learning and statistical techniques to spot the anomalies. There exists many unsupervised network anomaly detectors in the literature. Each algorithm puts forward its good detection performance, therefore it is difficult to select one detector among the large set of available detectors. Therefore, this paper, presents an extensive study and assessment of a set of well known unsupervised network anomaly detectors, and underlines their strengths and weaknesses. This study overwhelms previous similar evaluation by considering for the comparison some new, original and of premier importance parameters as detection similarity, detectors sensitivity and curse of dimensionality, together with the classical detection performance, and execution time parameters.

Keywords: Unsupervised Network Anomaly Detection, Outlier Detection, Subspace PCA Method, Clustering algorithm, Curse of Dimensionality

1 Introduction

With the booming in the number of network attacks, the problem of network anomaly detection has received increasing attention over the last decades. Current detectors are mainly based on prior knowledge of the attacks or of the normal traffic like signature-based detectors or behavioral-based detectors. This knowledge must be continuously updated to protect the network as the nature of the attacks keeps changing in time to evade new network protections. However, building signatures or new normal profiles to feed these detectors takes time and money as this work is usually done by network experts. As a result, they are unable to deal with zero-day attacks and/or new network behaviors. To overcome these issues, a new generation of detectors has emerged which takes benefit of intelligent techniques which automatically learns from network traffic and allows

bypassing the strenuous human input: unsupervised network anomaly detectors. These detectors aim at detecting network anomalies in an unsupervised way, i.e. without any previous knowledge on the anomalies. These anomalies may be due to attacks (DOS, DDOS, network scan, worm, etc), to network failures or mis-configurations (route failures, traffic overload, imbalanced network traffic, etc.) or to some strange behaviors which should be monitored (use of multiple proxies, IP spoofing, etc.). Therefore, detecting these anomalies is of big interest for a network administrator. It can help him protect and gain an insight on its network.

Detectors rely on outlier detection algorithms which can be classified in three categories [9]: algorithms based on statistical models, algorithms based on spatial proximity and finally algorithms which deal with high dimensions. In this paper we analyse and evaluate the performance of 6 well known detectors, (two from each category): the Principal Component Analysis (PCA) subspace [16,23] and the robust PCA subspace method [14], DBSCAN [26], LOF (Local Outlier Factor) [3], SOD (Subspace Outlier Degree) [13] and UNADA [4].

The objective of this paper is to exhibit the practical state of the art in the domain of unsupervised anomaly detection. For this purpose, the major contribution of the paper deals with the evaluation kind that has been applied on the six detectors mentioned right over. It does not limit itself to the analysis of ROC curves and detection time, but tries to go further by considering their self and relative performance, especially when facing the same anomalies, but configured differently; the paper then deeply studies the sensitivity properties of these detectors. Given also the importance of the "Big data" keyword nowadays, with the need of analyzing huge amount of data containing many dimensions, the paper integrates in the evaluation of the detectors the study of their performance when facing such kinds of big data having a large number of dimensions. Given the large spectrum of the detectors considered in the paper, the evaluation methodology has been adapted depending on the detectors, trying as much as possible to remain fair despite evaluating slightly different features for each of the detection tools. The variability of the results that have been obtained puts forward the difficulty of making a right choice that can work in all conditions. It also points out the difficulty to parametrize such unsupervised detectors. We expect these results to show some of the research directions for improving the unsupervised detectors, and making them practically easy to use, and efficient in terms of global detection performance.

For providing such an evaluation, a valuable ground truth is required. It must be fair for providing the same realistic level of complexity for all the different algorithms implemented in the selected detection tools. That is why the ground truth must first contain very accurate labels for making the evaluation process relevant and accurate. It must also reproduce the difficulty of finding anomalies traffic that is quite small compared to gobal traffic. The way the ground truth is built is detailed in section 4. The existing dataset matching the most the expressed requirement is KDD'99 thanks to the quality of its labels, and despite the fact that it is quite old. However, analyzing the current state of the art in

traffic anomalies, it is not very far from the one in 1998, when KDD dataset was built: indeed, DDoS attacks, flash crowds or misconfigurations have similar effect on the traffic characteristics in 2017 and in 1998. New kinds of such anomalies are not very numerous. Despite not fully complete, we argue that traffic anomalies contained in KDD'99 represent a significant part of existing anomalies in 2017, and is enough for providing a high quality comparative evaluation of the 6 selected detectors. We then built our evaluation ground truth based on KDD'99 anomalies, but also adapting the background traffic to actual one, especially for reproducing the actual ratio between anomalous and background traffic.

To summarize, the main contributions of this paper are the following:

- It proposes a method to compare and study unsupervised network anomaly detectors.
- It proposes a new method to evaluate detectors sensitivity inspired by the Morris method.
- It gives guideline to parametrize these detectors.
- It points out the strengths and weaknesses of each detector.
- It uncovers important facts on the nature of network anomalies.

This paper is organized as follows. In a second section, unsupervised network anomaly detectors principle is presented. Then, a set of detectors are described and their configuration are discussed. A fourth section presents the detectors evaluation and discussed the obtained results. These latter are compared in terms of detection performance, detection similarity, execution time, detectors sensitivity and curse of dimensionality. Finally, section 5 concludes.

2 Unsupervised Network Anomaly Detection Principle

Existing unsupervised network anomaly detectors include two main steps, the preprocessing and the outlier detection steps. Some detectors may integrate a third and optional step: the anomalies post-processing.

The first step aims at capturing the network traffic, usually in consecutive time-bins and at processing it to build a data matrix X. Collected packets are aggregated in flows according to a specific flow key which can be, for example, the IP source, the IP destination or a combination of both. For each flow a set of statistics are computed like its number of IP destinations, of packets or of ICMPs. A normalized data matrix X of size $p * d$ is built, with d being the number of statistics (or dimensions) used to describe a flow and p the total number of flows (or points). We will keep this notation throughout the paper. The outlier detection step aims at detecting anomalous flows in the data matrix X using outlier detection algorithms. These algorithms aim at identifying flows which have different patterns from the rest of the traffic. This phase has received most of the researchers attention as the detectors intelligence relies in it.

The post-processing step aims at extracting and displaying information about the anomalies to assist network administrators in their task. This stage has received little attention for the moment although it is a crucial one. The post-

processing phase helps the network administrator understand, sort and classify the spotted anomalies in order to take appropriate counter-measures. Post-processing output can take different forms, for example in [4] the authors build signatures from the anomalies, in [16] they classify the anomalies using clustering techniques and in [11] they remove persistent anomalies to ease the network administrator task.

3 Outlier Detection Algorithms

To detect anomalous flows in the data matrix X, unsupervised network anomaly detectors rely on outlier detection algorithms. An outlier detection algorithm can either have a global view or a local view of the data. Thus, to evaluate a point abnormality level, a detector will either compare it to its neighbors (local view) or to the whole data (global view). Furthermore, a detector can either output a label for each point (normal vs abnormal point) or a score of outlierness. In the case of scores, the outlier detection algorithm must be followed by an additional step to extract anomalous points (flows) from scores. As stated in the introduction, outlier detection algorithms can be classified in three categories [9]: algorithms based on statistical models, algorithms based on spatial proximity and algorithms dealing with high dimensions.

3.1 Algorithms Based on Statistical Models

Outlier detection algorithms based on statistical models rely on the assumption that the data has been generated according to a statistical distribution. Outliers are then flows that deviate strongly from this distribution. Many statistical approaches have been applied to unsupervised network anomalies detection such as histograms [12], EM-clustering [25], the PCA subspace method [16,23] and the Gaussian mixture model [2].

The PCA subspace method has been extensively used for network anomaly detection. This approach divides the whole space of dimension d in two subspaces: the normal subspace made up of the k principal component (PC) directions of the data matrix X and the abnormal subspace made up of the $d-k$ PC directions left. There exists variants of the PCA subspace method [23], however, in the context of this study, we only evaluate the one proposed in [16] which has been extensively studied. In this approach, one score of outlierness is computed for each point. Once projected on the abnormal subspace, a point's score is equal to its l^2 norm. Points with a high score are more likely to follow a pattern which does not conform to the normal or natural one. This method takes as input one parameter k which defines the normal subspace dimension. The PCA subspace method complexity is in $O(p.d^2)$.

In [22], Ringberg et al. highlight that k must be picked up such that the k dominant PC directions capture most of the total deviation of the data to get good detection performance.

Algorithm	View	Output	HD*	Complexity	Parameter	Parameters setting
Sub. PCA [16]	global	score	no	$O(p.d^2)$	k: nb of PC directions	must capture most of the total deviation of the data
DBSCAN [7]	global	label	no	$O(p.log(p))$	r: radius	percentage of the distance between the space two farthest points
					$minPts$: min nb of points to form a cluster	percentage of the total number of points
LOF [3]	local	score	no	$O(p.log(p))$	nn: nb of nearest neighbors	percentage of the total nb of flows
UNADA [4]	global	score	yes	$O(d^2.p.log(p))$	r: radius (different for each subspace)	percentage of the distance between the subspace two farthest points
					$minPts$: min nb of points to form a cluster	percentage of the total number of points
SOD [13]	local	score	yes	$O(d^2.p^2)$	α_{lof}	advice 0.8
					nn: nb of nearest neighbors	percentage of the total nb of flows
					l: number of reference points	percentage of the total nb of flows
Naive alg.	global	label	yes	$O(d.p)$	α_{naive}: nb of standard deviations	set high enough to only detect flows with extreme values
HD: deal with high dimensions; nb is used for "numbers"						

Table 1. Outlier detection algorithms

Some recent articles have underlined that PCA-based detectors suffer from the contamination subspace problem. This phenomena appears when some large outliers are included in the measured traffic data. These latter contaminate the subspaces and as a result, a large part of the anomalies are projected onto the normal subspace and are not detected. In order to solve the subspace contamination problem, [14] use robust PCA mechanisms to obtain PCs which are not influenced by the existence of outliers. In the following, we use the GRID algorithm [6] to get robust PCs as it does not take any input parameter and finds good quality PCs in a reasonable time.

3.2 Algorithms based on on spatial proximity

Many outlier detection algorithms rely on models based on spatial proximity like DBSCAN [7] [26], K-mean [28], LOF [3], etc. Algorithms based on spatial proximity should be used with an index like the R-tree [20] or the k-d tree [27] to improve their time complexity. These detectors are based on the idea that points isolated from the others are outliers.

DBSCAN [7] is a density-based clustering algorithm which groups points that are closely packed together in clusters. Points that lay in low-density regions are considered as outliers. It can discover clusters of various shapes and sizes from a large amount of data which contains noise. It takes two input parameters r and $minPts$ which respectively describe the neighborhood radius of a point and the minimum number of points to form a cluster. There exists no rule of thumb to fix DBSCAN parameters and its configuration may differ with the data and the problem considered. In order to avoid that DBSCAN groups flows which belong to similar anomalies in the same cluster, the parameter $minPts$ should be superior to the maximum number of flows induced by similar attacks. For example, if 9 flows are induced by SYN attacks, then $minPts$ should be superior to 9 so that they do not form a cluster. Furthermore, as anomalies are flows which deviate strongly from the others, r must be chosen large enough so that points which are slightly different from the majority belong to a cluster. Thus, we propose to set r as a percentage of the distance between the space two farthest points and $minPts$ as a percentage of the total number of flows. DBSCAN time complexity is $O(p.log(p))$ when used with an R-tree index.

LOF [3] is a local spatial-based approach which assigns to each point an outlier factor representing its degree of outlierness regarding its local neighborhood. A point whose density is lower to that of its nn nearest neighbors is considered as an outlier. Thus, LOF is able to deal with regions of different densities. It takes as input one parameter nn, which represents the number of nearest neighbors considered to evaluate a point's abnormality. The value of nn must be carefully chosen. Indeed, if it is too low, LOF may then compare an anomalous flow with only similar anomalous flows, i.e. with flows generated by the same type of attack and may therefore not detect them as outliers. To overcome this issue, nn must be set larger than the maximal number of flows induced by similar attacks. We propose to fix it as a percentage of the total number of flows. For medium to high-dimensional data, the algorithm provides an average complexity of $O(p.log(p))$ [3].

3.3 Algorithms Dealing with the Curse of Dimensionality

In high dimensional data, distance between points become meaningless: the proportional difference between the farthest point distance and the closest point distance vanishes. This phenomena is called curse of dimensionality and can have an important impact on detectors detection performance. Some outlier detection algorithms have been specifically devised to deal with this curse like UNADA [4] or SOD [13]. To deal with this curse UNADA relies on a divide and conquer approach. It divides the space made up of d dimensions in $N = \binom{d}{2}$ two-dimensional subspaces. It then applies DBSCAN on each subspace. It finally combines the N obtained partitions in one final partition and computes for each point a score. For each point, it computes a core which is the sum of its distance to the biggest cluster in every subspace. UNADA has the same input parameters

as DBSCAN: a radius r and $minPts$. However, the value of r must be adapted to every subspace.

SOD [13] is a local outlier algorithm which deals with high dimensions by selecting in an intelligent way subspaces to compute each point's score. It computes a score for each point which reflects how well it fits to the subspace that is spanned by a set of l reference points. The l points which shares the highest number of nearest neighbors with a point form its l reference points. For each point, SOD computes a subspace made up of the set of dimensions whose variance is low with respect to its l reference points. SOD takes three input parameters: α_{lof} a threshold to decide about the significance of a dimension, l the number of reference points and nn the number of nearest neighbors. Authors advise to set α_{lof} at 0.8. Furthermore, to avoid comparing an anomalous point with only similar anomalous points, l should be chosen much higher than the maximum number of flows induced by a same type of attack. SOD's time complexity is $O(d.p^2)$.

For the sake of comparison, we propose a naive outlier detection algorithm which aims at detecting points with extreme values. For each dimension, this algorithm detects as outliers the points which are α_{naive} standard deviations from the median. As it deals with one dimension at a time, our naive algorithm should be able to deal with high dimensions. Table 1 summarizes detectors characteristics.

3.4 Detectors based on scores

For algorithms which output scores (LOF, PCA subspace, SOD, UNADA), a final step is required to extract outliers. A threshold th is set and all the scores which are above th are considered as outliers. We have identified in the literature three main methods to set th:

- The knee method [4,5]. This approach consists in plotting the sorted outlier scores to get a convex curve. Usually, a knee in the curve can be observed indicating a change in the nature of the scores. The threshold is set at the curve knee point.
- The quantile based method [23,22]. The threshold is set at the q-qantile of the scores empirical distribution. For example in [23] and in [22], they fix it at the 0.9899 quantile and the 0.90 quantile respectively. However, this method implies that the percentage of anomalies in the data is known in advance, which is, in most cases, unrealistic.
- The statistical hypothesis testing method. This method assumes that normal flows follow a specific data distribution for which a statistical hypothesis testing exists. The threshold is set at the test $1 - \alpha$ confidence level. This limit corresponds to a false alarm rate of α, if the starting assumptions are satisfied. For example in [15,16], the authors assume that normal flows follow a multivariate Gaussian distribution which allow them to apply the Q-statistic developed in [10]. However, it has not yet been demonstrated that network traffic follows any specific distribution.

Algorithm	Parameter	Value	Range
DBSCAN	r	not fixed	1% to 20%
	$minPts$	10% of the total nb of flows	1% to 20%
LOF	nn	20% of the total nb of flows	10% to 50%
PCA subspace	k	the k first PCs capture at least 90% of the total deviation	85 to 98%
rob. PCA subspace	k	the k first PCs capture at least 90% of the total deviation	85 to 98%
UNADA	radius r	10% of the distance between the subspace two farthest points	1 to 10%
	$minPts$	10% of the total nb of flows	1 to 20%
SOD	nn	20% of the total nb of flows	10% to 40%
	l	10% of the total nb of flows	10% to 40%
	α_{sod}	0.8	
NAIVE	α_{naive}	not fixed	0.5 to 3

Table 2. Detectors parameters

For most outlier detection algorithms, there exists no guideline to set their parameters. Good sense and a good understanding of the current problem are essential to set detectors parameters and get relevant outcomes.

4 Evaluation on our new KDD'99 inspired dataset

In the field of network anomaly detection, as pointed out in [24], there is a lack of available public ground truth. In the literature, two main public available ground truths are often cited: the KDD99 ground truth [1] (summary of the DARPA98 traces) and the MAWI ground truth [8]. Many other datasets exists but, for the moment, do not provide the same amount of data, the same level of labels, are not easy to get, etc. The KDD99 contains multiple weeks of network activity from a simulated Air Force network, generated in 1998. Although the KDD99 dataset is quite old, it is still considered as a landmark in the field, because of the accuracy of the provided labels. On the contrary, the MAWILab dataset is more recent and is still being updated. It consists of labeled 15 minutes network traces collected daily from a trans-Pacific link between Japan and the United States. However, the MAWILab ground truth is questionable as it has been obtained by combining the results of four ancient unsupervised network anomaly detectors [8]. indeed, labels are often not very relevant; for example many anomalies are labeled as 'HTTP traffic'. In addition, after manual inspections, some anomalous flows do not seem to exhibit unusual patterns.

For all these reasons, we have decided to perform our evaluation on a dataset built out the KDD99 dataset. This choice has been made because it outputs consistent labels that are fully accepted by the community. The evaluation has then been performed on a portion containing 10% of KDD99 dataset which contains 23 different types of attack, see [19] for more information on these

attacks. To obtain this dataset, packets have been aggregated according to the TCP connection they belong to. Each flow is described by 41 attributes, 34 of which are numeric and 7 are categorical. As detectors do not deal with categorical variables, they have been turned into dummy variables, lifting the total number of variables to 118. The dataset cannot be used as it is, due to a too large number of anomalous flows; no detector based on outlier detection techniques can possibly detect the attacks as they are not rare. This problem could have been solved by aggregating the flows into another level (by IP source for example), but this is not possible with the KDD99 dataset as the IP addresses are not displayed. To overcome this issue and as in [26,17,5,21], we have, selected randomly some flows, so that the percentage of anomalous flows stays under a certain threshold. We have built two datasets. The first one "dataset1" is made up of 1000 flows and includes 160 attacks, there is at most 8 flows for each type of attack. The second one "dataset2" is made up of 10000 flows and includes 979 attacks, there is at most 80 flows for each type of attack. Some dimensions may have a larger range than others, as a consequence they have a higher weight and may hide other features. To overcome this issue, both datasets are normalized using the max-min normalization so that each dimension scales in [0,1].

4.1 Evaluation in terms of detection performance

In a first time, the algorithms are compared in terms of detection performance using the Area Under the ROC curve (AUC) measure. A ROC curve is obtained by plotting the true positives rate (TPR) against the false positive rate (FPR) at various parameters settings. The AUC takes its value in [0,1]; an AUC of 1 represents a perfect detector and an AUC of 0.5 a detector with complete random guess. The parameters used for this evaluation are displayed in table 2. The column "Range" will be used later in this paper. The ROC curve points have been computed by varying

- the threshold th for algorithms which output scores (SOD, LOF, UNADA, and PCA).
- the radius r for DBSCAN.
- the parameter α_{naive} for the naive outlier algorithm.

Table 3 and figure 1 presents the AUC obtained by each detector for each dataset. It can be noticed that the PCA subspace method has the worst detection performance, this result can be explained by the contamination subspace problem. This assumption is confirmed by the robust PCA subspace method results. Indeed, by resolving the subspace contamination problem, this letter achieves the best detection performance among all the detectors with an AUC superior to 0.97 for both dataset. Except the PCA subspace method, every algorithm achieves good detection performance with an AUC superior or equal to 0.9. Naive detector AUC is superior to 0.96 for both experiments, therefore it outperforms most of the detectors. This result implies that most network anomalies in KDD99 dataset possess an extreme value in at least one dimension. Some

	dataset1				dataset2			
	AUC	TPs	FPs	Time	AUC	TPs	FPs	Time
UNADA	0.90	146	85	98s	0.93	922	380	3h 8m
LOF	0.90	160	207	2.5s	0.97	894	361	17m
PCA	0.71	107	75	55ms	0.70	638	1188	454ms
rob. PCA	0.97	158	65	9s	0.97	895	478	3m 43s
SOD	0.96	153	70	50s	0.90	915	1612	4h 30m
NAIVE	0.96	160	57	20ms	0.97	894	259	68ms
DBSCAN	0.94	149	47	430ms	0.96	902	232	58s

Table 3. Detectors AUC, number of TPs and FPs and execution time

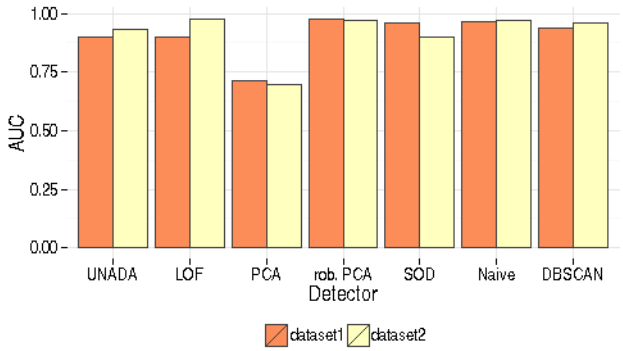

Fig. 1. Detectors AUC

further studies on other datasets should be carried out to check whether this observation can be extended to every network anomaly. If it is the case, this puts into question the use of complex algorithms to detect network anomalies.

In the following, each detector is set at its best setting. A detector best setting is defined as the setting which maximizes its informedness. The informedness is a statistical measure of the performance of a binary classification test which considers equally the TPR and FPR. It takes its value in [1,-1] and is computed as follows:

$$informedness = TPR - FPR \tag{1}$$

Figure 2 displays detectors ROC curve obtained with dataset1. The square on each detector curve represents the results obtained at the detector best setting. A visual analysis of these figures shows that the maximum informedness is a good measure to select each detector best setting.

The AUC provides information about the proportion of each detector TPR and FPR, however, this information is not sufficient to fully evaluate a detector performance. Indeed, as pointed out in [24], even with a low FPR, the number

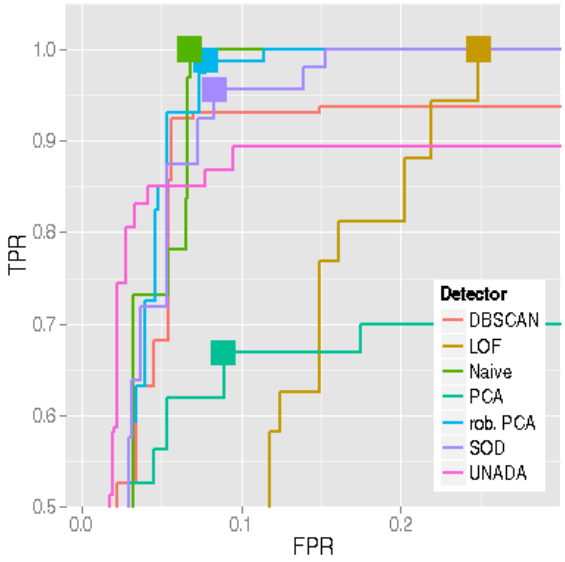

Fig. 2. Detectors ROC curve and their point of maximum informedness obtained with dataset1

of false negatives (FNs) generated by a detector may be substantial and can overwhelm a network administrator.

Figure 3 displays detectors number of TPs and FPs obtained at their best setting according to the informedness. It can be noticed that even with a high AUC (>0.9), LOF and SOD, in dataset1, get a high number of FPs; their number of FPs is superior to their number of TPs. Such a situation may lead the network administrator to mis-classify and interrupt many normal flows.

4.2 Evaluation in terms of execution time

A detector execution time is a very important parameter to consider while selecting a network anomaly detector. Indeed, the faster the detector identifies attacks, the quicker the network administrator can take relevant counter-measures and the less important the damages on the network are. Figure 4 depicts the execution time of each algorithm for both datasets, the y-axis is in log scale. These results have been obtained on a single machine with 16 GB of RAM and an Intel Core i5-4310U CPU 2.00GHz. As expected the execution time increases with the data size. As a reminder, dataset1 is made up of 1000 flows and dataset2 of 10,000 flows. The obtained results are logical according to detectors complexity displayed in table 1.

It can be noticed that UNADA and SOD do not scale well with the number of flows; for dataset2, UNADA completes the detection in 3 hours and SOD in 4

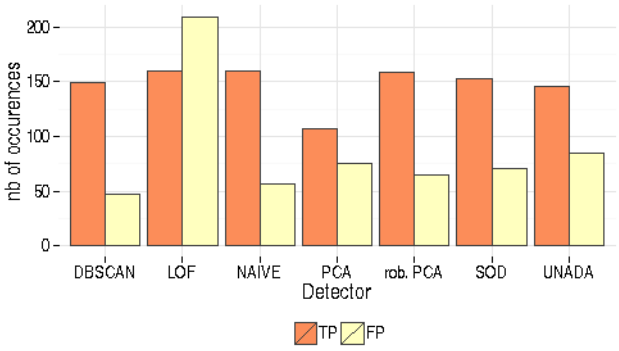

Fig. 3. Detectors number of true and false positives obtained with dataset1

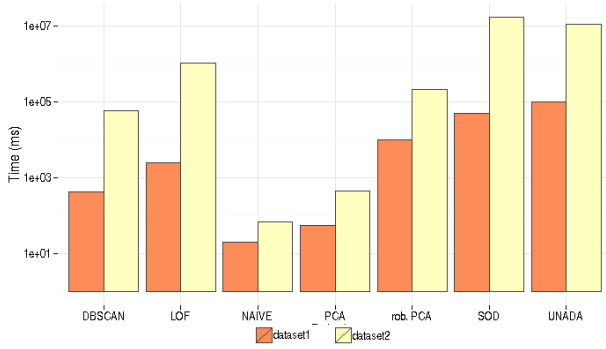

Fig. 4. Detectors execution time in milliseconds

hours and a half. On the other hand, the naive and the PCA subspace detector complete in less than a second.

5 Evaluation in terms of similarity

To evaluate the similarity between the anomalies found by the different algorithms we use the Jacquard index (JI). The JI measures the similarity between two finite sets and is defined as their intersection size divided by their union size. Thus, if A and B are the set of anomalies identified by two different detectors, their similarity according do the JI is computed as follows:

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{2}$$

If JI is close to one then the detectors are very similar and if it is close to 0 then they are considered as very dissimilar. Figure 5 displays the similarity between

the TPs of the different detectors for dataset1 (similar results have been obtained for dataset 2). It can be noticed that the JI is high for every algorithm (all the squares are red) in both datasets except the PCA subspace method as this latter has bad performance in terms of of TPR and FPR. This implies that detectors mainly find the same anomalies.

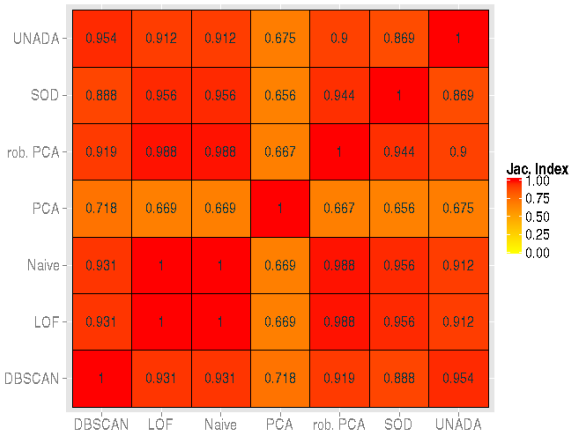

Fig. 5. Similarity between detectors TPs in dataset1

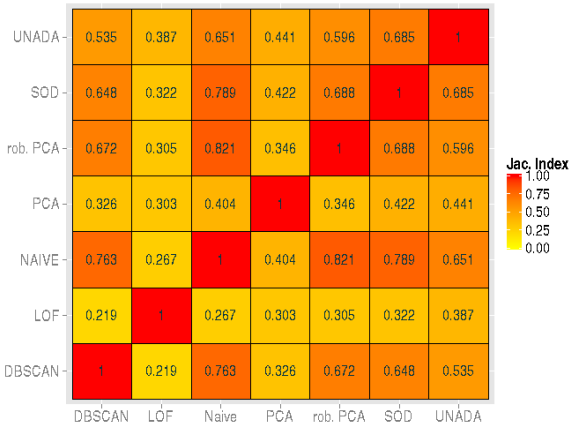

Fig. 6. Similarity between detectors FPs in dataset1

Figure 6 displays the similarity between the FPs found by the detectors for dataset 1. One can observe that the JI is often very low (many squares are yellow), which implies that their FPs are different. Thus, it would be interesting to combine the outputs of these different algorithms to keep only the anomalies found by most detectors. As the similarity between their FPs is very low, most FPs would then be discarded and as the similarity between their TPs is high, most TPs would be kept. Therefore, combining detectors output would allow improving the overall detection performance by reducing the number of FPs while maintaining a high number of TPs.

5.1 Evaluation in terms of curse of dimensionality

To evaluate detectors capacity to deal efficiently with high dimensions, we propose to evaluate their detection performance on dataset1 to which noisy dimensions are added. As in [29], the noisy dimensions are generated with a random uniform distribution which takes its values in [0,1].

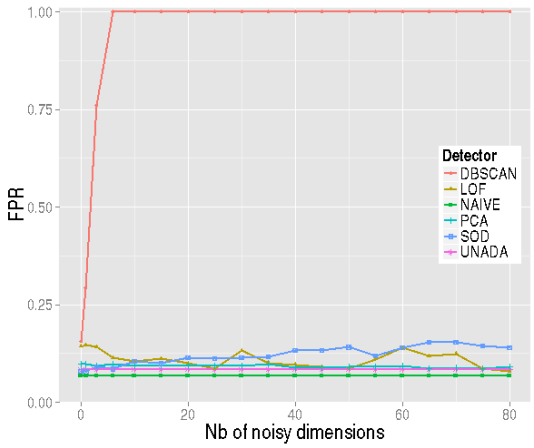

Fig. 7. Detectors FPR according to the number of added dimensions to dataset1

Figure 7 displays each detector FPR as a function of the number of added noisy dimensions. It can be noticed that noisy dimensions have no impact on UNADA and the naive algorithm performance. It can be explained by the fact that they both divide the space in subspaces of low dimensions and process them independently. PCA-based methods show little sensitivity to high dimensions, especially the robust PCA subspace detector. As it considers neighborhood rather than distance, LOF reacts well to the curse of dimensionality: a point neighbors stay the same when noisy dimensions are added to the data. Even though SOD has been devised to deal with high dimensions, its FPR tends to increase when noisy dimensions are added. Even though DBSCAN radius r is re-computed each

time some new dimensions are added (it is set at 10% of the distance between the space two farthest points), DBSCAN is the detector which suffers the most from the curse. With the increase in the number of noisy dimensions, points tend to move away from each other. As a result, DBSCAN identifies them all as outliers. Before adding these noisy dimensions, the curse had no effect on DBSCAN, even though KDD99 has many dimensions. This phenomenon can be explained by the fact that each dimension in KDD99 brings mainly information and few noise. Similar behaviors have been observed in [29]. xplained by the fact that each dimension in KDD99 brings mainly information and few noise. Similar behaviors have been observed in [29].

5.2 Evaluation in terms of parameters sensitivity

This section aims at evaluating and comparing detectors sensitivity. For each detector, we want to determine if its input parameters can be easily set such that it gets good detection performance and therefore a high informedness. Even with a high AUC, a detector can be unable to detect correctly anomalies if it is badly configured. A detector, very sensitive to its input parameters, may be very difficult to parameterize. As a result, a network administrator may fail in configuring it, the detector output becomes then useless. Therefore, the sensitivity of a detector is an important parameter to take into account even though it is rarely considered in the current literature.

To evaluate whether a detector can be easily configured, we propose an approach inspired by the Morris method [18]. This latter is a sensitivity analysis method which evaluates the influence of each input parameter on the output of a function. It computes for each parameter two statistics, its mean and its standard deviation impact on the output function.

We have modified the Morris method so that it applies to the analysis of detectors sensitivity. Our method aims at evaluating the impact of the input parameters of each detector on its informedness. To reach this goal, we have defined for each input parameter of each detector a range of possible "values" (see table 2). By possible values, we mean values which could have been chosen by any "reasonable" expert in the field.

We apply each detector many times on the dataset1. For each input parameter described in table 2, its range of possible values is discretized. Each detector is launched as many times as there are possible combinations of its input parameters. Finally, two statistics are computed to evaluate each detector: its informedness mean and standard deviation. The informedness standard deviation of a detector captures its sensitivity to its input parameters whereas its informedness mean provides an indication on its average detection performance. A detector is all the more easy to configure that its informedness mean is high and its informedness variance is low. As explained previously, unsupervised detectors either output labels or scores for each flow. For detectors which output scores, an extra step is required to extract anomalies from scores. Indeed, there is no clear gap between anomalous and normal flows scores. Therefore, extracting anomalies from scores is a difficult task.

To evaluate the sensitivity of detectors which output scores, we use the "best threshold method". This method sets the threshold, used to extract the anomalies from scores, at the value which maximizes the detector informedness. This method implies that flows labels are known in advance.

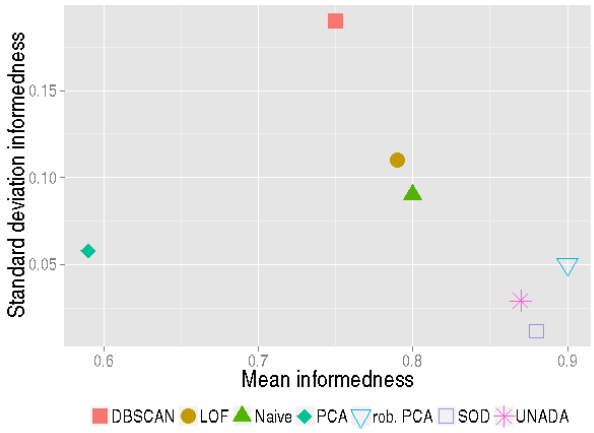

Fig. 8. Comparison of detectors sensitivity

Figure 8 displays the results of the detectors sensitivity analysis. Once again the PCA subspace method has bad performance due to its subspace contamination problem: its mean and standard deviation informedness is low. It implies that any configuration of this detector may lead to poor performance.

It can be noticed that DBSCAN standard deviation informedness is high which implies that it is difficult to configure correctly. Figures 9 and 10 can explain these results. Figure 9 displays DBSCAN informedness according to its radius (which is set as a percentage of the distance between the space two farthest points) with different $minPts$ values. It clearly shows that its informedness rises when the radius increases till a certain point. This is maybe because a larger radius increases the number of "normal" flows which belong to a cluster. Figure 10 depicts DBSCAN informednesss according to its minPts (which is set as a percentage of the total number of flows) with different values of radius. It can be noticed that when minPts increases, DBSCAN informedness tends to decrease which can be explained by the fact that fewer "normal" flows belong then to a cluster.

LOF standard deviation informedness is moderately high, it implies that it is quite sensitive to its input parameter nn. This sensitivity can be explained by its local view. Indeed, when nn is low, the probability that an anomalous point is compared only to other anomalous points is high. As a consequence, it may not appear as an outlier. This phenomena is illustrated by figure 11 which

displays LOF informedness according to its parameter value nn (which is set as a percentage of the total number of flows).

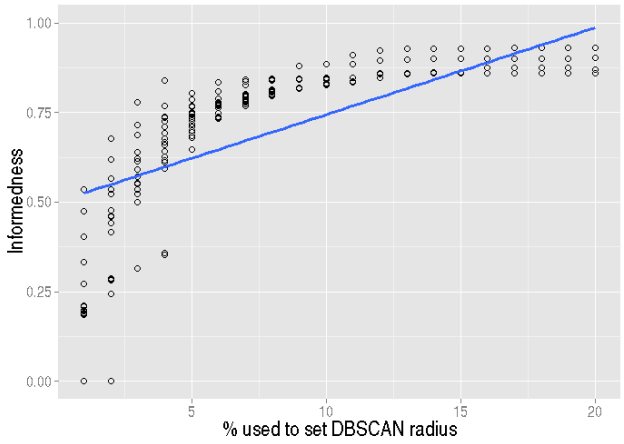

Fig. 9. DBSCAN informedness according to its radius.

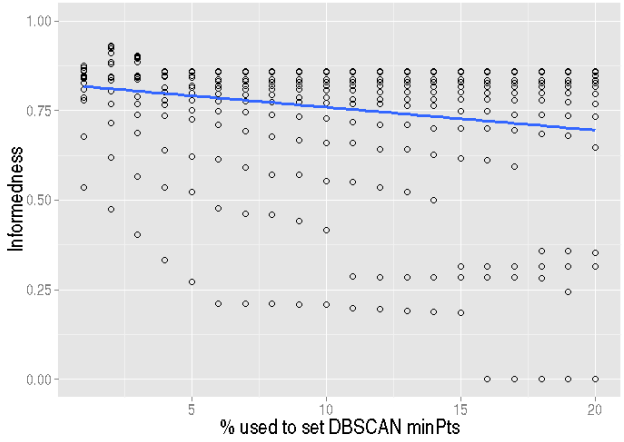

Fig. 10. DBSCAN informedness according to its minPts.

In depth studies (not presented in the paper because of space limitation) exhibit that DBSCAN informedness rises when the radius increases till a certain point. This is maybe because a larger radius increases the number of "normal" flows which belong to a cluster. It can also be noticed that when minPts increases,

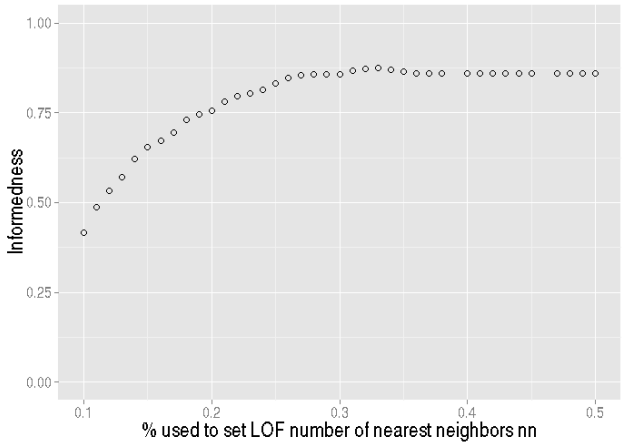

Fig. 11. LOF informedness according to the value of its input parameter nn.

DBSCAN informedness tends to decrease which can be explained by the fact that fewer "normal" flows then belong to a cluster.

Similarly, these studies show that LOF standard deviation informedness is moderately high. It implies that it is quite sensitive to its input parameter nn. This sensitivity can be explained by its local view. Indeed, when nn is low, the probability that an anomalous point is compared only to other anomalous points is high. As a consequence, it may not appear as an outlier.

6 Conclusion

This paper presents a comparison of different unsupervised network anomaly detectors in terms of detection performance, detector sensitivity, execution time and curse of dimensionality. It also proposes some guidelines to configure them. It points out the challenges raised by the extraction of anomalies from scores. Every detector except the PCA subspace method reaches very good detection performance in terms of AUC. However, at the light of other parameters some detectors may be difficult to apply in real life due to their high execution time like UNADA and SOD or their high sensitivity to their input parameters like DBSCAN. This study highlights the importance of using many parameters to evaluate a network detector and therefore underlines the weakness of evaluations only based on ROC curves. The results have pointed out that every network anomaly selected from the KDD99 dataset has an extreme value in at least one dimension and can therefore be easily identified by a naive algorithm. Some further studies on other datasets should be carried out to check whether this observation can be extended to every network anomaly. Among every algorithm the robust PCA subspace method has shown very good performance in terms of detection, input parameters sensitivity and robustness to high dimensions. To

reach near to real time detection, the robust PCA subspace can re-use the PCs directions multiple times assuming that the normal space changes little in time. Anomaly detection is the first step to protect the network. The spotted anomalies must then be processed by the network administrator so that it takes relevant counter-measures. An important effort should now be made to take advantage of detectors output and to propose solutions to identify anomalies root causes.

References

1. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed: 2016-02-03.
2. M. Bahrololum and M. Khaleghi. Anomaly Intrusion Detection System Using Gaussian Mixture Model. In *Convergence and Hybrid Information Technology, ICCIT '08.*, volume 1, pages 1162–1167, Nov 2008.
3. Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying Density-based Local Outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.
4. P. Casas, J. Mazel, and P. Owezarski. *NETWORKING 2011: 10th International IFIP TC 6 Networking Conference*, chapter UNADA: Unsupervised Network Anomaly Detection Using Sub-space Outliers Ranking, pages 40–51. Springer Berlin Heidelberg, 2011.
5. P. Casas, J. Mazel, and P. Owezarski. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. *Computer Communications*, 35(7):772 – 783, 2012.
6. C. Croux, P. Filzmoser, and M.R. Oliveira. Algorithms for Projection-Pursuit robust principal component analysis . *Chemometrics and Intelligent Laboratory Systems*, 87(2):218 – 225, 2007.
7. M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
8. R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In *ACM CoNEXT '10*, Philadelphia, PA, 2010.
9. A. Zimek H-P. Kriegel, P. Kröger. Outlier Detection Techniques. In *Tutorial Notes: SIAM SDM 2010, Columbus, Ohio*, 2010.
10. D. R. Jensen and H. Solomon. A Gaussian Approximation to the Distribution of a Definite Quadratic Form. *Journal of the American Statistical Association*, 67(340):898–902, 1972.
11. K. Julisch. Clustering Intrusion Detection Alarms to Support Root Cause Analysis. *ACM Transactions on Information and System Security*, 6:443–471, 2003.
12. A. Kind, M.P. Stoecklin, and X. Dimitropoulos. Histogram-based traffic anomaly detection. *Network and Service Management, IEEE Transactions on*, 6(2):110–121, June 2009.
13. H-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, year = 2009, publisher=Springer Berlin Heidelberg, pages=831–838,*, chapter Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data.
14. R. Kwitt and U. Hofmann. Unsupervised Anomaly Detection in Network Traffic by Means of Robust PCA. In *Computing in the Global Information Technology*, pages 37–37, March 2007.

15. A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-wide Traffic Anomalies. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 219–230. ACM, 2004.
16. A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Computer Communication Review*, 35(4):217, 2005.
17. K. Leung and C. Leck. Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38*, ACSC '05, pages 333–342, Darlinghurst, Australia, 2005. Australian Computer Society, Inc.
18. Max D. Morris. Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics*, 33(2):161–174, April 1991.
19. A. A. Olusola, A. S. Oladele, and D. O. Abosede. Analysis of KDD 99 Intrusion Detection Dataset for Selection of Relevance Features. In *World Congress on Engineering and Computer Science*, pages 162–168, 2010.
20. Hans peter Kriegel, Ralf Schneider, Bernhard Seeger, and Norbert Beckmann. The R*-tree: an efficient and robust access method for points and rectangles. *Sigmod Record*, 19:322–331, 1990.
21. L. Portnoy, E.Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, pages 5–8, 2001.
22. H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for Traffic Anomaly Detection. *SIGMETRICS Perform. Eval. Rev.*, 35(1):109–120, June 2007.
23. M-L Shyu, S-C Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *IEEE Foundations and New Directions of Data Mining Workshop*, pages 171–179, 2003.
24. R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, 0(May):305–316, 2010.
25. I. Syarif and G. Prugel-Bennett, A. andWills. *Networked Digital Technologies: 4th International Conference*, chapter Unsupervised Clustering Approach for Network Anomaly Detection. Springer Berlin Heidelberg, 2012.
26. T. M. Thang and J. Kim. The Anomaly Detection by Using DBSCAN Clustering with Multiple Parameters. In *Information Science and Applications (ICISA), 2011 International Conference on*, pages 1–5, April 2011.
27. J. A. Tsakok, W. Bishop, and Af. Kennings. kd-Tree traversal techniques. In *Interactive Ray Tracing*, pages 190–190, Aug 2008.
28. Y. Yasami, S. Khorsandi, S. P. Mozaffari, and A. Jalalian. An unsupervised network anomaly detection approach by k-means clustering & id3 algorithms. In *Computers and Communications*, pages 398–403, July 2008.
29. A. Zimek, E. Schubert, and H-P. Kriegel. A Survey on Unsupervised Outlier Detection in High-dimensional Numerical Data. *Stat. Anal. Data Min.*, 5(5):363–387, October 2012.