UNADA: Unsupervised Network Anomaly Detection using Sub-Space Outliers Ranking

Pedro Casas[1,2], Johan Mazel[1,2], and Philippe Owezarski[1,2]

[1]CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
Email: {pcasashe,jmazel,owe}@laas.fr

Abstract. Current network monitoring systems rely strongly on signature-based and supervised-learning-based detection methods to hunt out network attacks and anomalies. Despite being opposite in nature, both approaches share a common downside: they require the knowledge provided by an expert system, either in terms of anomaly signatures, or as normal-operation profiles. In a diametrically opposite perspective we introduce UNADA, an Unsupervised Network Anomaly Detection Algorithm for knowledge-independent detection of anomalous traffic. UNADA uses a novel clustering technique based on Sub-Space-Density clustering to identify clusters and outliers in multiple low-dimensional spaces. The evidence of traffic structure provided by these multiple clusterings is then combined to produce an abnormality ranking of traffic flows, using a correlation-distance-based approach. We evaluate the ability of UNADA to discover network attacks in real traffic without relying on signatures, learning, or labeled traffic. Additionally, we compare its performance against previous unsupervised detection methods using traffic from two different networks.

Key words: Unsupervised Anomaly Detection, Sub-Space-Density Clustering, Evidence Accumulation, Outliers Detection, Abnormality Ranking.

1 Introduction

Network traffic anomaly detection has become a vital network building-block for any ISP in today's Internet. Ranging from non-malicious unexpected events such as flash-crowds and failures, to network attacks such as Denials-of-Service (DoS/DDoS), network scans, and spreading worms, network traffic anomalies can have serious detrimental effects on the performance and integrity of the network. The principal challenge in automatically detecting and analyzing traffic anomalies is that these are a moving target. It is virtually impossible to precisely define the set of anomalies that may arise, especially in the case of network attacks, because new attacks as well a new variants of already known attacks are continuously emerging. A general anomaly detection system should therefore be able to detect a wide range of anomalies with diverse structure, using the least amount of previous information, ideally no information at all.

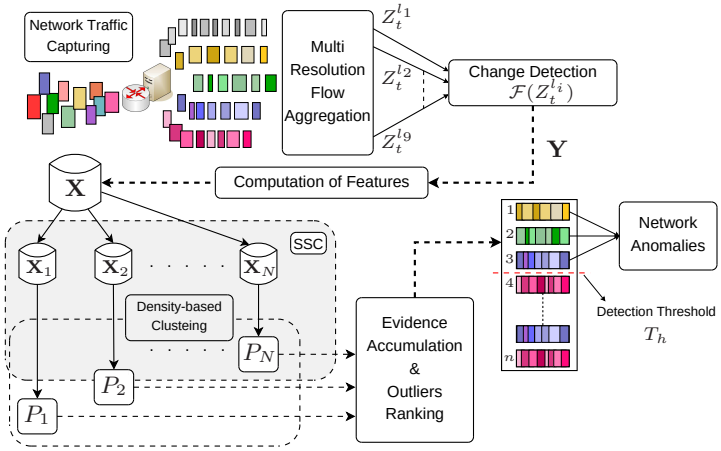


Fig. 1. High-level description of UNADA.

Two different approaches are by far dominant in current research literature and commercial detection systems: signature-based detection and anomaly detection. Signature-based detection is the de-facto approach used in standard security devices such as IDSs, IPSs, and firewalls. When an attack is discovered, generally after its occurrence during a diagnosis phase, the associated anomalous traffic pattern is coded as a signature by human experts, which is then used to detect a new occurrence of the same attack. Signature-based detection methods are highly effective to detect those attacks which they are programmed to alert on. However, they cannot defend the network against new attacks, simply because they cannot recognize what they do not know. In addition, building new signatures is a resources-consuming task, as it involves manual traffic inspection by human experts.

On the other hand, anomaly detection uses labeled data to build normal-operation-traffic profiles, detecting anomalies as activities that deviate from this baseline. Such methods can detect new kinds of network attacks not seen before. Nevertheless, anomaly detection requires training for profiling, which is time-consuming and depends on the availability of purely anomaly-free traffic datasets. Labeling traffic as anomaly-free is not only time consuming and expensive, but also prone to errors in the practice, since it is difficult to guarantee that no anomalies are buried inside the collected data. In addition, it is not easy to keep an accurate and up-to-date normal-operation profile.

Our thesis is that these two knowledge-based approaches are not sufficient to tackle the anomaly detection problem, and that a holistic solution should also include knowledge-independent analysis techniques. To this aim we propose UNADA, an Unsupervised Network Anomaly Detection Algorithm that detects network traffic anomalies without relying on signatures, training, or labeled traffic of any kind. Based on the observation that network traffic anomalies are, by definition, sparse events that deviate markedly from the majority of the traffic, UNADA relies on robust clustering algorithms to detect *outlying* traffic flows.

UNADA runs in three consecutive steps, analyzing packets captured in contiguous time slots of fixed length. Figure 1 depicts a modular, high-level description of UNADA. The first step consists in detecting an anomalous time slot in which the clustering analysis will be performed. For doing so, captured packets are first aggregated into multi-resolution traffic flows. Different time-series are then built on top of these flows, and any generic change-detection algorithm based on time-series analysis is finally used to flag an anomalous change. The second step takes as input all the flows in the time slot flagged as anomalous. In this step, outlying flows are identified using a robust multi-clustering algorithm, based on a combination of Sub-Space Clustering (SSC) [11], Density-based Clustering [17], and Evidence Accumulation Clustering (EAC) [15] techniques. The evidence of traffic structure provided by this clustering algorithm is used to rank the degree of *abnormality* of all the identified outlying flows, building an *outliers ranking*. In the third and final step, the top-ranked outlying flows are flagged as anomalies, using a simple thresholding detection approach. As we will show through out the paper, the main contribution provided by UNADA relies on its ability to work in a completely unsupervised fashion, outperforming previous proposals for unsupervised anomaly detection.

The remainder of the paper is organized as follows. Section 2 presents a short state of the art in the unsupervised anomaly detection field, additionally describing our main contributions. Section 3 describes the multi-resolution traffic aggregation and change-detection procedures used in the first step of UNADA. In section 4 we introduce the core of the proposal, presenting an in depth description of the different clustering techniques used by UNADA to construct the outliers ranking. Section 5 evaluates the ability of UNADA to discover single-source, single-destination, and distributed network anomalies in real network traffic from two different datasets: the public MAWI traffic repository of the WIDE project [21], and the METROSEC project dataset [22]. In this section we also compare the performance of UNADA against previous proposals for unsupervised anomaly detection. Finally, section 6 concludes this paper.

2 Related Work & Contributions

The problem of network anomaly detection has been extensively studied during the last decade. Most of the approaches analyze statistical variations of traffic volume descriptors (e.g., number of packets, bytes, or new flows) and/or particular traffic features (e.g., distribution of IP addresses and ports), using either single-link measurements or network-wide data. A non-exhaustive list of standard methods includes the use of signal processing techniques (e.g., ARIMA modeling, wavelets-based filtering) on single-link traffic measurements [1, 2], Kalman filters [5] for network-wide anomaly detection, and Sketches applied to IP-flows [6–8].

Our approach falls within the unsupervised anomaly detection domain. The vast majority of the unsupervised detection schemes proposed in the literature are based on clustering and outliers detection, being [12–14] some relevant ex-

amples. In [12], authors use a single-linkage hierarchical clustering method to cluster data from the KDD'99 data-set, based on the standard Euclidean distance for inter-patterns similarity. [13] reports improved results in the same data-set, using three different clustering algorithms: Fixed-Width clustering, an optimized version of k-NN, and one class SVM. [14] presents a combined density-grid-based clustering algorithm to improve computational complexity, obtaining similar detection results. PCA and the sub-space approach is another well-known unsupervised anomaly detection technique, used in [3,4] to detect network-wide traffic anomalies in highly aggregated traffic flows.

UNADA presents several advantages w.r.t. current state of the art. First and most important, it works in a completely unsupervised fashion, which means that it can be directly plugged-in to any monitoring system and start to work from scratch, without any kind of calibration and/or training step. Secondly, it uses a robust density-based clustering technique to avoid general clustering problems such as sensitivity to initialization, specification of number of clusters, detection of particular cluster shapes, or structure-masking by irrelevant features. Thirdly, it performs clustering in very-low-dimensional spaces, avoiding sparsity problems when working with high-dimensional data [9]. Finally, we show that UNADA clearly outperforms previously proposed methods for unsupervised anomaly detection in real network traffic.

3 Multi-resolution Flow Aggregation & Change-Detection

UNADA performs unsupervised anomaly detection on single-link packet-level traffic, captured in consecutive time slots of fixed length ΔT and aggregated in IP flows (standard *5-tuples*). IP flows are additionally aggregated at different *flow-resolution* levels, using 9 different *aggregation keys* l_i. These include (from coarser to finer-grained resolution): *traffic per Time Slot* (l_1:tpTS), *source Network Prefixes* ($l_{2,3,4}$: IPsrc/8, /16, /24), *destination Network Prefixes* ($l_{5,6,7}$: IPdst/8, /16, /24), *source IPs* (l_8: IPsrc), and *destination IPs* (l_9: IPdst). The 7 coarsest-grained resolutions are used for change-detection, while the remaining 2 are exclusively used in the clustering step.

To detect an anomalous time slot, time-series $Z_t^{l_i}$ are constructed for simple traffic metrics such as number of bytes, packets, and IP flows per time slot, using aggregation keys $i = 1, \ldots, 7$. Any generic change-detection algorithm $\mathcal{F}(.)$ based on time-series analysis is then used on $Z_t^{l_i}$. At each new time slot, $\mathcal{F}(.)$ analyses the different time-series associated with each aggregation key, going from coarser (l_1) to finer resolution (l_7). Time slot t_0 is flagged as anomalous if $\mathcal{F}(Z_{t_0}^{l_i})$ triggers an alarm for any of the traffic metrics at any of the 7 aggregation levels. Tracking anomalies at multiple aggregation levels provides additional reliability to the change-detection algorithm, and permits to detect both single source-destination and distributed anomalies of very different intensities.

Figure 2 shows how a low intensity DDoS attack might be dwarfed by highly-aggregated traffic. The time-series associated with the number of packets, namely $Z_t = \#\text{pkts}_t$, does not present a perceptible change $\Delta(\#\text{pkts}_t)$ at tpTS aggre-

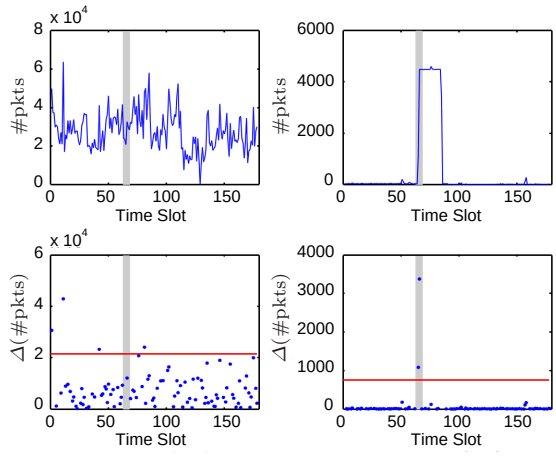


Fig. 2. Low-intensity anomalies might be hidden inside highly aggregated traffic, but are visible at finer-grained aggregations. The DDoS attack is evident at the victim's network.

gation (left); however, the attack can be easily detected using a finer-grained resolution, e.g., at the victim's network (IPdst/24 aggregation, on the right).

4 Unsupervised Anomaly Detection through Clustering

The unsupervised anomaly detection step takes as input all the IP flows in the flagged time slot. At this step UNADA ranks the degree of abnormality of each of these flows, using clustering and outliers analysis techniques. For doing so, IP flows are analyzed at two different resolutions, using either IPsrc or IPdst aggregation key. Traffic anomalies can be roughly grouped in two different classes, depending on their spatial structure and number of impacted IP flows: *1-to-N* anomalies and *N-to-1* anomalies. *1-to-N* anomalies involve many IP flows from the same source towards different destinations; examples include network scans and spreading worms/virus. On the other hand, *N-to-1* anomalies involve IP flows from different sources towards a single destination; examples include DDoS attacks and flash-crowds. *1-to-1* anomalies are a particular case of these classes, while *N-to-N* anomalies can be treated as multiple *N-to-1* or *1-to-N* instances. Using IPsrc key permits to highlight *1-to-N* anomalies, while *N-to-1* anomalies are more easily detected with IPdst key. The choice of both keys for clustering analysis ensures that even highly distributed anomalies, which may possibly involve a large number of IP flows, can be represented as outliers.

Without loss of generality, let $\mathbf{Y} = \{\mathbf{y}_1, .., \mathbf{y}_n\}$ be the set of n aggregated-flows (at IPsrc or IPdst) in the flagged slot. Each flow $\mathbf{y}_i \in \mathbf{Y}$ is described by a set of m traffic attributes or *features*, like number of sources, destination ports, or packet rate. Let $\mathbf{x}_i \in \mathbb{R}^m$ be the corresponding vector of traffic features describing flow $\mathbf{y}_i$, and $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_n\}$ the complete matrix of features, referred to as the *feature space*.

UNADA is based on clustering techniques applied to $\mathbf{X}$. The objective of clustering is to partition a set of unlabeled samples into homogeneous groups of similar characteristics or *clusters*, based on some measure of similarity. Samples that do not belong to any of these clusters are classified as *outliers*. Our particular goal is to identify those outliers that are remarkably different from the rest of the samples, additionally ranking how much different these are. The most appropriate approach to find outliers is, ironically, to properly identify clusters. After all, an outlier is a sample that does not belong to any cluster. Unfortunately, even if hundreds of clustering algorithms exist [9], it is very difficult to find a single one that can handle all types of cluster shapes and sizes. Different clustering algorithms produce different partitions of data, and even the same clustering algorithm provides different results when using different initializations and/or different algorithm parameters. This is in fact one of the major drawbacks in current cluster analysis techniques: the lack of robustness.

To avoid such a limitation, we have developed a divide & conquer clustering approach, using the notions of clustering ensemble [16] and multiple clusterings combination. The idea is novel and appealing: why not taking advantage of the information provided by multiple partitions of $\mathbf{X}$ to improve clustering robustness and identification of outliers? A clustering ensemble $\mathbf{P}$ consists of a set of multiple partitions P_i produced for the same data. Each of these partitions provides a different and independent evidence of data structure, which can be combined to construct a new measure of similarity that better reflects natural groupings and outliers. There are different ways to produce a clustering ensemble. We use Sub-Space Clustering (SSC) [11] to produce multiple data partitions, doing Density-based clustering in N different sub-spaces $\mathbf{X}_i$ of the original space. We refer the reader once again to figure 1 to better understand this approach.

4.1 Clustering Ensemble and Sub-Space Clustering

Each of the N sub-spaces $\mathbf{X}_i \subset \mathbf{X}$ is obtained by selecting k features from the complete set of m attributes. To deeply explore the complete feature space, the number of sub-spaces N that are analyzed corresponds to the number of k-combinations-obtained-from-m. Each partition P_i is obtained by applying DBSCAN [17] to sub-space $\mathbf{X}_i$. DBSCAN is a powerful clustering algorithm that discovers clusters of arbitrary shapes and sizes [9], relying on a density-based notion of clusters: clusters are high-density regions of the space, separated by low-density areas. This algorithm perfectly fits our unsupervised traffic analysis, because it is not necessary to specify a-priori difficult to set parameters such as the number of clusters to identify. Results provided by applying DBSCAN to sub-space $\mathbf{X}_i$ are twofold: a set of $p(i)$ clusters $\{C_1^i, C_2^i, .., C_{p(i)}^i\}$ and a set of $q(i)$ outliers $\{o_1^i, o_2^i, .., o_{q(i)}^i\}$. To set the number of dimensions k of each sub-space, we take a very useful property of monotonicity in clustering sets, known as the downward closure property: "if a collection of elements is a cluster in a k-dimensional space, then it is also part of a cluster in any $(k-1)$ projections of this space" [18]. This directly implies that, if there exists any interesting

Algorithm 1 Evidence Accumulation for Ranking Outliers (EA4RO)

Initialization:
  Set dissimilarity vector D to a null $n \times 1$ vector
  Set smallest cluster-size $n_{\min} = \alpha \,.\, n$
for $i = 1 : N$ **do**
  Set density neighborhood δ_i for DBSCAN
  P_i = DBSCAN $(\mathbf{X}_i, \delta_i, n_{\min})$
  Update $D(j)$, $\forall$ outlier $o_j^i \in P_i$:
    $w_i \leftarrow \dfrac{n}{(n - n_{\max_i}) + \epsilon}$
    $D(j) \leftarrow D(j) + \mathrm{d_M}(o_j^i, C_{\max}^i)\, w_i$
end for
Rank flows: $D_{rank} = \mathrm{sort}(D)$
Set anomaly detection threshold: T_h = find-slope-break(D_{rank})

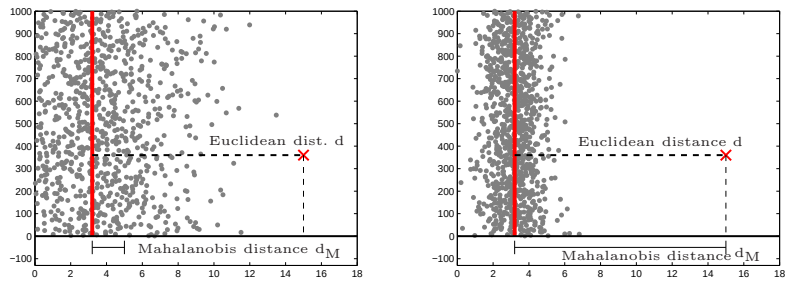
evidence of density in $\mathbf{X}$, it will certainly be present in its lowest-dimensional sub-spaces. Using small values for k provides several advantages: firstly, doing clustering in low-dimensional spaces is more efficient and faster than clustering in bigger dimensions. Secondly, density-based clustering algorithms such as DBSCAN provide better results in low-dimensional spaces [18], because high-dimensional spaces are usually sparse, making it difficult to distinguish between high and low density regions. Finally, clustering multiple low-dimensional sub-spaces provides a finer-grained analysis, which improves the ability of UNADA to detect anomalies of very different characteristics. We shall therefore use $k = 2$ for SSC, which gives $N = m(m-1)/2$ partitions.

4.2 Ranking Outliers using Evidence Accumulation

Having produced the N partitions, the question now is how to use the information provided by the multiple clusters and outliers identified by density-based clustering. A possible answer is provided in [15], where authors introduced the idea of Evidence Accumulation Clustering (EAC). EAC uses the clustering results of multiple partitions P_i to produce a new inter-samples similarity measure that better reflects their natural groupings.

UNADA implements a particular algorithm for Evidence Accumulation, called Evidence Accumulation for Ranking Outliers (EA4RO): instead of producing a similarity measure between the n different aggregated flows described in $\mathbf{X}$, EA4RO constructs a dissimilarity vector $D \in \mathbb{R}^n$ in which it accumulates the distance between the different outliers o_j^i found in each sub-space $i = 1, .., N$ and the centroid of the corresponding sub-space-biggest-cluster $C_{\max}^i$. The idea is to clearly highlight those flows that are far from the normal-operation traffic at each of the different sub-spaces, statistically represented by $C_{\max}^i$.

Algorithm 1 presents a pseudo-code for EA4RO. The different parameters used by EA4RO are automatically set by the algorithm itself. The first two parameters are used by the density-based clustering algorithm: $n_{\min}$ specifies the minimum number of flows that can be classified as a cluster, while δ_i indicates



(a) (b)

Fig. 3. Euclidean vs. Mahalanobis distance as a measure of dissimilarity.

the neighborhood distance of a sample to identify dense regions. $n_{\min}$ is set at the initialization of the algorithm, simply as a fraction α of the total number of flows n to analyze (we take $\alpha = 5\%$ of n). δ_i is set as a fraction of the average distance between flows in sub-space $\mathbf{X}_i$ (we take a fraction 1/10), which is estimated from 10% of the flows, randomly selected. This permits to fast-up computations. The weighting factor w_i is used as an outlier-boosting parameter, as it gives more relevance to those outliers that are "less probable": w_i takes bigger values when the size $n_{\max_i}$ of cluster $C^i_{\max}$ is closer to the total number of flows n. Finally, instead of using a simple Euclidean distance as a measure of dissimilarity, we compute the Mahalanobis distance $\mathrm{d_M}$ between outliers and the centroid of the biggest cluster. The Mahalanobis distance takes into account the correlation between samples, dividing the standard Euclidean distance by the variance of the samples. This permits to boost the degree of abnormality of an outlier when the variance of the samples is smaller. Figure 3 permits to evidence the advantage of using Mahalanobis. The Euclidean distance between the centroid of the cluster and the outlier is the same in both figures 3.(a) and 3.(b). However, variance in figure 3.(a) is bigger than in figure 3.(b), which intuitively should reduce the abnormality degree of the outlier. Such a notion is correctly captured by the Mahalanobis distance, which is smaller in figure 3.(a).

In the last part of EA4RO, flows are ranked according to the dissimilarity obtained in D, and the anomaly detection threshold T_h is set. The computation of T_h is simply achieved by finding the value for which the slope of the sorted dissimilarity values in D_{rank} presents a major change. In the evaluation section we explain how to perform this computation with an example of real traffic analysis. Anomaly detection is finally done as a binary thresholding operation on D: if $D(i) > T_h$, UNADA flags an anomaly in flow $\mathbf{y}_i$.

5 Experimental Evaluation of UNADA

We evaluate the ability of UNADA to detect different attacks in real traffic traces from the public MAWI repository of the WIDE project [21]. The WIDE operational network provides interconnection between different research institutions in

Table 1. Features used by UNADA in the detection of DoS, DDoS, network/port scans, and spreading worms. For each type of attack, we describe its impact on the selected traffic features.

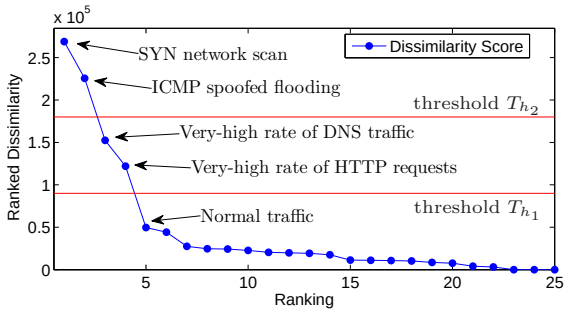
Type of Attack	Class	Agg-Key	Impact on Traffic Features
DoS (ICMP/SYN)	1-to-1	IPdst	nSrcs = nDsts = 1, nPkts/sec $> \lambda_1$, avgPktsSize $< \lambda_2$, nICMP/nPkts $> \lambda_3$, nSYN/nPkts $> \lambda_4$.
DDoS (ICMP/SYN)	N-to-1	IPdst	nDsts = 1, nSrcs $> \alpha_1$, nPkts/sec $> \alpha_2$, avgPktsSize $< \alpha_3$, nICMP/nPkts $> \alpha_4$, nSYN/nPkts $> \alpha_5$.
Port scan	1-to-1	IPsrc	nSrcs = nDsts = 1, nDstPorts $> \beta_1$, avgPktsSize $< \beta_2$, nSYN/nPkts $> \beta_3$.
Network scan	1-to-N	IPsrc	nSrcs = 1, nDsts $> \delta_1$, nDstPorts $> \delta_2$, avgPktsSize $< \delta_3$, nSYN/nPkts $> \delta_4$.
Spreading worms	1-to-N	IPsrc	nSrcs = 1, nDsts $> \eta_1$, nDstPorts $< \eta_2$, avgPktsSize $< \eta_3$, nSYN/nPkts $> \eta_4$.

Japan, as well as connection to different commercial ISPs and universities in the U.S.. The traffic repository consists of 15 minutes-long raw packet traces daily collected for the last ten years. The traces we shall work with consist of traffic from one of the trans-pacific links between Japan and the U.S.. MAWI traces are not labeled, but some previous work on anomaly detection has been done on them [8, 20]. In particular, [20] detects network attacks using a signature-based approach, while [8] detects both attacks and anomalous flows using non-Gaussian modeling. We shall therefore refer to the combination of results obtained in both works as our *ground truth* for MAWI traffic.

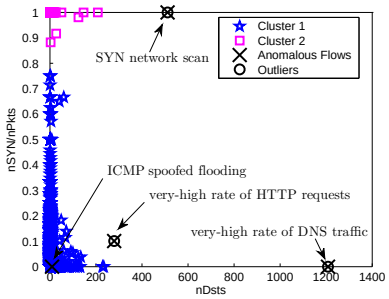
We shall also test the true positive and false positive rates obtained with UNADA in the detection of flooding attacks in traffic traces from the METROSEC project [22]. These traces consist of real traffic collected on the French RENATER network, containing simulated attacks performed with well-known DDoS attack tools. Traces were collected between 2004 and 2006, and contain DDoS attacks that range from very low intensity (i.e., less than 4% of the overall traffic volume) to massive attacks (i.e., more than 80% of the overall traffic volume). In addition, we compare the performance of UNADA against some previous methods for unsupervised anomaly detection presented in section 2.

5.1 Features Selection for Detection of Attacks

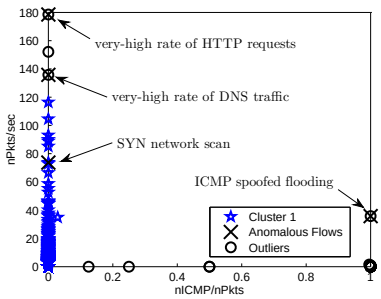
The selection of the m features used in $\mathbf{X}$ to describe the aggregated flows in $\mathbf{Y}$ is a key issue to any anomaly detection algorithm, but it becomes critical and challenging in the case of unsupervised detection, because there is no additional information to select the most relevant set. In general terms, using different traffic features permits to detect different types of anomalies. In this paper we shall limit our study to detect well-known attacks, using a set of standard traffic features widely used in the literature. However, the reader should note that UNADA can be extended to detect other types of anomalies, considering different sets of traffic features. In fact, more features can be added to any standard list to improve detection results. For example, we could use the set of traffic features generally used in the traffic classification domain [19] for our problem of anomaly detection, as this set is generally broader; if these features are good enough to classify different traffic applications, they should be useful to perform anomaly detection. The main advantage of UNADA is that we have devised an algorithm



(a) Detecting network attacks using dissimilarity vector D.



(b) Scan and flooding attacks (1/2)



(c) Scan and flooding attacks (2/2)

Fig. 4. Detection and analysis of network attacks in MAWI.

to highlight outliers respect to any set of features, and this is why we claim that our algorithm is highly applicable.

In this paper we shall use the following list of $m = 9$ traffic features: number of source/destination IP addresses and ports (nSrcs, nDsts, nSrcPorts, nDstPorts), ratio of number of sources to number of destinations, packet rate (nPkts/sec), fraction of ICMP and SYN packets (nICMP/nPkts, nSYN/nPkts), and average packet size (avgPktsSize). According to previous work on signature-based anomaly characterization [20], such simple traffic descriptors permit to describe standard network attacks such as DoS, DDoS, scans, and spreading worms/virus.

Table 1 describes the impacts of different types of attacks on the selected traffic features. All the thresholds used in the description are introduced to better explain the evidence of an attack in some of these features. DoS/DDoS attacks are characterized by many small packets sent from one or more source IPs towards a single destination IP. These attacks generally use particular packets such as TCP SYN or ICMP echo-reply. echo-request, or host-unreachable packets. Port and network scans involve small packets from one source IP to several ports in one or more destination IPs, and are usually performed with SYN packets. Spreading worms differ from network scans in that they are directed towards a small specific group of ports for which there is a known vulnerability to exploit (e.g. Blaster on TCP port 135, Slammer on UDP port 1434, Sasser on TCP port 455), and they generally use slightly bigger packets. Some of these attacks can

use other types of traffic, such as FIN, PUSH, URG TCP packets or small UDP datagrams.

5.2 Detecting Attacks in MAWI traffic

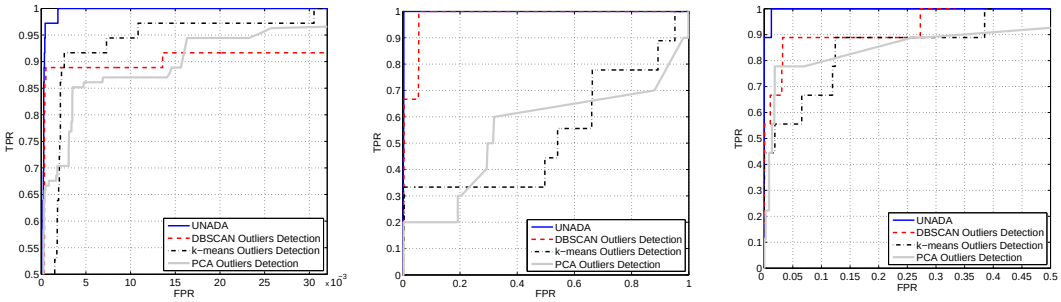
We shall begin by analyzing the performance of UNADA to detect network attacks and other types of anomalies in one of the traces previously analyzed in [8]. IP flows are aggregated according to IPsrc key. Figure 4.(a) shows the ordered dissimilarity values in D obtained by the EA4RO method, along with their corresponding manual classification. The first two most dissimilar flows correspond to a highly distributed SYN network scan (more than 500 destination hosts) and an ICMP spoofed flooding attack directed to a small number of victims (ICMP redirect traffic, directed towards port 0). The following two flows correspond to unusual large rates of DNS traffic and HTTP requests; from there on, flows correspond to normal-operation traffic. The ICMP flooding attack and the two unusual flows are also detected in [8]; the SYN scan was missed by their method, but it was correctly detected with accurate signatures [20].

Setting the detection threshold according to the previously discussed approach results in T_{h_1}. Indeed, if we focus on the shape of the ranked dissimilarity in figure 4.(a), we can clearly appreciate a major change in the slope after the $5th$ ranked flow. Note however that both attacks can be easily detected and isolated from the anomalous but yet legitimate traffic without false alarms, using for example the threshold T_{h_2} on D.

Figures 4.(b,c) depict the corresponding four flows in two of the N partitions produced by the EA4RO method. Besides showing typical characteristics of the attacks, such as a large value of nPkts/sec or a value 1 for attributes nICMP/nPkts and nSYN/nPkts respectively, both figures permit to appreciate that the detected attacks do not necessarily represent the largest elephant flows in the time slot. This emphasizes the ability of UNADA to detect attacks of low intensity, event lower than normal traffic.

5.3 Detecting Attacks with Ground Truth

Figure 5 depicts the True Positives Rate (TPR) as a function of the False Positives Rates (FTR) in the detection of different attacks in MAWI and METROSEC. Figure 5.(a) corresponds to the detection of 36 anomalies in MAWI traffic, using IPsrc as key. These anomalies include network and port scans, worm scanning activities (Sasser and Dabber variants), and some anomalous flows consisting on very high volumes of NNTP traffic. Figure 5.(b) also corresponds to anomalies in MAWI traffic, but using IPdst as key. In this case, there are 9 anomalies, including different kinds of flooding DoS/DDoS attacks. Finally, figure 5.(c) corresponds to the detection of 9 DDoS attacks in the METROSEC data-set. From these, 5 correspond to massive attacks (more than 70% of traffic), 1 to a high intensity attack (about 40%), 2 are low intensity attacks (about 10%), and 1 is a very-low intensity attack (about 4%). The detection is performed using traffic aggregated with IPdst key. In the three evaluation



(a) MAWI, IPsrc key. (b) MAWI, IPdst key. (c) METROSEC, IPdst key.

Fig. 5. True Positives Rate vs False Alarms in MAWI and METROSEC.

scenarios, the ROC plot is obtained by comparing the sorted dissimilarities in D_{rank} to a variable detection threshold.

We compare the performance of UNADA against three previous approaches for unsupervised anomaly detection: DBSCAN-based, k-means-based, and PCA-based outliers detection. The first two consist in applying either DBSCAN or k-means to the complete feature space $\mathbf{X}$, identify the largest cluster $C_{\max}$, and compute the Mahalanobis distance of all the flows lying outside $C_{\max}$ to its centroid. The ROC is finally obtained by comparing the sorted distances to a variable detection threshold. These approaches are similar to those used in previous work [12–14]. In the PCA-based approach, PCA and the sub-space methods [3,4] are applied to the complete matrix $\mathbf{X}$, and the attacks are detected by comparing the residuals to a variable threshold. Both the k-means and the PCA-based approaches require fine tuning: in k-means, we repeat the clustering for different values of clusters k, and take the average results. In the case of PCA we present the best performance obtained for each evaluation scenario.

Obtained results permit to evidence the great advantage of using the SSC-Density-based algorithm in the clustering step w.r.t. to previous approaches. In particular, all the approaches used in the comparison generally fail to detect all the attacks with a reasonable false alarm rate. Both the DBSCAN-based and the k-means-based algorithms get confused by masking features when analyzing the complete feature space $\mathbf{X}$. The PCA approach shows to be not sensitive enough to discriminate different kinds of attacks of very different intensities, using the same representation for normal-operation traffic.

6 Conclusions

The Unsupervised Network Anomaly Detection Algorithm that we have proposed presents many interesting advantages w.r.t. previous proposals in the field of unsupervised anomaly detection. It uses exclusively unlabeled data to detect traffic anomalies, without assuming any particular model or any canonical data distribution, and without using signatures of anomalies or training. Despite using ordinary clustering techniques to identify traffic anomalies, UNADA avoids

the lack of robustness of general clustering approaches, by combining the notions of Sub-Space Clustering, Density-based Clustering, and multiple Evidence Accumulation.

We have verified the effectiveness of UNADA to detect real single source-destination and distributed network attacks in real traffic traces from different networks, all in a completely blind fashion, without assuming any particular traffic model, clustering parameters, or even clusters structure beyond a basic definition of what an anomaly is. Additionally, we have shown detection results that outperform traditional approaches for outliers detection, providing a stronger evidence of the accuracy of UNADA to detect network anomalies.

Acknowledgments

This work has been done in the framework of the ECODE project, funded by the European commission under grant FP7-ICT-2007-2/223936.

References

1. P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies", in *Proc. ACM IMW*, 2002.
2. J. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring", in *Proc. 14th Systems Administration Conference*, 2000.
3. A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies", in *Proc. ACM SIGCOMM*, 2004.
4. A. Lakhina, M. Crovella and C. Diot, "Mining Anomalies Using Traffic Feature Distributions", in *Proc. ACM SIGCOMM*, 2005.
5. A. Soule et al., "Combining Filtering and Statistical Methods for Anomaly Detection", in in *Proc. ACM IMC*, 2005.
6. B. Krishnamurthy et al., "Sketch-based Change Detection: Methods, Evaluation, and Applications", in *Proc. ACM IMC*, 2003.
7. X. Li et al., "Detection and Identification of Network Anomalies Using Sketch Subspaces", in *Proc. ACM IMC*, 2006.
8. G. Dewaele et al., "Extracting Hidden Anomalies using Sketch and non Gaussian Multi-resolution Statistical Detection Procedures", in *Proc. SIGCOMM LSAD*, 2007.
9. A. K. Jain, "Data Clustering: 50 Years Beyond K-Means", in *Pattern Recognition Letters*, vol. 31 (8), pp. 651-666, 2010.
10. L. Kaufman and P. J. Rosseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, 1990.
11. L. Parsons et al., "Subspace Clustering for High Dimensional Data: a Review", in *ACM SIGKDD Expl. Newsletter*, vol. 6 (1), pp. 90-105, 2004.
12. L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering", in *Proc. ACM DMSA Workshop*, 2001.
13. E. Eskin et al., "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data", in *Applications of Data Mining in Computer Security*, Kluwer Publisher, 2002.

14. K. Leung and C. Leckie, "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clustering", in *Proc. ACSC05*, 2005.
15. A. Fred and A. K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation", in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27 (6), pp. 835-850, 2005.
16. A. Strehl and J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework For Combining Multiple Partitions", in *Journal on Machine Learning Research*, vol. 3, pp. 583-617, 2002.
17. M. Ester et al., "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in *Proc. ACM SIGKDD*, 1996.
18. R. Agrawal et al., "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", in *Proc. ACM SIGMOD*, 1998.
19. N. Williams, S. Zander, and G. Armitage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification", in *ACM SIGCOMM Computer Communication Review*, vol. 36 (5), 2006.
20. G. Fernandes and P. Owezarski, "Automated Classification of Network Traffic Anomalies", in *Proc. SecureComm'09*, 2009.
21. K. Cho, K. Mitsuya and A. Kato, "Traffic Data Repository at the WIDE Project", in *USENIX Annual Technical Conference*, 2000.
22. "METROlogy for SECurity and QoS", at `http://laas.fr/METROSEC`