

# Statistical Network Anomaly Detection: An Experimental Study

Christian Callegari<sup>1</sup>(✉), Stefano Giordano<sup>2</sup>, and Michele Pagano<sup>2</sup>

<sup>1</sup> RaSS National Laboratory, CNIT, Pisa, Italy  
`christian.callegari@cnit.it`

<sup>2</sup> Department of Information Engineering, University of Pisa, Pisa, Italy  
`{stefano.giordano,michele.pagano}@iet.unipi.it`

**Abstract.** The number and impact of attack over the Internet have been continuously increasing in the last years, pushing the focus of many research activities into the development of effective techniques to promptly detect and identify anomalies in the network traffic. In this paper, we propose a performance comparison between two different histogram based anomaly detection methods, which use either the Euclidean distance or the entropy to measure the deviation from the normal behaviour. Such an analysis has been carried out taking into consideration different traffic features.

The experimental results, obtained testing our systems over the publicly available MAWILab dataset, point out that both the applied method and the chosen descriptor strongly impact the detection performance.

## 1 Introduction

The ever increasing number of attacks over the Internet and the serious consequences that these can have in the citizens life have pushed the focus of many research activities into the design and development of effective tools to promptly detect and identify anomalies in the network traffic. As a result, many different approaches have been proposed in the last decade, but the ultimate solution is still far from being identified.

Among the different proposals, promising results are offered by the methods based on the estimation of the distribution of a given traffic feature (histogram based methods).

Nonetheless, even these anomaly detection systems are still affected by serious limitations (mainly in terms of missed detections and false alarms), either due to the intrinsic inability of the chosen method to deal with some kind of anomalies or to the low appropriateness of the chosen traffic feature to properly discriminate between normal and anomalous activities.

For this reason, in this paper, we propose an experimental study of two distinct approaches:

- one based on the computation of the Euclidean distance between the histograms of a given traffic descriptor computed in different time-bins (namely the current time-bin and a reference anomaly-free time-bin);

- one based on the variation of the entropy associated to the histograms of a given traffic descriptor computed in different time-bins.

Apart from simply comparing such methods, in this paper we have also investigated if their “relative” performance is constant when varying the considered traffic metric. In other words, we have verified if we can identify a method that outperform the other despite the chosen traffic descriptor. For this reason, focusing on volume anomalies as representative of a very widespread phenomenon, we have chosen to take into consideration two distinct traffic descriptors, namely the number of distinct flow destined to a given traffic aggregate and the quantity of bytes received by the same aggregate.

Interestingly, the experimental results, obtained testing our systems over the publicly available MAWILAb dataset, show that not only does the choice of both the statistical detection method and the considered traffic descriptor have a strong impact on the performance, but that the choice of the “best” detection method also depends on the considered traffic feature (as pointed out by the experimental results).

It is worth highlighting that for addressing the scalability issues, both the methods work on top of traffic aggregates (not traffic flow). Given the literature on the topic (see next section for more details), we have chosen to aggregate the traffic using probabilistic data structures (i.e., reversible sketches).

The rest of the paper is organised as follows: Sect. 2 gives a brief overview of the related works, and in Sect. 3 we provide a quick review of some background knowledge. Then in Sect. 4 we detail the proposed anomaly detection method. Hence, the used data-set for the experimental tests is described in Sect. 5, and the achieved performance is discussed in Sect. 6. Finally, Sect. 7 concludes the paper with some final remarks.

## 2 Related Work

Anomaly detection is a general framework including different analysis techniques, so it is not surprising that several works have been published in recent years, dealing with specific methods or providing a *general overview* of the different approaches (see, for instance, [1–3], which focuses on the features of network data and provides general guidelines for the design of IDSs). In the following, we only discuss the most relevant contributions closely related to this work.

Sketches, by themselves, cannot be considered as a detection method, but they are frequently used as a building block of several IDSs [4–8]. Indeed, random aggregation performed by sketches “efficiently” reduces the dimension of the data (wrt “classical” deterministic aggregations, such as according to input/output routers [9]); moreover, through the use of reversible sketches [10] it is possible to identify the flows responsible for the anomalies.

Regarding histogram based IDSs, in [11] the behavior of the monitored network during every time bin is characterized by means of histograms representing the distribution of the number of flows, packets or bytes over the values

of a traffic feature. Anomalies are detected by comparing, through a distance function (namely, Euclidean distance, Manhattan distance, Mahalanobis distance, Kullback-Leibler divergence, and Jensen-Shannon divergence) the current histogram with a reference one, built during the training phase. In [12] the histogram cloning method is introduced: multiple randomized histograms are obtained through independent hash functions and the Kullback-Leibler divergence is used to detect anomalies.

Entropy has been applied to intrusion detection in different frameworks. For instance, in [13] fast Internet worms are detected taking into account the entropy contents (more precisely, the Kolmogorov complexity) of traffic parameters, such as IP addresses, while [14] focuses on network traffic running over TCP. In both cases an upper bound of Shannon entropy has been estimated through the use of different state-of-the-art compressors. Instead, in [15] Shannon entropy “summarizes” the distribution of specific traffic features to detect unusual traffic patterns.

### 3 Theoretical Background

In this section we present some theoretical background information, necessary to understand the proposed architecture. Note that we focus on the useful details only, referring the reader to the provided references for a complete description of the different topics.

#### 3.1 Reversible Sketches

A sketch is a probabilistic data structure (a two-dimensional array) that can be used to summarize a data stream, by exploiting the properties of the hash functions [6]. Sketches differ in how they update hash buckets and use hashed data to derive estimates.

In more detail, a sketch is a two-dimensional  $D \times W$  array  $T_{D \times W}$ , where each row  $d$  ( $d = 0, \dots, D - 1$ ) is associated to a given hash function  $h_d$ . These functions give an output in the interval  $(0, \dots, W - 1)$  and these outputs are associated to the columns of the array. As an example, the element  $T[d][j]$  is associated to the output value  $j$  of the hash function  $h_d$ .

When a new item  $(i_t, c_t)$ , where  $i_t$  is the key (e.g., a destination IP address) and  $c_t$  is the weight (e.g., the number of received bytes), arrives, the sketch is updated as follows:

$$T[d][h_d(i_t)] \leftarrow T[d][h_d(i_t)] + c_t \quad (1)$$

and the update procedure is repeated for all the different hash functions.

Given the use of the hash functions, such data structures are not reversible, which makes impossible, after the detection, to identify the IP addresses responsible of an anomaly. To overcome such a limitation, in our system we have used an improved version of the sketch, that is the reversible sketch [10].

### 3.2 Entropy

The most basic concept in information theory is the entropy of a random variable (RV)  $X$ , often called Shannon entropy [16]. Roughly speaking, it is a measure of the uncertainty (or variability) associated with the RV.

In more detail, let  $P = \{p_1, p_2, \dots, p_L\}$  be the probability distribution of the discrete RV  $X$ , i.e.

$$0 \leq p_l \leq 1 \quad \text{and} \quad \sum_{l=1}^L p_l = 1$$

Then its Shannon entropy is defined as follows:

$$H(X) = - \sum_{l=1}^L p_l \log_2 p_l = \mathbb{E}[-\log_2 P(X)] \quad (2)$$

where  $\mathbb{E}$  denotes the expectation operator, and is measured in bits (or shannon). Note that a change in the base of the logarithm just corresponds to a multiplication by a constant and a change in the unit of measure (nat for the natural logarithm and hartley (or ban) for the base 10 logarithm). In particular, when the natural algorithm is considered, (2) coincides with the well-known Boltzman–Gibbs entropy in statistical mechanics.

It is well-known that  $0 \leq H(X) \leq \log_2 L$ , where the infimum corresponds to the degenerate distribution (i.e.,  $p_l = \delta_{k-l}$  for some integer  $k$  with  $1 \leq k \leq L$ ) and the supremum is attained in case of uniform distribution (i.e.,  $p_l = 1/L \forall l$ ).

### 3.3 Euclidean Distance

The Euclidean distance (or Euclidean metric) corresponds to the usual distance between two points in an Euclidean space (in  $\mathbb{R}^2$  it is equivalent to the well-known Pythagorean theorem). It can be seen as a special case (for  $p = 2$ ) of the Minkowski distance of order  $p$

$$d_p(P, Q) = \left( \sum_{l=1}^L |p_l - q_l|^p \right)^{1/p}$$

We recall that for  $p \geq 1$ , the Minkowski distance is a metric (as a result of the Minkowski inequality); instead for  $p < 1$  the triangle inequality does not hold (see, for instance, [17] for further details).

## 4 System Architecture

The proposed system takes as input traffic data over a predefined time-bin (in the following we assume to have  $N$  distinct time-bins), whose length can be arbitrarily set by the network administrator. Note that the duration of the time-bin is a compromise between the detection delay (the decision is taken at the

end of the time bin) and the need of collecting enough data in order to build significant statistics. In more detail, the information associated to each time-bin is a list of keys  $i_t$  (e.g., the list of destination IP addresses) observed during that time-bin and the associated weights  $c_t$  (in our case, the number of bytes and flows for that IP address). Such information can be easily extracted from standard network traffic data, for instance parsing NetFlow traces by using the Flow-Tools [18].

The input data are processed to build the reversible sketch tables. In our case, each bucket will contain an histogram, representing the empirical distribution (estimated over  $L$  bins) of the weight values associated to all the keys that are mapped, by the corresponding hash function, in the given bucket. In this way, we have obtained  $T$  distinct sketches  $T_{D \times W \times L}^t$ , where  $t \in [1, N]$  is the time-bin (in the experimental tests we have set  $W = 512$ ,  $D = 16$ , and  $L = 64$ ).

Then, the sketches are passed to the actual anomaly detection phase, where, for each bucket of the current sketch  $T^t[d][w][\cdot]$ , the system performs one of the following operations:

- entropy based method: the system computes the entropy associated to the current histogram and the difference between such a value and the entropy associated to the same bucket in the reference sketch (i.e., the last non-anomalous processed sketch);
- distance based method: the system computes the Euclidean distance between the current histogram and the histogram stored in the same bucket of the reference sketch.

Finally, such a value (either the entropy difference or the Euclidean distance) is compared with a threshold to decide if there is an anomaly or not. Note that, given the nature of the sketches, each traffic flow is part of  $D$  random aggregates and hence it will be checked  $D$  times to verify if any anomaly is present (indeed, an anomalous flow could be masked in a given traffic aggregate, while being detectable in another one).

Due to this fact, a voting algorithm is applied for each time-bin: the algorithm simply verifies if at least  $H$  (where  $H$  is a tunable parameter, with  $H = D/2 + 1$  in our experiments) rows of the sketch contain at least one anomalous bucket. If so, the system reveals an anomaly and the responsible IP addresses are identified (by using the reversible sketch functionalities [10]).

## 5 MAWILab Dataset

The dataset used to evaluate our anomaly detection methods consists of packet traces from the MAWI (Measurement and Analysis on the WIDE Internet) archive (sample-points B and F), publicly available at [19]. Each trace in this database collects the traffic captured for 15 min in a specific day, since 2001 until nowadays, on a trans-Pacific link between Japan and the USA.

As in almost all existing databases, the key problem in testing the IDS performance is represented by a precise knowledge of the anomalies existing in the captured traffic. Such information is essential for building a proper ROC (Receiver

Operating Characteristic) curve and evaluating new approaches. Although also for the MAWI archive, an exact description of the attacks is not available, the dataset presents two important features that make it suitable for the performance evaluation procedure:

- unlike the widely-used DARPA dataset, the network is not emulated and the traffic mixture is representative of the current mixtures of network services and applications;
- in the framework of the successive project MAWILab [20], every traffic flow is classified by means of labels, which indicate the probability (according to well-known anomaly detection algorithms) that an anomaly is present. Since these labels are available together with the traces, they can be used as a common reference for testing a new IDS.

In more detail, the traces classification has been obtained combining the output of four anomaly detectors (based respectively on the Hough transform, the Gamma distribution, the Kullback-Leibler divergence and the Principal Component Analysis) [21]. As a result, the traffic is split into four categories:

- *anomalous*: traffic that is anomalous with high probability;
- *suspicious*: traffic that is probably anomalous, but not clearly identified by the MAWI classification methods;
- *notice*: non anomalous traffic, but that has been reported by at least one of the four anomaly detectors;
- *benign*: normal traffic.

The anomalies (*anomalous* and *suspicious* flows) are listed in an xml file for each trace, identifying them by means of traffic features as source and destination IP addresses, source port, destination port and transport protocol. Furthermore, some information about the kind of anomaly are also given:

- *attack*: anomalies representing a well known attack;
- *special*: anomalies involving well known ports;
- *unknown*: unknown kinds of anomalies.

Hence, the effectiveness of an IDS can be evaluated comparing the alarms generated by the new IDS with the labeled flows in the traffic traces, possibly referring to the three above-mentioned anomalous behaviors. Nevertheless, it is important to take into account the probabilistic nature of the MAWI classification in the interpretation of the achieved results.

## 6 Experimental Results

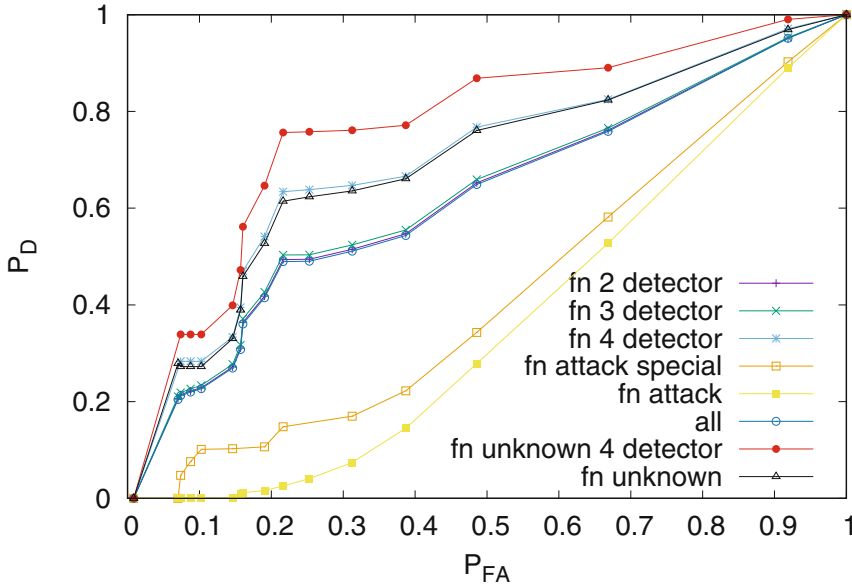
In this section we discuss the experimental results over the MAWILab dataset. The most widely used performance indicators are represented by the ROC curve and the Area under the Curve (AuC). Taking into account the MAWI labels,

we consider as “false positives” the flows that are not labeled as “anomalous” or “suspicious” in the MAWI archive, but that are anomalous according to the tested IDS, so the false alarm probability  $P_{FA}$  is the ratio between the number of “false positive flows” and the number of flows that are neither “anomalous” nor “suspicious”.

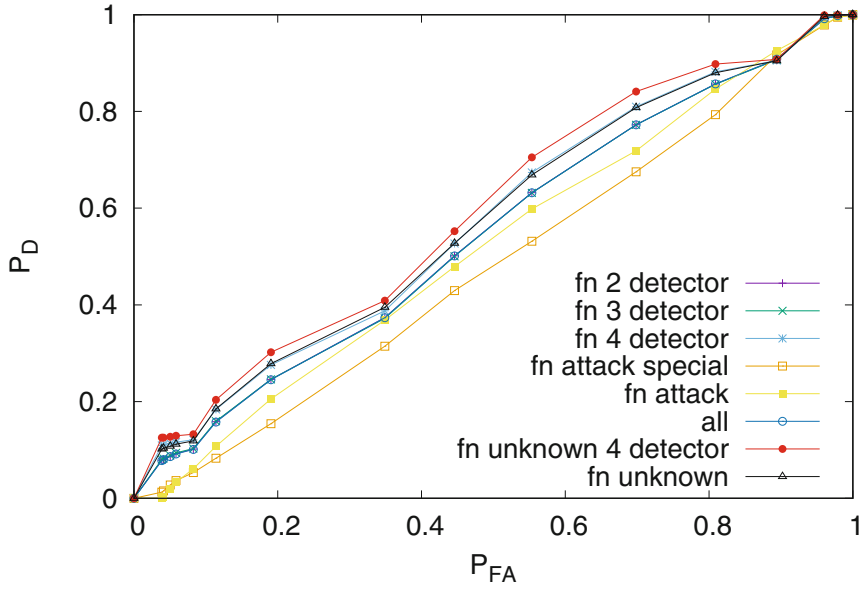
On the other hand, the false negative rate  $P_{FN}$  (note that the detection probability  $P_D$  can be obtained simply as  $P_D = 1 - P_{FN}$ ) is the ratio between the number of false negatives and the number of “anomalous” flows. But, in this case  $P_{FN}$  depends on the actual interpretation of the MAWILab labels, and can be defined in several ways.

In more detail, as discussed in [22], the number of false negatives can be calculated as (the labels are used in the following figures to identifies the corresponding definitions of  $P_D$ ):

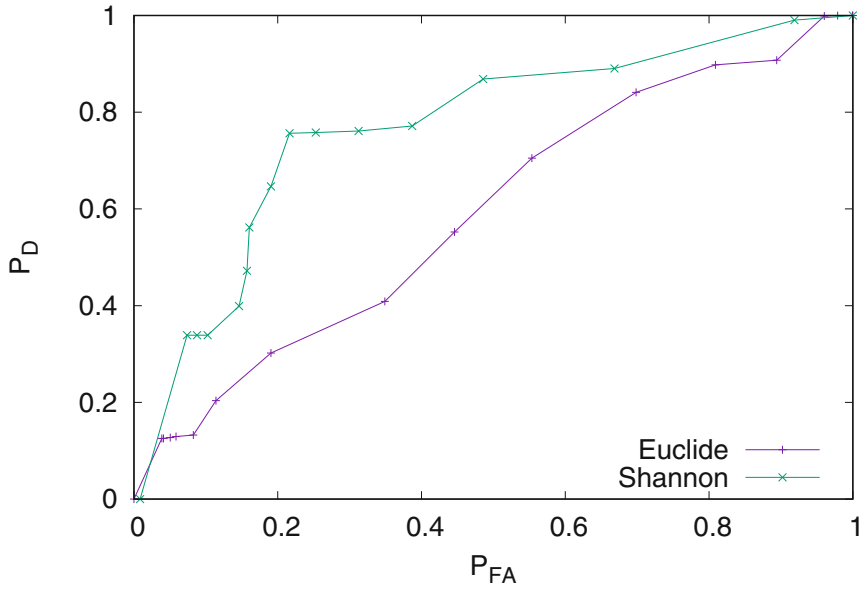
- “all”: the number of unrevealed flows labeled as “anomalous”;
- “fn 2/3/4 detector”: the number of unrevealed flows labeled as “anomalous” and detected at least by two/three/four of the four detectors used in MAWI classification;
- “fn attack”: the number of unrevealed flows labeled as “anomalous” belonging to the “attack” category (known attacks);
- “fn attack special”: the number of unrevealed flows labeled as “anomalous” belonging to the “attack” category or the “special” category (attacks involving well-known ports);



**Fig. 1.** ROC curves - entropy (flow)



**Fig. 2.** ROC curves - Euclidean distance (flow)



**Fig. 3.** ROC curves - comparison (flow)

- “fn unknown”: the number of unrevealed flows labeled as “anomalous” belonging to the “unknown” category (unknown anomalous activities);
- “fn unknown 4 detector”: the number of unrevealed flows labeled as “anomalous” belonging to the “unknown” category and detected by all the four detectors used in MAWI classification.

Given these definitions, in the following we discuss the results achieved by our system when taking into consideration, as traffic descriptors, either the number of flows with the same destination IP address (referred to as Flow in the following) or the quantity of traffic received by each IP address expressed in bytes (referred to as Byte in the following).

The first set of figures (namely Figs. 1, 2, and 3) refers to the Flow case. In more detail Fig. 1 shows the results achieved when using the entropy, for all the above mentioned definitions of  $P_{FN}$ . As it appears clearly, the offered performance strongly depends on the definition of  $P_{FN}$ , ranging from the completely unacceptable cases of “fn attack” and “fn attack special” to the very good case of “fn unknown 4 detector” (with a detection rate of about 80 % in correspondence of a false alarm rate less than 20 %). These results are very promising, taking into consideration that the usage of an anomaly detection system (normally in cascade to a misuse-based detection system) is conceived for detecting the “unknown” anomalies (given that the known attacks can be better detected by the other system).

**Table 1.** AuC (Flow)

Method	Label	AuC
Euclidean distance	All	0.546382
Euclidean distance	fn 2 detector	0.546917
Euclidean distance	fn 3 detector	0.546582
Euclidean distance	fn 4 detector	0.570564
Euclidean distance	fn attack	0.520335
Euclidean distance	fn attack special	0.481449
Euclidean distance	fn unknown	0.57054
Euclidean distance	fn unknown 4 detector	0.590988
Entropy	All	0.61949
Entropy	fn 2 detector	0.621851
Entropy	fn 3 detector	0.627007
Entropy	fn 4 detector	0.699259
Entropy	fn attack	0.362535
Entropy	fn attack special	0.418059
Entropy	fn unknown	0.693165
Entropy	fn unknown 4 detector	0.768803

Figure 2 presents the same analysis when using the Euclidean distance, showing that such a method is far from providing good results, having for all of the plots a behaviour very close to the diagonal.

A more precise comparison between the two methods is shown in Table 1 and in Fig. 3, where we respectively present the AuC obtained by the two methods when varying the definition of  $P_{FN}$  and the ROC achieved in the “fn unknown 4 detector” case. Hence, we can easily conclude that the entropy method definitely offers better performance than Euclidean distance when using Byte as traffic descriptor.

A completely analogous performance analysis is presented in the subsequent figures and table, where we show the ROCs and the AUC values for the two systems, obtained when using Byte as traffic descriptor. In this case it is very interesting to make two observations:

- the offered performance is, in any case, very far from those related to the use of Flow as traffic descriptor (see Figs. 4 and 5, where “almost unacceptable” ROC are shown), demonstrating how the choice of the correct traffic descriptor is crucial in anomaly detection;
- contrarily to the Flow case, better performance is offered by the Euclidean distance (see Fig. 6), demonstrating how the anomaly detection method must be properly chosen (also taking into account the used traffic descriptor).

Finally, Table 2 presents all the values of the AuC for the Byte case, confirming the results shown in Figs. 4 and 5.

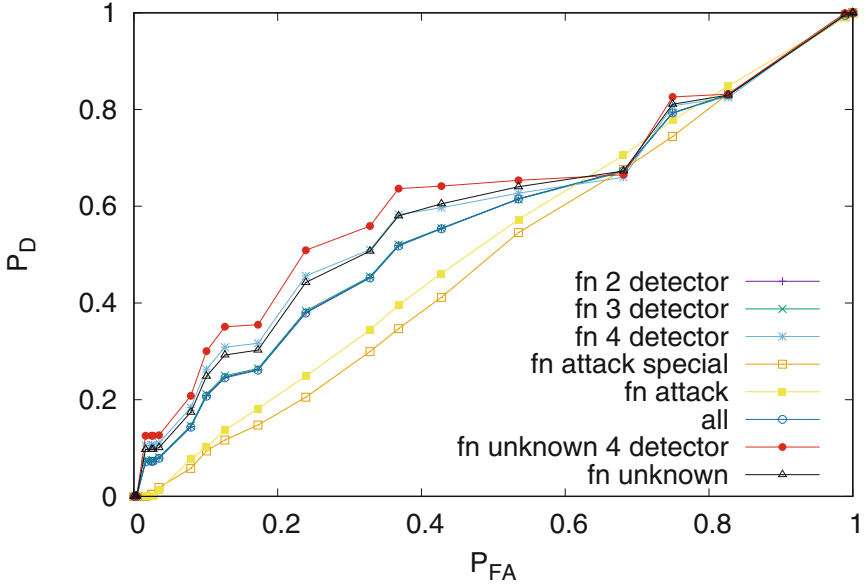


Fig. 4. ROC curves - Euclidean distance (byte)

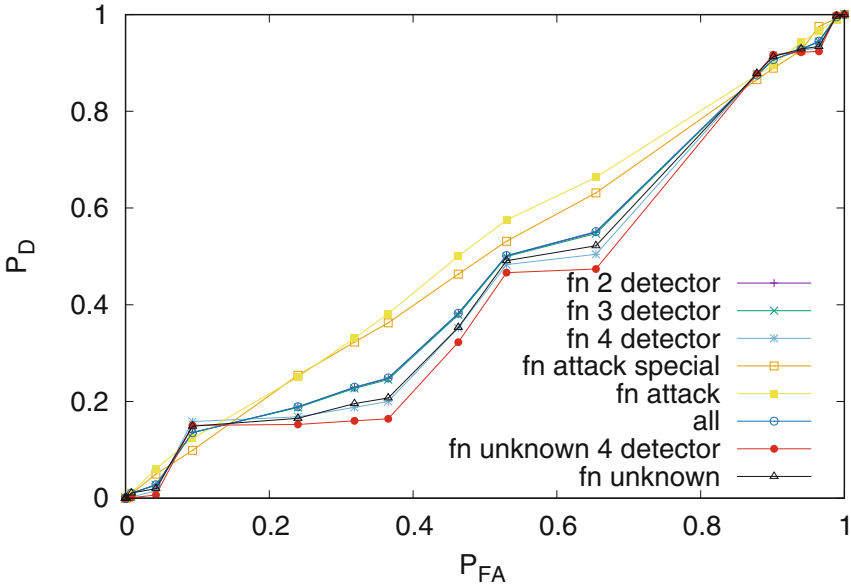


Fig. 5. ROC curves - Entropy (byte)

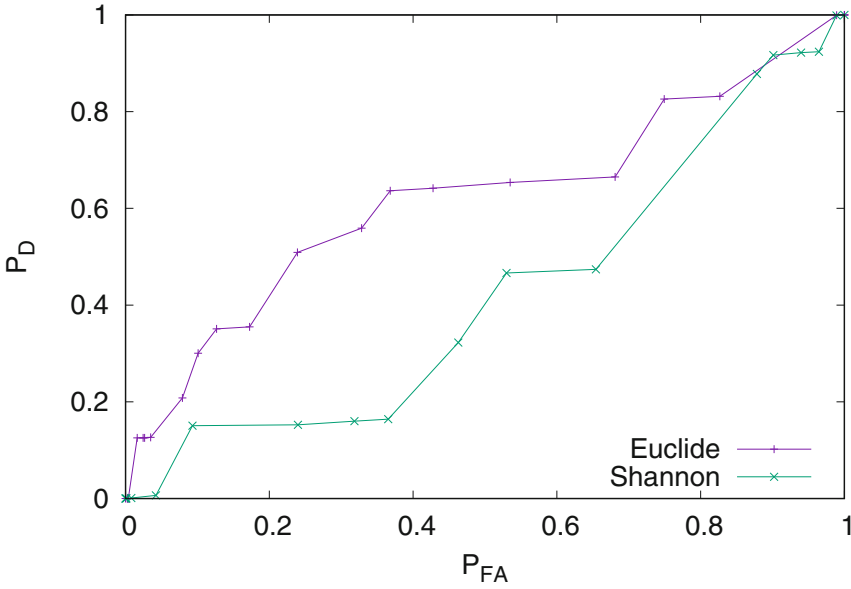


Fig. 6. ROC curves - comparison (byte)

**Table 2.** AuC (Byte)

Method	Label	AuC
Euclidean distance	All	0.566218
Euclidean distance	fn 2 detector	0.566777
Euclidean distance	fn 3 detector	0.567148
Euclidean distance	fn 4 detector	0.593179
Euclidean distance	fn attack	0.517885
Euclidean distance	fn attack special	0.49092
Euclidean distance	fn unknown	0.59376
Euclidean distance	fn unknown 4 detector	0.619295
Entropy	All	0.455644
Entropy	fn 2 detector	0.455011
Entropy	fn 3 detector	0.453766
Entropy	fn 4 detector	0.436463
Entropy	fn attack	0.515499
Entropy	fn attack special	0.496697
Entropy	fn unknown	0.440622
Entropy	fn unknown 4 detector	0.421189

## 7 Conclusion

In this paper we have proposed an experimental comparison between two different histogram based anomaly detection methods. Moreover, the impact of the considered traffic descriptor on the achieved performance has been investigated.

The experimental results, obtained testing our systems over the publicly available MAWILab dataset, have clearly demonstrated that

- the choice of the correct traffic descriptor is crucial in anomaly detection;
- the anomaly detection method must be properly defined (also taking into account the used traffic descriptor).

These results show that the deployment of an anomaly detection tool requires a fine tuning of the system, also based on a good knowledge of the considered network scenario and traffic.

**Acknowledgment.** This work was partially supported by Multitech SeCurity system for intercOnnected space control groUnD staTions (SCOUT), a FP7 EU project.

## References

1. Thottan, M., Liu, G., Ji, C.: Anomaly detection approaches for communication networks. In: Cormode, G., Thottan, M., Sammes, A.J. (eds.) *Algorithms for Next Generation Networks*. Computer Communications and Networks, pp. 239–261. Springer, London (2010)
2. Ahmed, M., Naser Mahmood, A., Hu, J.: A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **60**(C), 19–31 (2016)
3. Callegari, C., Coluccia, A., D’Alconzo, A., Ellens, W., Giordano, S., Mandjes, M., Pagano, M., Pepe, T., Ricciato, F., Zuraniewski, P.: A methodological overview on anomaly detection. In: Matijasevic, M., Callegari, C., Biersack, E. (eds.) *Data Traffic Monitoring and Analysis*. LNCS, vol. 7754, pp. 148–183. Springer, Berlin (2013)
4. Subhabrata, B.K., Krishnamurthy, E., Sen, S., Zhang, Y., Chen, Y.: Sketch-based change detection: methods, evaluation, and applications. In: *Internet Measurement Conference*, pp. 234–247 (2003)
5. Borgnat, P., Dewaele, G., Fukuda, K., Abry, P., Cho, K.: Seven years and one day: sketching the evolution of internet traffic. In: *INFOCOM*, April 2009
6. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. *J. Algorithms* **55**(1), 58–75 (2005)
7. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature. In: *ACM SIGCOMM* (2005)
8. Salem, O., Vaton, S., Gravey, A.: A scalable, efficient and informative approach for anomaly-based intrusion detection systems: theory and practice. *Int. J. Netw. Manag.* **20**, 271–293 (2010)
9. Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., Pepe, T.: When randomness improves the anomaly detection performance. In: *Proceedings of 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)* (2010)
10. Schwellen, R., Gupta, A., Parsons, E., Chen, Y.: Reversible sketches for efficient and accurate change detection over network data streams. In: *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*. IMC 2004, pp. 207–212. ACM, New York (2004)
11. Kind, A., Stoecklin, M.P., Dimitropoulos, X.: Histogram-based traffic anomaly detection. *IEEE Trans. Netw. Serv. Manag.* **6**(2), 110–121 (2009)
12. Brauckhoff, D., Dimitropoulos, X., Wagner, A., Salamati, K.: Anomaly extraction in backbone networks using association rules. *IEEE/ACM Trans. Netw.* **20**(6), 1788–1799 (2012)
13. Wagner, A., Plattner, B.: Entropy based worm and anomaly detection in fast IP networks. In: *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE 2005)*, pp. 172–177, June 2005
14. Callegari, C., Giordano, S., Pagano, M.: On the use of compression algorithms for network anomaly detection. In: *2009 IEEE International Conference on Communications*, pp. 1–5, June 2009
15. Lakhina, A.: Diagnosing network-wide traffic anomalies. In: *ACM SIGCOMM*, pp. 219–230 (2004)
16. Shannon, C.E., Weaver, W.: *The Mathematical Theory of Communication*. University of Illinois Press, Champaign (1949)

17. Kolmogorov, A., Fomin, S.: Elements of the Theory of Functions and Functional Analysis. Number v. 1 in Dover Books on Mathematics. Dover (1999)
18. Flow-Tools Home Page. <http://www.ietf.org/rfc/rfc3954.txt>
19. MAWI Working Group Traffic Archive. <http://mawi.wide.ad.jp/mawi/>. Accessed Nov 2011
20. MAWILab. <http://www.fukuda-lab.org/mawilab/> Accessed Nov 2011
21. Fontugne, R., Borgnat, P., Abry, P., Fukuda, K.: MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In: ACM CoNEXT (2010)
22. Callegari, C., Casella, A., Giordano, S., Pagano, M., Pepe, T.: Sketch-based multidimensional IDS: a new approach for network anomaly detection. In: IEEE Conference on Communications and Network Security, CNS 2013, National Harbor, MD, USA, 14–16 October 2013, pp. 350–358 (2013)

Future Network Systems and Security  
Second International Conference, FNSS 2016, Paris,  
France, November 23-25, 2016, Proceedings  
Doss, R.; Piramuthu, S.; Zhou, W. (Eds.)  
2016, X, 195 p. 81 illus., Softcover  
ISBN: 978-3-319-48020-6