MAWILab : Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking

Romain Fontugne, Pierre Borgnat, Patrice Abry, Kensuke Fukuda

MAWILab : Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking

Romain Fontugne[1], Pierre Borgnat[2], Patrice Abry[2], and Kensuke Fukuda[1,3]
[1]The Graduate University for Advanced Studies, Tokyo, JP [2]CNRS, ENSL, Physics Lab., Lyon, FR
[3]National Institute of Informatics / PRESTO JST, Tokyo, JP

ABSTRACT

Evaluating anomaly detectors is a crucial task in traffic monitoring made particularly difficult due to the lack of ground truth. The goal of the present article is to assist researchers in the evaluation of detectors by providing them with labeled anomaly traffic traces. We aim at automatically finding anomalies in the MAWI archive using a new methodology that combines different and independent detectors. A key challenge is to compare the alarms raised by these detectors, though they operate at different traffic granularities. The main contribution is to propose a reliable graph-based methodology that combines any anomaly detector outputs. We evaluated four unsupervised combination strategies; the best is the one that is based on dimensionality reduction. The synergy between anomaly detectors permits to detect twice as many anomalies as the most accurate detector, and to reject numerous false positive alarms reported by the detectors. Significant anomalous traffic features are extracted from reported alarms, hence the labels assigned to the MAWI archive are concise. The results on the MAWI traffic are publicly available and updated daily. Also, this approach permits to include the results of upcoming anomaly detectors so as to improve over time the quality and variety of labels.

1. INTRODUCTION

Anomalies in Internet traffic penalize legitimate users from accessing optimal network resources. Identifying anomalous events is a crucial network management task that requires automation. Consequently, anomaly detection has received a lot of attention in the last decade, and numerous detectors have been proposed. Operators, however, often disregard the alarms reported by anomaly detectors because of several drawbacks discrediting them [17, 30]. The key task for improving anomaly detectors is to thoroughly inspect results generated by such detectors and precisely identify any drawbacks. However, identifying anomaly detectors vulnerabilities is particularly difficult due to a lack of ground truth data and of rigorous methodology. Hence, anomaly detectors are evaluated by using distinct methodologies analyzing traffic with real or simulated anomalies.

With real anomalies, researchers evaluate anomaly detectors by manually checking the reported alarms [8, 11, 22, 23], or by comparing them to those reported by other anomaly detectors [14, 21, 22, 23]. Sometimes researchers construct ground truth data by manually inspecting the analyzed traffic [4]. However, these evaluations are hardly comparable, trustworthy, or reproducible, as they require significant human intervention and as traffic traces are usually inaccessible due to privacy issues. Also, a common shortcoming of these evaluation methodologies is the omission of the false negative rate of the detector, in spite of the fact that this metric is the good indicator of the number of missed anomalies and of the sensitivity of the detector to different kinds of anomalies.

Simulating anomalies is also a common way to evaluate an anomaly detector [21, 27, 31, 32]. In this case, the parameters of anomalies are tunable (e.g., in intensity and time duration), helping researchers to measure the sensitivity of their detectors to particular kinds of anomalies. However, simulating traffic as diverse as it is on the Internet is notoriously difficult [12], especially for anomalous traffic. Consequently, the evaluation of a detector with simulated anomalies is restricted to certain kinds of anomaly, and thus, is insufficient for measuring the detector performance [29].

Ideally, an anomaly detector has to be evaluated using ground truth data containing real and nonspecific traffic where there is a wide range of anomalies. This ground truth data should be publicly available to allow all researchers to access the same data set and compare their results. Furthermore, the data set should follow the evolution of the Internet traffic to include traffic from emerging applications and anomalies. Currently, there is no such crucial data with ground truth; providing such data is our objective.

The goal is to find and label anomalies in the traffic from the MAWI archive [9], and to make it available to researchers so that they can refer to it while evaluating their own anomaly

ACM CoNEXT 2010, November 30 – December 3 2010, Philadelphia, USA.

detection methods. The main advantages of the MAWI archive are that it is updated daily and it currently contains more than nine years of real publicly available Internet traffic data. However, manually labeling anomalies in such a large data set is certainly impractical, and therefore, the challenge we face is to accurately find anomalies in an automated and unsupervised manner. The numerous anomaly detectors that have recently been proposed in literature are the main support that will help us to reach the goal. Therefore, we are selecting diverse anomaly detectors and combining their results to accurately find anomalies in the MAWI archive. The synergy between detectors with different theoretical backgrounds allows a more accurate level of detection to be achieved. However, a key issue in combining such diverse detectors is that they report different granularities of the traffic that are difficult to rigorously compare.

The contribution of the present article is twofold. Firstly, we establish a reliable methodology, which is based on graph and community mining, that compares and combines the results from any anomaly detectors, even though they operate at different traffic granularities. The proposed method outperforms the combined detectors, and enables us to precisely find twice as many anomalies as the most accurate detector from the experiments. Secondly, results are made available in the form of labeled data set, providing a benchmark for anomaly detection methods. The database currently stands for more than nine years of traffic and it is growing along with the MAWI archive. Furthermore, this approach permits the enhancement of the database over time by integrating the results from emerging anomaly detectors. Thus, the proposed database is constantly updated with new traffic and anomaly detectors, and it is a valuable tool to assist researchers designing anomaly detectors.

Related work. Providing ground truth data to evaluate anomaly detectors is a challenge that has been addressed several times in the past. For example, the DARPA Intrusion Detection Evaluation Program [24] has been a great effort to provide labeled traffic to evaluate intrusion detection systems (IDS). It has been extensively studied, mainly through the KDD Cup 1999 data (KDD'99), and has been a profitable support for researchers. The main distinctions between this work and ours are the size of the network measured and the detectors to be evaluated. The DARPA Intrusion Detection Evaluation Program focuses on the evaluation of IDS and provides labeled LAN traffic where the packet payload is available and flows are complete. Whereas our work focuses on the evaluation of backbone traffic anomaly detectors and we provide labeled backbone traffic where the packet payload is not available, and the flows are incomplete and asymmetric. Furthermore, several critical drawbacks of the KDD'99 have been reported [25]. Also, the traffic data was captured in 1998, hence it contains no traffic from recent applications or anomalies. Therefore, this data must be carefully used as it is not representative of real traffic [34] and does not contain recent anomalies.

Closer to our work, Owezarski [28] recently proposed a data set containing real backbone traffic where anomalies are precisely located. In this work the traffic is captured at different points in the RENATER network, which is supposed to be anomaly free, and the researchers generate two kinds of anomalies (i.e., flash crowd and DDoS attack). Their experiment consist of different scenarios where the intensity of the anomalies varies. Thus, the sensitivity of the detectors to DDoS and flash crowd is easily identified. However, there are only a few kinds of anomalies in their data and they are not a realistic representation of the diverse anomalies found on the Internet. Due to privacy issues, their data is not downloadable and only accessible by visiting their laboratory.

Being conscious of the shortcomings of previous works, the data set is chosen and design to overcome these issues.

The proposed approach takes advantage of combination strategies in order to merge the results from several detectors. Although the combination of classifiers is a hot topic in the clustering community [20], only a few works have been conducted in the field of network anomaly detection. For example, Shanbhag and Wolf [33] have studied the combination of five rate-based detectors to accurately identify the real-time variance in traffic volume. They analyzed seven different combination strategies and emphasize that the best strategy improves the accuracy of the overall detectors. The goal here differs from theirs as they aim at detecting anomalies in real time by running several detectors in parallel. Thus, they restrict their study to a particular kind of computationally efficient anomaly detector (rate-based detector), whereas our approach takes advantage of diverse anomaly detectors.

Another recent study on the combination of anomaly detectors was conducted by Ashfaq et al. [3]. They propose a new combination strategy that takes into account the accuracy of the detectors; first, the accuracy of each detector is evaluated on a training data set, and then, the results of the detectors are combined regarding their accuracy. Their results emphasized the benefit of taking into account the detectors accuracies when combining them. Nevertheless, we avoid such methods as they involve a training step that increases the necessity of human intervention. The proposed approach focuses on unsupervised anomaly detectors that are combined with unsupervised combination strategies.

Proposed method. The method consists of four main steps, executed for each traffic trace: (1) Several anomaly detectors analyze the traffic and report alarms. (2) The similarities between the reported alarms are uncovered using a **similarity estimator** that groups similar alarms into communities. (3) Each community is investigated and classified by the **combiner**. Namely, the combiner decides if the community has to be reported as anomalous, or ignored, depending on the overall outputs of the detectors. (4) The anomalies are characterized using association rule mining on the combiner results so as to label anomalies in the analyzed data set.

The paper is organized as follows: Steps 2 and 3 are detailed in Section 2, and evaluated in Section 4. For that, the data set and anomaly detectors that are used are depicted in Section 3. Step 4 is described in Section 5. The results are further discussed in Section 6 and we conclude in Section 7.

2. METHODOLOGY

2.1 Similarity estimator

Since the benefit of combining detectors relies on the diversity among the detectors ensemble, we combine various anomaly detectors based on different theoretical backgrounds. Nevertheless, these different anomaly detectors are inherently reporting traffic at different granularities (e.g., flow, host, or packet) that are difficult to systematically compare. A detector might reports alarms at the host level, for example A_1 for IP_X, and another detector reports alarms at the flow level, for example B_1 and B_2 for $< IP_X, 80, IP_Y, 1234 >$ and $< IP_X, 80, IP_Z, 2345 >$. In that case, A_1 includes B_1 and B_2; however, telling that the three alarms are the same is hard because B_1 and B_2 are obviously reporting distinct traffic. Therefore, a rigorous method precisely measuring the similarities between alarms is required.

The similarity estimator presented in this section is an extension of a previous work [13]. Its role is to uncover the relations between the outputs of any kinds of anomaly detector. First, it reads the alarms reported by the detectors and the original traffic, and it extracts the traffic described by each alarm. Second, it constructs a graph that highlights the alarm similarities based on the traffic they have in common. Finally, similar alarms are identified by finding communities (i.e., dense connected components) in the graph.

2.1.1 *Traffic extractor*

The traffic extractor (called oracle in [13]) retrieves the traffic described by each alarm. Let an alarm be a set of traffic features that designates a particular traffic identified by a detector. The traffic extractor records the association between the alarm and this traffic. In [13], traffic associated with given alarms is always a set of packets, whereas the current work evaluates the benefits of associated traffic at different granularities: either packet, or flow (unidirectional or bidirectional). Figure 1 depicts the main differences in using flows and packets. The three alarms in Fig. 1 report three sets of packets from the same flow. By using packet as the traffic granularity, we observe that $Alarm2$ and $Alarm3$ have traffic in common but no packet is shared with $Alarm1$. Whereas using flow, the three alarms are reporting the same traffic and thus will have similatities.

2.1.2 *Graph generator*

The graph generator uses the traffic retrieved by the traffic extractor to build an undirected graph called similarity graph, highlighting the similarities among all the alarms reported by the detectors. In this graph, a **node** stands for an **alarm**, and there is an **edge** between two nodes if their associated traffic intersects. In addition, an **edge** is weighted with a similarity measure that quantifies the **traffic intersection** of the two alarms it connects. Therefore, the similarity measure enables to discriminate edges connecting dissimilar alarms having an irrelevant number of packets or flows in common. We selected three similarity measures for the experiments: the Jaccard index, the Simpson index and a constant function. Since the Simpson index outperformed the two other metrics, and due to page limitation, only the Simpson index is discussed in this article. The Simpson index is defined as

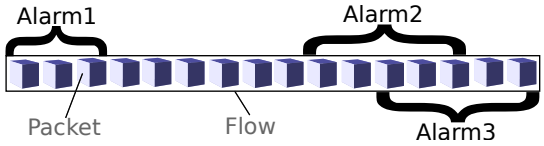


Figure 1: A flow composed of packets corresponding to three different alarms. *Alarm2* and *Alarm3* have common packets, whereas *Alarm1* consists of a distinct set of packets.

$$S(E_1, E_2) = |E_1 \cap E_2| / \min(|E_1|, |E_2|)$$

where E_i is the traffic associated with alarm i. This metric ranges [0,1], where 0 means that the two traffic do not intersect, and that the two alarms are fully dissimilar; 1 means that they are identical or that one is included in the other.

2.1.3 *Community mining*

The similarity graph describes the alarm similarities, however alarms that are identical are not yet determined. Identical alarms are characterized in the graph as being a set of strongly connected nodes: this is called a community. Identifying the communities in a graph is a problem that has been extensively studied in the past [15]. Although numerous community mining algorithms have been proposed, the interest here focuses on those designed for sparse graph since the generated graphs have disconnected nodes (e.g., a false positive alarm reported by one detector). In the experiments, we selected a method based on the modularity: the Louvain algorithm [6]. This algorithm has the advantage of locally identifying the communities, thus allowing us to identify groups of a few alarms. Furthermore, this algorithm performs a fast and accurate analysis of the graph [15].

2.2 Combiner

The similarity estimator clusters similar alarms into communities: each community represents a set of distinct alarms (i.e., nodes) reported by various detectors. The role of the proposed combiner is to decide whether each community corresponds to an anomalous traffic or not. For that, the combiner classifies the communities into two categories, $accepted$ and $rejected$, respectively standing for the communities reported as anomalous or those ignored. The class of a community is determined by a combination strategy, adapted from machine learning or pattern classifiers [20].

2.2.1 Background: combining detectors

A combination strategy is generally categorized as a detector selection or an output fusion. On the one hand, detector selection consists of selecting the detector that is the most suitable for classifying an element (i.e., a community in our case) and makes the same decisions as the single selected detector. Since each element is analyzed by only one detector, this approach is usually a good candidate for performing a quick analysis. However, selecting an appropriate detector is in practice challenging. In particular, the sensitivity of detectors in the context of network anomaly detection is misunderstood and prevents us from applying such techniques. On the other hand, output fusion makes no assumption on the detectors as it inspects the results of all the detectors. The output of a detector is assimilated to a vote for a certain class, and the combination strategy refers to a voting procedure.

In order to emphasize the advantages of combining detectors with output fusion let us review perhaps the oldest and best-known strategy, the majority vote. It is a basic, but still powerful way, where the final decision is the simple majority of the detectors outputs (i.e., more than 50 percent of the outputs). The probability of making the correct decision with the majority vote depends on the probability of each detector for providing the correct output, that is:

$$P_{maj}(L) = \sum_{m=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{m} p^m (1-p)^{L-m}$$

where L is the number of detectors and p is their accuracy. The result, also known as the Condorcet Jury Theorem, is as follows; if $p > 0.5$, then $P_{maj}(L)$ is monotonically increasing in L and $P_{maj}(L) \to 1$ as $L \to \infty$. If $p < 0.5$, then $P_{maj}(L)$ is monotonically decreasing in L and $P_{maj}(L) \to 0$ as $L \to \infty$. If $p = 0.5$, then $P_{maj}(L) = 0.5$ for any L. This theorem highlights the benefit of combining reasonable detectors (i.e., with an accuracy $p > 0.5$) over the use of a single detector.

2.2.2 Application to traffic anomaly detection

Each anomaly detector outputs a binary value telling if a traffic is anomalous or not. Namely, for each community in the similarity graph, a detector votes for it being anomalous if at least one of its alarms is in the community. Although this is sufficient to compute a majority vote, this binary value is too coarse for a precise combination. Also, the votes of the detectors may significantly vary, depending on the tuning of their parameters.

To prevent these difficulties we propose to score the confidence of each vote of the detectors. Hereafter we refer to a certain detector with a specific parameter set as a **configuration**. Running a detector with several parameter sets and measuring the variability of its output quantifies its parameter sensitivity. The outputs of all configurations are merged through the similarity estimator, and the variability in the outputs is computed by inspecting each community. The **confidence score** φ of a detector d for a community c is defined as:

$$\varphi_d(c) = \phi_d(c)/T_d$$

where T_d is the total number of configurations with the detector d and $\phi_d(c)$ is the number of these configurations that reports at least one alarm belonging to the community c. The confidence score is a continuous value that ranges [0,1], 0 representing that a given detector ignores the community whereas 1 means that all configurations of the detector identify the community. For example, Fig. 2 is a community c_{ex} composed of fives alarms. Assuming that the input of the similarity estimator, X_i, consists of the output of nine configurations corresponding to three detectors $X = A, B, C$ with three different parameter sets $i = 0, 1, 2$, then the confidence scores for this community are: $\varphi_A(c_{ex}) = 0.66$, $\varphi_B(c_{ex}) = 1.0$ and $\varphi_C(c_{ex}) = 0.0$.

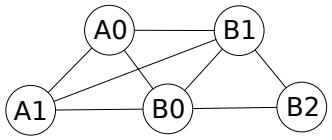


Figure 2: Example of community c_{ex} composed of five alarms. Assuming that the input of the similarity estimator, X_i, consists of the output of three detectors $X = A, B, C$ with three different parameter sets $i = 0, 1, 2$, then the confidence scores are: $\varphi_A(c_{ex}) = 0.66$, $\varphi_B(c_{ex}) = 1.0$ and $\varphi_C(c_{ex}) = 0.0$.

2.2.3 Combination strategies

Average, Minimum, & Maximum. Let us now present three different combination strategies that aggregate the confidence scores relative to a given community c in a value $\mu(c)$, and classify a community c as *accepted* (i.e., labeled anomalous) only if $\mu(c) > 0.5$.

Aggregating the confidences score of a community by *average* allows us to rely equally on the votes of all the detectors. Formally, for a community c and using L detectors, the average is: $\mu(c) = \frac{1}{L} \sum_{i=1}^{L} \varphi_i(c)$. In the example shown in Fig. 2 the average of all the confidence scores equals $5/9$, and thus, this combination strategy would classify the community c_{ex} as accepted.

Another strategy consists in selecting the *minimum* confidence score. This pessimistic decision classify a community as accepted only if all the detectors support this decision. Consequently, the ratio of false positive is substantially reduced at the cost of an increase in the ratio of true negative. Formally, the decision made for the community c depends on its minimum confidence score: $\mu(c) = \min_i\{\varphi_i(c)\}$. In the example shown in Fig. 2, the minimum of all the confidence scores is 0, and thus, this combination strategy would classify the community c_{ex} as rejected.

On the contrary, a third strategy is to select the $maximum$ confidence score. This optimistic decision classify a community as accepted only if at least one detector supports this decision. Consequently, the ratio of true positive is substantially increased, but so is the ratio of false positive. Formally, the decision made for the community c depends on its maximum confidence score: $\mu(c) = \max_i\{\varphi_i(c)\}$. In the example shown in Fig. 2, the maximum of all confidence scores is 1, and thus, this combination strategy would classify the community c_{ex} as accepted.

Correspondence analysis: SCANN. Correspondence analysis [5] is a multivariate statistical technique for analyzing multiway tables. It represents a data set in a lower-dimensional space based on its singular value decomposition. Although its role is similar to the principal component analysis one, correspondence analysis is designed for categorical data.

Using correspondence analysis, Merz [26] proposes an unsupervised combination strategy called SCANN that is used here as an alternative combination strategy. This method stores all the decisions of the detectors in a table, such that each entry is a vector representing the votes of all detectors for a certain community. Then, using correspondence analysis, this table is reduced such that the entries are smaller vectors containing only the main features characterizing the detectors votes. The benefit is that the reduced table contains only the significant votes. For instance, an irrelevant detector is one constantly making the same vote; in the first table built by SCANN, such a detector's votes are constant values, hence they will be ignored in the reduced table because they do not help for discriminating between the communities.

The reduced table contains the characteristics of each community in a low-dimensional space. Onto this low-dimensional space, SCANN projects two reference points which are two representative communities unanimously reported by the detectors as accepted, or as rejected. The class of each community is then determined according to which representative community the closest in the low-dimensional space.

Note that, since correspondence analysis is designed for categorical data, SCANN is unable to deal with the confidence scores previously defined. In order to take advantage of different configurations, the implementation of SCANN that is used consider directly the binary outputs of different configurations as its input.

3. DATA SET AND ANOMALY DETECTORS

3.1 Data set

The traffic we are labeling is from the MAWI (Measurement and Analysis on the WIDE Internet) archive sample-points B and F [9]. This archive contains daily traces representing 15 minutes of traffic captured from a trans-Pacific link between Japan and the United States. The data is publicly available; packet payloads are omitted and IP addresses are anonymized. MAWI started in January 2001, and thus, currently contains more than 9 years of traffic. Since 2001, the link has been updated three times, originally it was an 18 Mbps CAR on a 100 Mbps link, but it was updated to a full 100 Mbps link in 2006/07/01 and is currently a 150 Mbps link since June 2007. MAWI has enabled researchers to study Internet traffic characteristics [7, 16, 19], anomaly detectors [11, 14], and traffic classifiers [10].

In the experiments, the similarity estimator is evaluated with the traffic traces from the first week of every month from 2001 to 2009, whereas the combiner is evaluated using all the traffic traces from 2001 to 2009.

3.2 Anomaly detectors

Four unsupervised anomaly detectors, based on distinct statistical-analysis techniques, are implemented. As they report traffic at different granularities, the proposed similarity estimator is necessary to compare their results. The confidence score for each detector is obtained by tuning them with three different parameter sets corresponding to optimal, sensitive or conservative setting. Hence, for experiment, the input for the proposed method consists in the 12 outputs of all the configurations (4 detectors using 3 parameter tunings). The main ideas of the four detectors are as follows.

(1) Principal component analysis (PCA) is an unsupervised technique highlighting the main features of the data. This is perhaps the most studied technique for anomaly detection in backbone traffic. It was first applied by Lakhina et al. [21], and it has received much attention in the last few years [23, 30, 31]. The key idea underlying a PCA-based anomaly detector is the extraction of the main features defining a normal traffic behavior using PCA, then the distinct traffic is reported as anomalous. An inherent problem with PCA-based detectors is the retrieval of the original traffic features of the anomalous traffic [30]. In the experiments we overcame this difficulty by using random projection techniques (sketches) [23, 18]. This approach enables the PCA-based detector to report the source IP address of the identified anomalous traffic.

(2) Dewaele et al. introduced an anomaly detection method based on sketching and multi-resolution gamma modeling [11]. In a nutshell, the traffic is split into sketches and modeled using Gamma distribution. Traffic that is distant from an adaptively computed reference is reported as anomalous. The sketches are computed twice; the traffic is hashed on source addresses and destination addresses. Thus, this method reports source or destination IP addresses.

(3) The Hough transform is a pattern recognition technique that allows for the identification of a specific shape in a picture. This technique has been applied to several domains including anomaly detection of backbone traffic [14]. The approach proposed in [14] consists of first, monitoring the traffic in a 2-D scatter plot where the anomalous traffic appears as "lines", and second, identifies the anomalies with the Hough transform. The original data is retrieved from the identified plots, and the alarms reported by this method are aggregated sets of flows.

(4) The work presented in [8] detected the prominent changes in traffic by applying the Kullback-Leibler (KL) divergence to several kinds of histograms that monitor distinct traffic features. Then, association rule mining allows for the extraction of the set of traffic features that describes the anomalies detected by the histograms. Thus, the alarms reported by this anomaly detector are association rules, namely 4-tuples (source and destination IP addresses, source and destination port numbers) where elements can be omitted.

4. EVALUATION

4.1 Similarity estimator

In this section the proposed similarity estimator is evaluated using the alarms reported by the twelve configurations. In particular, the sensitivity of the similarity estimator to the traffic granularity is discussed.

4.1.1 Metrics for evaluation

The following tools enable a comparison of the results given by different configurations of the similarity estimator, and a validation of its efficiency.

Size of communities. The size of a community is the number of nodes that belong to that community, that is the number of similar alarms clustered in the community. We distinguish a specific class of community, called the **single communities**, that is the size 1 communities (communities with a single alarm). An alarm falls into a single community if the similarity estimator fails to find other alarms related to it. Consequently, we expect a good similarity estimator to minimize the number of single communities.

Obviously, the number of single communities is not a sufficient scale to evaluate the similarity estimator, as it reports a value 0 when all the alarms are connected regardless of their similarities. Consequently, we also score the relevance of the communities using association rule mining.

Traffic summary with association rules. One key task for validating the efficiency of the proposed similarity estimator is to inspect the traffic corresponding to each community. The goal of this inspection is to assess that each community is a group of related alarms standing for the traffic with common features; this is a similar goal to the *dominant state analysis* presented in [35], or the *association rule mining* of [8].

The traffic related to each community is profiled here using an association rule mining algorithm that finds sets of features, which are called rules, describing the prominent trends in a given list of properties. We choose the Apriori [2] algorithm as it is a well-known algorithm for achieving association rule mining. The Apriori algorithm efficiently counts the candidate rules in a breadth-first search manner. It finds all the rules that describe more than s elements of the data, where s is the only parameter of this algorithm. We slightly modify the Apriori algorithm to express s in a percentage rather than a fixed number of elements. For instance, the modified version of Apriori computed with $s = 20\%$ outputs each rule that describes at least 20% of the data.

Table 1: Heuristics labeling the traffic corresponding to a community into three categories ("Attack", "Special", and "Unknown"). These are originated from the anomalies previously reported [7, 14] and the manual inspection of MAWI.

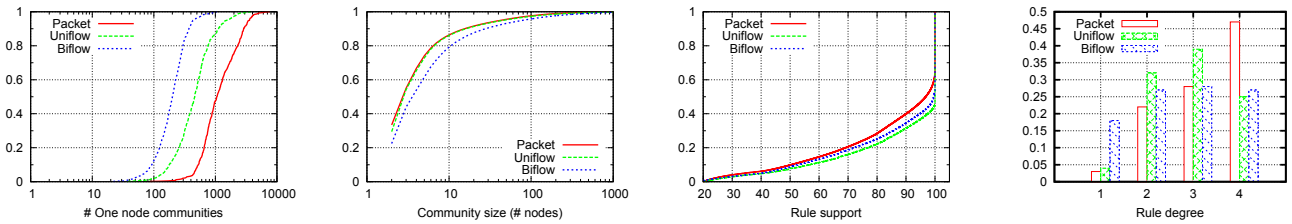
Label	Category	Details
Attack	Sasser	Traffic on ports 1023/tcp, 5554/tcp or 9898/tcp
Attack	RPC	Traffic on port 135/tcp
Attack	SMB	Traffic on port 445/tcp
Attack	Ping	High ICMP traffic
Attack	Other attacks	Traffic with more than 7 packets and: SYN, RST or FIN flag $\geq 50\%$ Or, http, ftp, ssh, dns traffic with SYN flag $\geq 30\%$
Attack	NetBIOS	Traffic on ports 137/udp or 139/tcp
Special	Http	Traffic on ports 80/tcp and 8080/tcp with less than 30% of SYN flag
Special	dns, ftp, ssh	Traffic on ports 20/tcp, 21/tcp, 22/tcp or 53/tcp&udp with less than 30% of SYN flag
Unknown	Unknown	Traffic that does not match other heuristics

In the experiments, the modified version of Apriori is arbitrarily tuned with $s = 20\%$, and it analyzes the packets or flows corresponding to each community. Thereby, the resulting rules describe the main characteristics of the traffic corresponding to a community in the form of 4-tuples — source IP address, source port number, destination IP address, destination port number. For example, a community corresponding to the traffic from a HTTP server IPA to two hosts, IPB and IPC, is depicted by the rules $< IPA, 80, IPB, * >$ and $< IPA, 80, IPC, * >$, where $*$ means that no specific destination port was identified.

The relevance of a community as a set of alarms is quantified by two efficiency metrics based on its rules:

- The **rule degree** of a community is the average number of items in its rules. For example, if a community has the two following rules, $< IPA, *, IPB, * >$ and $< IPA, 80, IPC, 12345 >$, then its rule degree is $(2 + 4)/2 = 3$. The rule degree ranges $[0, 4]$, and values close to 4 mean that the rules are specific, and thus, correspond to a particular kind of traffic, whereas values close to 0 mean that the mining rule algorithm failed to characterize specificities of the traffic.
- The **rule support** of a community is the percentage of data covered by all the rules of this community. For instance, if the two previous rules cover, respectively, 50% and 25% of the traffic captured by the community, and because the rules are disjoint, then the rule support is $50 + 25 = 75\%$.

Traffic inspection. The heuristics of Table 1 help to characterize traffic corresponding to communities. These heuris-



(a) CDF of number of single communities per traffic trace (log. scale)

(b) CDF of size of communities except for single communities (log. scale)

(c) CDF of rule support except for single communities

(d) Probability distribution of rule degree except for single communities

Figure 3: Characteristics of communities reported by the similarity estimator with different traffic granularities.

tics are designed from previous works [7, 14] and the manual inspection of the MAWI traffic. They assign three general labels to the traffic, "Attack", "Special", or "Unknown", highlighting the type of traffic corresponding to a community. Furthermore, they inspect only the TCP flag, ICMP code, and port number related information, and allow us to conduct a fair evaluation as they are independent of the mechanisms of the chosen detectors.

4.1.2 *Results*

The similarity estimator is evaluated using three distinct traffic granularities (packet, unidirectional flow, and bidirectional flow) by looking at the size of the communities and by inspecting the traffic corresponding to it.

Size of communities. The results highlight the benefit of flows to uncover similar alarms as Fig. 3(a) depicts a substantial decrease in the number of single communities using unidirectional or bidirectional flows. In addition, we observed a significant increase in the size of the communities when using bidirectional flows (Fig. 3(b)). These observations emphasize the ability of the similarity estimator to relate more alarms using flows.

Traffic summary. Let us check the consistency of the communities, that is whether all the alarms of the same community are actually related. Community consistency is analyzed using the rules that are assigned to each community by the modified Apriori algorithm. Figure 3(c) shows that the best rule support is achieved by using unidirectional flow, and the results obtained when using bidirectional flows are slightly inferior. By using unidirectional flows more than 50% of the communities have the rule support equal to 100%. However, the results are different regarding the rule degree (Fig.3(d)); the most accurate rules are obtained using packets whereas the least accurate are from bidirectional flows. We observe about 18% of the communities found using bidirectional flows are described with rules having only one traffic feature.

To understand which communities are suffering from coarse rules, thus containing dissimilar alarms, we investigated the relation between the size of communities and the rules efficiency. Figure 4 is the rule support, rule degree, and community size obtained when using unidirectional flows. We observe that the largest communities tend to have a rule degree equal to 1 and a rule support equal to 100%. A manual inspection of these communities reveals that they have coarse rules reporting a single traffic feature, usually a well known port such as 80 or 53 (Fig. 5). However 90% of the communities, namely with less than 20 nodes (Fig. 3(b)), have a rule degree higher than 2 and a rule support higher than 75% (Fig. 4). Similar observations are made using bidirectional flows, whereas using packets the rule degree is higher than 2.5 and the rule support above 70%. Therefore, the consistency of the communities identified the the similarity detector is satisfactory for the three traffic granularities. Selecting a traffic granularity is a trade off between the size of the communities and their consistency.

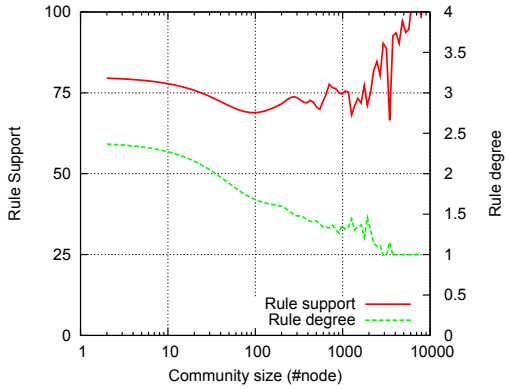
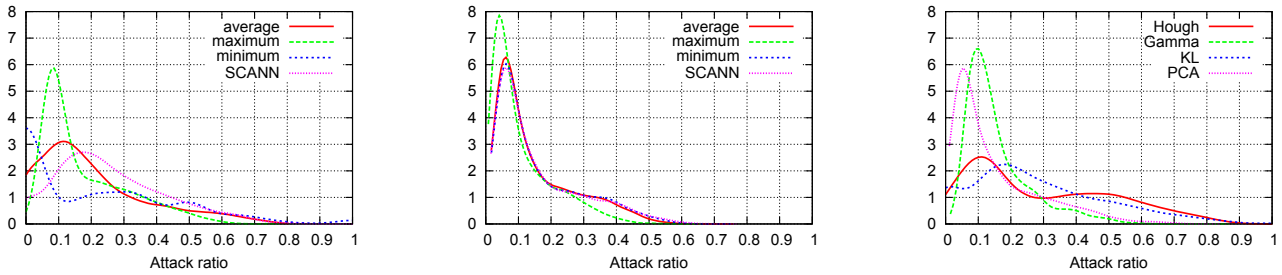


Figure 4: Rule degree, rule support, and size of communities identified by the similarity estimator using unidirectional flow. The curves are smoothed using weighted spline approximation.

Traffic inspection. Figure 5 depicts the intersections of the detectors outputs and the type of corresponding traffic. The main results are: (1) The intersection of the four detectors is significantly small in comparison to the total number of identified communities, therefore, the four detectors are sensitive to distinct traffic; (2) The number of single communities containing one alarm only from the PCA-based detector is significantly high, while only a few single communities



(a) PDF of attack ratio for accepted communities (large probability for high attack ratio is better)

(b) PDF of attack ratio for rejected communities (large probability for low attack ratio is better)

(c) PDF of attack ratio for detectors (large probability for high attack ratio is better)

Figure 6: PDF of attack ratio for four combination strategies and four detectors evaluated on 9 years.

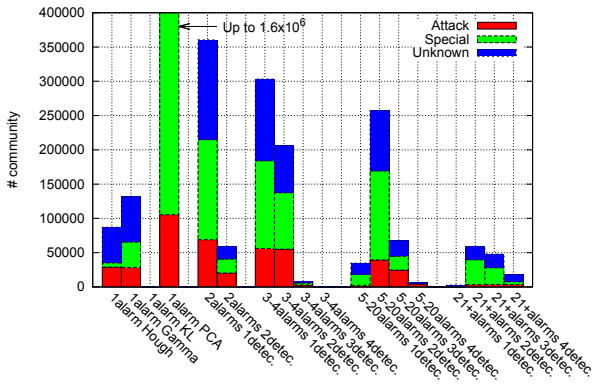


Figure 5: Number of communities as function of the size and the number of detectors reporting their alarms. Colors indicate the type of traffic corresponding (see Table 1).

are reported by the KL-based detector. Furthermore, 6% of the single communities identified by the PCA-based detector are labeled "Attack" whereas this ratio is significantly higher for other detectors: 33% for Hough, 22% for Gamma and 56% for KL. We also observe that the PCA-based detector represents 58% of the non-single communities identified by one detector. Thus, the output of the PCA-based is separated from others and its detection ratio is low in terms of the heuristics of Table 1. Regarding the communities identified by more than one detector, their attack ratio increases in tandem with the number of detectors identifying them.

The communities identified by several detectors certainly highlight anomalous traffic that have to be reported by the combiner. Nevertheless, the communities reported by a single detector have to be thoroughly investigated as they perhaps stand for anomalous traffic, particularly for those reported by the Hough, Gamma and KL-based detector.

4.2 Combiner

4.2.1 Attack ratio

In this work, measuring the accuracy of the four combination strategies is a contradictory task due to the lack of ground truth data. We bypass this issue by inspecting the results of the combiner with the heuristics of Table 1.

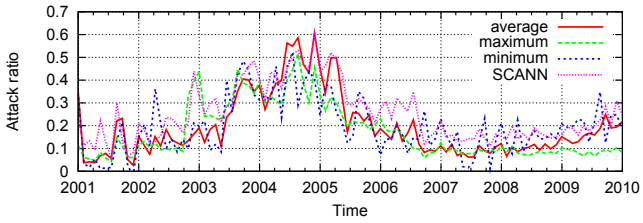
The heuristics label the communities reported by the similarity estimator into three groups: "Attack", "Special", and "Unknown". Since a relevant combination strategy is presumed to report the largest proportion of the communities labeled "Attack", we define the **attack ratio** as the amount of communities labeled "Attack" divided by the total number of identified communities. The combination strategies are expected to also report numerous communities labeled "Special" or "Unknown", thus low attack ratio, as the proposed heuristics might label incorrectly several kinds of anomalies. Nevertheless, the attack ratio is a reliable indicator that helps us to identify the best combination strategy, that is the one accepting the highest ratio of communities labeled "Attack" (Fig. 6(a) and 7(a)) and rejecting the lowest ratio of communities labeled "Attack" (Fig. 6(b) and 7(b)).

4.2.2 Comparison of combining strategies

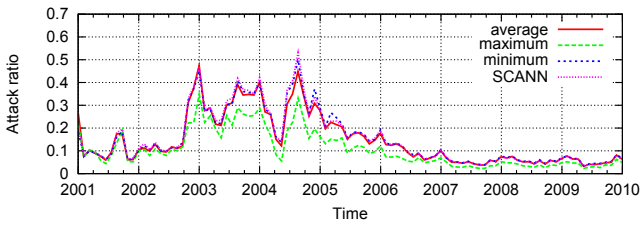
This section evaluates the ability of the four combination strategies to label communities. The analyzed communities are produced by the similarity estimator with the alarms reported by the four detectors on nine years of MAWI traffic and using unidirectional flow as traffic granularity. These communities are classified by the combination strategies into two classes (i.e., accepted and rejected) and the attack ratio of both classes are computed for each day of the analyzed traffic. Probability density functions (Fig. 6) and time-series of the attack ratio (Fig. 7) are displayed.

Regarding accepted communities, the best combination strategy is SCANN as it features the largest probability for highest attack ratio (Fig. 6(a)). Nevertheless, the best combination strategy regarding rejected communities is the *maximum* strategy because it has the largest probability for lowest attack ratio (Fig. 6(a)). Since the prominent variance between the attack ratio probability of the accepted communities and the one of the rejected communities highlights the best combination strategy, the experiments support SCANN as the best strategy for discriminating the communities representing anomalous traffic.

The probability density functions of the four anomaly detectors attack ratio emphasizes that all detectors, except the



(a) Accepted community attack ratio (higher value is better)



(b) Rejected community attack ratio (lower value is better)

Figure 7: Attack ratio of four combining strategies for nine years of MAWI traffic.

KL-based one, have an average attack ratio that is inferior to SCANN (Fig. 6(c)). Although the KL-based detector attack ratio is close to that of SCANN, the thorough investigation of the SCANN output in Section 4.2.3 asserts that SCANN detected twice more traffic than the KL-based detector.

The time evolution of the attack ratio for each combination strategy is depicted in Figures 7(a) and 7(b). Although the SCANN algorithm is not constantly outperforming the other combination strategies, it never has the worst attack ratio. The low attack ratio of both the accepted and rejected communities from 2007 is due to the simple heuristics listed in Table 1 that mislabeled the numerous elephant flows from peer-to-peer traffic and other anomalies using random ports. Still, between 2007 and 2010, the efficiency of SCANN is noticeable as its attack ratio for accepted communities was 2 to 3 times higher than the one for rejected communities.

However, the increase in the attack ratio for rejected communities from 2003 to 2005 (Fig. 7(b)) highlights the particular traffic that is missed by the combination strategies. The release of the Blaster worm in August 2003 followed by the release of the Sasser worm in May 2004 were two of the main events reported during this time period [7]. These two worms have substantially affected the main characteristics of the traffic and the four detectors were differently affected by this variance in traffic. The detectors reported numerous alarms that were not related to those of the other detectors, and consequently, the combiner failed in distinguishing several anomalous traffic. Nevertheless, this shortcoming of the combiner is inherently diminished by the combination of more detectors thus increasing the intersection of their outputs. Furthermore, we observed that selecting a single detector to analyze this traffic was also challenging, as the attack ratio of each detector critically fluctuated during this time period.

Table 2: Four measures quantifying benefits and losses when using SCANN

		SCANN	
		Accepted	Rejected
Label	Attack	gain_{acc}	cost_{rej}
	Special, Unknown	cost_{acc}	gain_{rej}

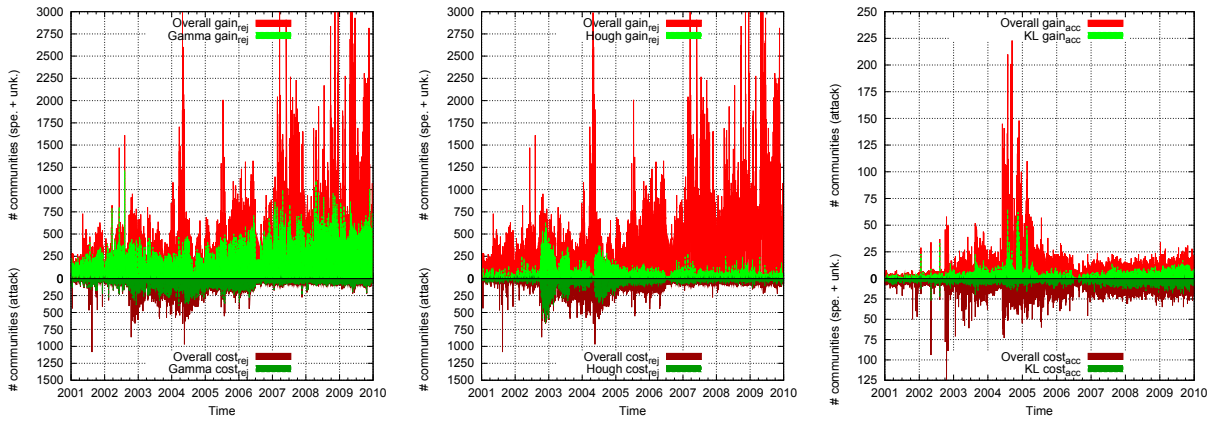
4.2.3 *Inspecting the SCANN output*

We evaluate the benefits and the losses of using SCANN based on the four quantities depicted in Table 2. For rejected communities the gain_{rej} is the amount of communities that are labeled "Special" or "Unknown", whereas the cost_{rej} is those labeled "Attack". Symmetrically, for the accepted communities, the gain_{acc} is the amount of communities that are labeled "Attack", whereas the cost_{acc} is those labeled "Special" or "Unknown".

Rejected communities. Figure 8 shows the breakout of communities classified by the SCANN algorithm. The two left-hand side plots are the communities rejected by SCANN where the alarms reported by the Hough- and the Gamma-based detectors are highlighted. The gain_{rej} for the Gamma-based detector (Fig. 8(a)) is substantial and stands for more than half of the overall gain_{rej} for all the detectors. Nevertheless, the high true positive rate of the Gamma-based detector is emphasized by its cost_{rej}, which represents most of the communities labeled "Attack" and rejected by SCANN. The gain_{rej} of the Hough-based detector was slightly higher than its cost_{rej} exhibiting the low false positive rate of this detector. In addition, Figure 8(b) depicts the high sensitivity of this detector to worm spreading (i.e., Blaster during 2003 and Sasser during 2004). The results for the PCA and KL-based detectors are omitted, as the former one has a significant gain_{rej} that is close to the overall gain_{rej}, and the latter one has no cost_{rej} and an negligible gain_{rej}. The experiments also exhibited the contamination of the normal subspace of the PCA-based detector [31] by the first release of the Sasser worm, and thus, a considerable gain_{rej} for this detector at this time period.

The PCA-based detector reported a significant number of alarms that were mostly unrelated to the alarms of other detectors (Fig. 5), particularly after the MAWI link update at the end of 2006 (overall gain_{rej} in Fig 8(a)). Since SCANN rejected most of the communities reported only by the PCA-based detector, the number of communities rejected by SCANN was notably higher than those of the accepted one (Fig. 8(b) and 8(c)). Figures 8(b) and 8(c) suggest that the overall cost_{rej} was higher than the overall gain_{acc}. However, we emphasize that the communities accepted by SCANN are more significant, in terms of the number of alarms and the amount of corresponding traffic, than the rejected ones.

Accepted communities. A manual inspection of the SCANN output reveals that several accepted communities contain only alarms from a single detector. Therefore, for the nine years of analyzed traffic, 8 accepted communities were identified



(a) Rejected communities (Gamma) (b) Rejected communities (Hough) (c) Accepted communities (KL)

Figure 8: Communities classified by SCANN as rejected with the alarms from the Hough (a) and the Gamma-based (b) detectors highlighted, and the communities accepted by SCANN with the alarms from the KL-based detector highlighted (c).

by only the PCA-based detector, 325 accepted communities were identified by only the Gamma-based detector, 2467 accepted communities were identified by only the Hough-based detector, and 352 accepted communities were identified by only the KL-based detector. Meaning that 82% of the communities reported exclusively by the KL-based detector are accepted by SCANN. This highlights the advantage of SCANN over the *average* combination strategy. Whereas the *average* combination strategy inherently rejects all the communities reported by a single detector, SCANN performs a finer analysis that emphasizes the output from accurate detectors and allows for the acceptance of small communities identified exclusively by these detectors. Indeed, the SCANN algorithm factorizes the detectors decisions by disregarding the unnecessary ones, thus, SCANN ignores the output of the detectors that are making irrelevant decisions and emphasizes the other results. For example, in the experiments the PCA-based detector output was mainly separated from the outputs of the other detectors (the single communities in Fig. 5). Consequently, SCANN frequently disregarded the PCA-based detector and accepted only 8 of the numerous communities exclusively identified by this detector. Conversely, the Hough-based detector reports more relevant alarms as many are related to those from other detectors, and thus, SCANN selects 2467 communities reported by only this detector.

In the experiments the best detector was the KL-based one (Fig. 6(c)). Almost all the alarms from this detector were related to another alarm (Fig. 5) and are accepted by SCANN. However, about 50% of the communities accepted by SCANN and labeled "Attack" are not identified by the KL-based detector (Fig. 8(c) and 9). These communities are mainly reported by the three other detectors and they highlight the high false negative rate (i.e., anomalies missed) of the KL-based detector (Fig. 9).

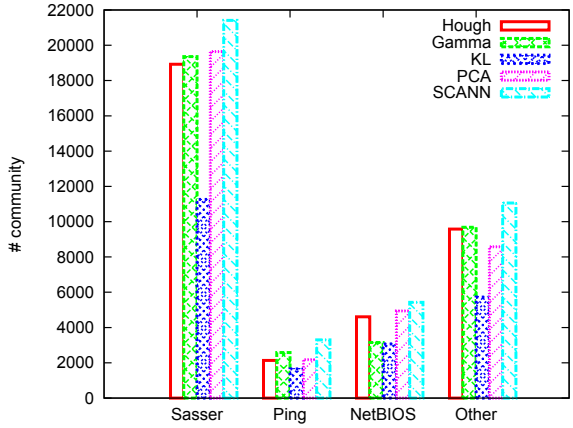


Figure 9: Breakdown of communities accepted by SCANN and labeled "Attack" by heuristics.

SCANN low dimensional space. Combining the four detectors with SCANN allows us to improve the results of the most accurate detector and to ignore the false alarms reported by all the detectors. However, Fig. 7(b) suggests that it misclassified several communities. As described in Section 2.2.3, the SCANN algorithm maps the communities in a reduced space and classifies them based on their distances to two reference points. Let d_{acc} and d_{rej} be the distance from a community to the reference point standing for, respectively, accepted and rejected communities, then the relative distance of the community is defined as $(d_{rej}/d_{acc}) - 1$. This metric ranges $[0, \infty)$, where 0 means that the community is on the threshold whereas higher values highlight the communities that are distant to it. The inspection of the rejected communities exhibits that the relative distance of those labeled "Attack" is lower than the one of those labeled "Special" or "Unknown" (Fig. 10).

We varied the discriminating threshold of SCANN during the experiments to investigate possible improvements. Tuning the threshold to accept more communities tends to

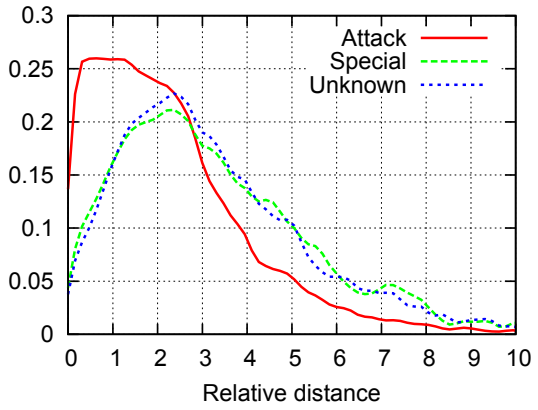


Figure 10: PDF of rejected communities relative distance classified with the heuristics of Table 1.

increase the fluctuations of the attack ratio of SCANN. For example, accepting all the communities within a relative distance of 0.5 achieved an attack ratio of 0.7 during the Sasser outbreak, but sometimes deteriorated the attack ratio, therefore, no global improvement was observed.

5. MAWI LABELING

Step 4 of the proposed method consists in labeling the analyzed traffic, here the MAWI archive. In agreement with the previous evaluation, the traffic is labeled using the SCANN combination strategy, and the similarity estimator was executed using unidirectional flow. Since several communities contained a significant number of alarms, we retrieved the common traffic features corresponding to all the alarms from the same community with the association rule mining algorithm presented in Section 4.1.1, and assigned labels to the traffic described by the community rules.

Using the SCANN output we define a simple traffic taxonomy with four different labels: *Anomalous*, *Suspicious*, *Notice*, and *Benign*. The traffic accepted by SCANN is labeled *Anomalous*, meaning that these traffic are abnormal and should be identified by any efficient anomaly detector. The traffic rejected by SCANN and having a relative distance lower or equal to 0.5 is labeled *Suspicious*. Most of these traffic are probably anomalous but are not clearly identified. The traffic also rejected by SCANN but having a relative distance greater than 0.5 is labeled *Notice*. Although these traffic are not anomalous and should not be identified by any anomaly detector, we do not label them as benign in order to trace all the alarms reported by the combined detectors. The other traffic is labeled *Benign* because none of the anomaly detectors identified it.

This labeling of the MAWI traffic is publicly available in the form of a database named MAWILab [1]. This database assists researchers in measuring the detection rate of their anomaly detector. The results of the emerging detectors can be accurately compared to the labels of MAWILab by using a similarity estimator like the one presented in this work.

6. DISCUSSION AND FUTURE WORK

In addition to its accurate detection, the proposed method has several advantages that are presented in this section.

The graph-based similarity estimator proposed in Section 2.1 is a valuable support for systematically benchmarking a detector against other detectors that report traffic at a different granularity. Indeed, by clustering diverse detectors alarms into communities, it allows the automated inspection of numerous detectors outputs in a rigorous manner.

Also, the community rules obtained from the rule mining algorithm consist of concise descriptions of the traffic identified by the numerous alarms being merged into the communities. Therefore, an anomalous traffic reported by numerous similar alarms is annotated with a single label. Thus, the number of labels assigned to the MAWI archive is significantly inferior to the number of alarms reported by the four detectors, and the labels are intelligible to humans.

Following the expansion of the MAWI archive, MAWILab is updated daily to track the latest trends in Internet traffic and upcoming anomalies. Furthermore, we will also take into account the results from emerging anomaly detectors, to improve the quality and variety of the labels over time. Indeed, by including new results from upcoming detectors the overlaps of the detectors outputs are emphasized and the accuracy of SCANN is improved. Therefore, MAWILab is constantly enhanced and represents a reference data set over time. In order to ease the evolution of MAWILab, we are planning to establish a collaborative system allowing researchers to easily contribute by submitting their anomaly detector or results.

We emphasize that the proposed implementation has the advantage of handling manual annotations or annotations from traffic classifiers [10]. Indeed, the similarity estimator is able to deal with any traffic annotations [13] containing at least two timestamps and one traffic feature. This significant ability of the approach allows us to label traffic with an exhaustive taxonomy. For instance, by adding in the method input the annotations from a traffic classifier, the similarity estimator aggregates similar alarms and corresponding annotations in the same community. Afterwards, the combiner classifies the communities by ignoring the annotations, but the accepted communities are still reported with the extra information provided by the annotation.

The goal of this work is to find and label traffic anomalies off-line, so we assume no constraint is placed on the execution time of the approach. Nevertheless, the experiments revealed that the current implementation requires only a few minutes to combine alarms with a 15-minute traffic trace, thus enabling for real time analysis. However, the study of concurrently running anomaly detectors in real time is left for future work.

Furthermore, we are also interested in studying the sensitivities of the anomaly detectors to estimate the best candidates to combine and to evaluate the combination strategies based on detector selection.

7. CONCLUSIONS

We proposed a methodology that find network traffic anomalies in the MAWI archive by comparing and combining the results from four anomaly detectors. The approach consists of two main ingredients; first, a graph-based similarity estimator systematically uncovers the relations between the alarms reported by the detectors, second, a combiner classifies the similar alarms using a combination strategy. We evaluated the effectiveness of both using different traffic aggregations and combination strategies. The experiments emphasized the benefit of combining detectors with SCANN, a strategy based on dimensionality reduction, as it ignored irrelevant alarms and detected twice more anomalous traffic than the more accurate combined detector.

The established methodology allows us to accurately detect anomalies in the MAWI archive and precisely assign concise labels. The results are updated daily using the MAWI archive and are publicly available [1] to assist researchers in benchmarking their detectors. We encourage researchers to contribute to the proposed system by submitting to us their results or detectors, so we can maintain a reliable labeling of the MAWI archive.

Acknowledgments. We thank Guillaume Dewaele, Yoshiki Kanda, and Jean-Loup Guillaume for providing us with their source codes of, respectively, the Gamma-based detector, the PCA-based detector, and the community mining algorithm. Work partially supported by the CNRS/JSPS-2010-2011-PRC program.

References

[1] MAWILab. *http://www.fukuda-lab.org/mawilab/*.
[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94*, pages 487–499, 1994.
[3] A. B. Ashfaq, M. Javed, S. A. Khayam, and H. Radha. An information-theoretic combining method for multi-classifier anomaly detection systems. *ICC '10*, page 5, 2010.
[4] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. *IMW '02*, pages 71–82, 2002.
[5] J.-P. Benzécri. *Correspondence Analysis Handbook*. Marcel Dekker, New York, 1992.
[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J.STAT.MECH.*, 2008.
[7] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho. Seven years and one day: Sketching the evolution of internet traffic. *INFOCOM '09*, pages 711–719, 2009.
[8] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. *IMC '09*, pages 28–34, 2009.
[9] K. Cho, K. Mitsuya, and A. Kato. Traffic data repository at the WIDE project. In *USENIX 2000 Annual Technical Conference: FREENIX Track*, pages 263–270, 2000.
[10] H. chul Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: Myths, caveats, and the best practices. *CoNEXT '08*, 2008.
[11] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. *SIGCOMM LSAD '07*, pages 145–152, 2007.
[12] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Trans. Netw.*, 9(4):392–403, 2001.
[13] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. Uncovering relations between traffic classifiers and anomaly detectors via graph theory. In *International Workshop on Traffic Monitoring and Analysis (TMA '10)*, pages 101–114, 2010.
[14] R. Fontugne and K. Fukuda. A Hough-transform-based anomaly detector with an adaptive time interval. *ACM SAC '11*, 2011.
[15] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.
[16] H. Gupta, V. J. Ribeiro, and A. Mahanti. A longitudinal study of small-time scaling behavior of internet traffic. In *Proceedings of NETWORKING 2010*, pages 83–95, 2010.
[17] Y. Himura, K. Fukuda, K. Cho, and H. Esaki. An automatic and dynamic parameter tuning of a statistics-based anomaly detection algorithm. *ICC '09*, page 6, 2009.
[18] Y. Kanda, K. Fukuda, and T. Sugawara. An evaluation of anomaly detection based on sketch and PCA. *GLOBECOM '10*, 2010.
[19] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary poisson view of internet traffic. In *INFOCOM '04*, 2004.
[20] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
[21] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. *SIGCOMM '04*, pages 219–230, 2004.
[22] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *SIGCOMM '05*, pages 217–228, 2005.
[23] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. *IMC '06*, pages 147–152, 2006.
[24] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579 – 595, 2000.
[25] J. Mchugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur.*, 3(4):262–294, 2000.
[26] C. J. Merz. Using correspondence analysis to combine classifiers. *Mach. Learn.*, 36(1-2):33–58, 1999.
[27] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An empirical evaluation of entropy-based traffic anomaly detection. *IMC '08*, pages 151–156, 2008.
[28] P. Owezarski. A database of anomalous traffic for assessing profile based IDS. In *International Workshop on Traffic Monitoring and Analysis (TMA '10)*, pages 59–72, 2010.
[29] H. Ringberg, M. Roughan, and J. Rexford. The need for simulation in evaluating anomaly detectors. *SIGCOMM Comput. Commun. Rev.*, 38(1):55–59, 2008.
[30] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 35(1):109–120, 2007.
[31] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. *IMC '09*, pages 1–14, 2009.
[32] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry. Non-Gaussian and Long Memory Statistical Characterisations for Internet Traffic with Anomalies. *IEEE Transaction on Dependable and Secure Computing*, 4(1):56–70, 02 2007.
[33] S. Shanbhag and T. Wolf. Accurate anomaly detection through parallelism. *Netwrk. Mag. of Global Internetwkg.*, 23(1):22–28, 2009.
[34] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. *IEEE international conference on Computational intelligence for security and defense applications (CISDA '09)*, pages 53–58, 2009.
[35] K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Trans. Netw.*, 16(6):1241–1252, 2008.