An Information-Theoretic Method for the Detection of Anomalies in Network Traffic

Christian Callegari[†‡], Stefano Giordano[†], and Michele Pagano[†]
[†] Dept. of Information Engineering, University of Pisa, Italy
[‡] RaSS National Laboratory – CNIT, Pisa, Italy
E-mail:{c.callegari, s.giordano, m.pagano}@iet.unipi.it

***Abstract*—Anomaly-based Intrusion Detection is a key research topic in network security due to its ability to face unknown attacks and new security threats. For this reason, many works on the topic have been proposed in the last decade. Nonetheless, an ultimate solution, able to provide a high detection rate with an acceptable false alarm rate, has still to be identified.**

In this paper we propose a novel intrusion detection system that performs anomaly detection by studying the variation in the entropy associated to the network traffic. To this aim, the traffic is first aggregated by means of random data structures (namely three-dimension reversible sketches) and then the entropy of different traffic descriptors is computed by using several definitions.

The experimental results obtained over the MAWILab dataset validate the system and demonstrate the effectiveness of our proposal for a proper set of entropy definitions.

***Index Terms*—Anomaly Detection, Information Theory, Shannon Entropy, Tsallis Entropy, Rényi Entropy, Kullback-Leibler Divergence, Jensen-Shannon Divergence, MAWILab**

I. Introduction

In recent years Internet has become the playground for providing sensitive services to an ever growing amount of end-users, most of them only partially aware of the risks deriving from information sharing on the net. In spite of the development of cryptographic primitives and their use in secure protocols, a major role in this evolutionary process will be played by Intrusion Detection Systems (IDSs), which should be able to protect legitimate users against malicious activities of any type.

In such a framework, while misuse-based IDSs represent a well established reality, anomaly-based IDSs are still a hot research topic, mainly for their ability in also detecting unknown attacks. Indeed, although many anomaly detection solutions have been proposed over the years, each approach has its own limitations (often related to the false alarm rate) and an ultimate solution has not been identified yet.

Among the different proposed approaches, a promising choice seems to be represented by those methods that rely on the estimation of the entropy associated to some traffic descriptor. Nonetheless, such methods are still far for being acceptable in real world scenarios, and several improvements have to be studied.

For this reason in this paper we propose a novel anomaly detection system that relies on the estimation of different kinds of entropy, associated to the descriptors of traffic aggregates, obtained through random data structures. It is worth noting that the proposed system significantly extends the work presented in [1], from which it inherits the general system architecture. Nonetheless, it is based on a slightly modified detection algorithm and thus offers different performance. In a nutshell, we propose an IDS which at first, for both addressing scalability issues and improving performance, aggregates network traffic by means of a revised three-dimensional version of the reversible sketches and then performs the actual anomaly detection by computing, according to several different definitions, the entropy associated to the most significant traffic descriptors.

In more detail, the main contributions of this paper are:

- three-dimensional reversible sketches: a modified version of the reversible sketches is proposed, to allow the storage of the histograms of the considered traffic feature
- definition of *random* histograms to overcome the limitations of "standard" histograms (as discussed in Section VI-B)
- use of sketches combined with entropy estimation: the use of sketches for random aggregating the network traffic permits on one side to achieve better performance with respect to "standard" aggregation schemes and on the other side to be robust against mimicry attacks that can be carried out against entropy-based anomaly detection systems (as discussed in Section IV)
- comparison of different kinds of entropy: an extensive evaluation of the most commonly used entropy definitions is carried out over real network traffic traces
- study of the impact of different traffic descriptors (namely received bytes, flows, or packets) on the system performance

To validate and to evaluate the effectiveness of the proposed system, an extensive evaluation phase has been carried out over the well-known MAWILab traffic traces.

The remainder of this paper is organized as follows: Section II discusses the related works, while Section III provides an overview of the theoretical background, focusing on the description of the different entropy definitions used in this work. Then, Section IV details the architecture of the proposed system. The dataset used for testing and validating our proposal is described in Section V and in Section VI we describe the experimental results. Finally, in Section VII we conclude the paper with some final remarks.

II. Related Work

Anomaly detection has been extensively studied over the past decade. Many different approaches have been applied to this problem in order to meet the ever-increasing demands. To provide context to our approach, we discuss here some of most notable works related to the application of entropy to the anomaly detection field and to the usage of sketches in such a framework.

The idea to use some entropy measurement in anomaly detection is not new, but in most cases just the classical Shannon entropy was taken into account. For instance, it has been applied in [2] to detect fast Internet worms taking into account the entropy contents (more precisely, the Kolmogorov complexity) of traffic parameters, such as IP addresses, and in [3] to detect anomalies in the network traffic running over TCP. In both works an upper bound of Shannon entropy has been estimated through the use of different state-of-the-art compressors. Some more recent works include [4], [5], and [6], where entropy-based anomaly detection methods have been applied to specific domains like cloud computing, android devices, and vehicular networks.

A different approach has been considered in [7], [8], [9], where Shannon entropy was used to "summarize" the distribution of specific traffic features to detect unusual traffic patterns. Starting from the principles of thermodynamics, in [10] entropy was used, together with energy and temperature, to model the baseline operating conditions of the network and reveal attacks.

In [11] several information theoretic measures (including Shannon entropy, conditional entropy and Kullback–Leibler divergence) have been considered and their specific use has been discussed defining a general formal framework for intrusion detection. The use of Tsallis entropy in intrusion detection has been proposed in [12], where it is also shown that the optimal value of the parameter q does not depend significantly on datasets and traffic patterns, while in [13] different values of q are considered, introducing the so-called Traffic Entropy Spectrum that permits to capture additional information on detected anomalies. Comparisons among Shannon, Tsallis and Rényi entropies are performed in [14] to identify the traffic features that are more relevant for detecting anomalies (but taking into account KDDCup99 dataset, which is hardly representative of nowadays traffic and attacks), as well as in [15], where the authors showed that it is possible to detect modern botnet-like malware based on the entropy of anomalous patterns.

It is worth mentioning that some general weaknesses of entropy-based approaches are highlighted in [16], [17], where "optimal camouflage" strategies are described. In our case, the combined effect of random aggregation and different kinds of entropy adds robustness to the method.

Finally, regarding sketches, even if they cannot be considered as a detection method, they have been used as a building block of several AD systems [18], [19], [20], [21], [22]. Indeed, the use of sketches corresponds to a random aggregation that "efficiently" reduces the dimension of the data (wrt other deterministic aggregations, such as according to input/output routers [23]); moreover, the use of reversible sketches [24] permits to trace back the flows responsible for the anomalies.

To the best of our knowledge, our proposal is original from the point of view of both combining sketches and entropy estimation and performing an extensive evaluation and comparison of different definitions of entropy in such a field. Moreover, also the three-dimensional reversible sketches represent an original contribution of the present paper.

III. Theoretical Background

In this section we recall some theoretical background, focusing at first on different definitions and concepts related to entropy measures, and then on the sketch data structures.

As far as entropy is concerned, taking into account the nature of traffic data under test, we will focus on discrete distributions with a finite number L of elements. Roughly speaking, we will compare two empirical distribution using *some kind of entropy* as a measure of their similarity. This can be done in two different ways: comparing the entropies of the two distributions or considering the relative entropy among them.

A. Shannon entropy

The most basic concept in information theory is the entropy of a random variable (RV) X (or its distribution), often called Shannon entropy [25]. Roughly speaking, it is a measure of the uncertainty (or variability) associated with the RV.

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

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

Then its Shannon entropy is defined as follows:

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

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, (1) coincides with the well-known Boltzman–Gibbs entropy in statistical mechanics.

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

According to the definition (1), Shannon entropy can be interpreted as the expectation of a particular function, known in the literature as self–information, which weights each p_l according to its logarithm. In many cases it could be useful to introduce a more general definition of entropy that provides additional information about the importance of specific events, for example outliers or rare events. In other words, an

additional parameter should be added in order to weight in a suitable way different parts of the distribution. This issue emerged in different frameworks, ranging from information theory and cybernetics to statistical mechanics and quantum physics; examples of such generalised entropies are the Tsallis and Rényi entropies, that reduce to the traditional Shannon entropy for a special value of their additional parameter, described in the following.

B. Tsallis entropy

Tsallis entropy (also known as Havrda–Charvát–Tsallis entropy, since it was originally proposed in [26] by Havrda and Charvát, although with a different prefactor, and then independently rediscovered by Tsallis [27]) is defined as

$$S_q(X) \;=\; \frac{1-\sum_{l=1}^{L} p_l^q}{q-1} \tag{2}$$

where $q \in \mathbb{R}$ is the nonextensivity parameter or entropic index and for $q \to 1$ the usual Boltzmann–Gibbs entropy is obtained.

It is easy to show that, as for Shannon entropy, $S_q(X) = 0$ in case of degenerate distribution and attains its maximum in case of maximum disorder (i.e., uniform distribution)

$$S_q^{\max} = \frac{1-L^{1-q}}{q-1}$$

Moreover, when $q-1$ (and hence q) assumes large positive values, $S_q(X)$ is more sensitive to events that occur often (corresponding to higher values of p_l), while for large negative q rare events contribute more.

Another interpretation of the parameter q is related to the non-additivity of Tsallis entropy [28]: if two systems A and B are independent (i.e., $p_{lm}^{A+B} = p_l^A \cdot p_m^B$), then

$$S_q(A+B) \;=\; S_q(A) + S_q(B) + (1-q) S_q(A) S_q(B)$$

and the parameter $1-q$ is a measure of the departure from additivity. The previous result is consistent with the extensivity of the Shannon entropy, which indeed is obtained when $q \to 1$.

C. Rényi entropy

Another generalization of (1) is given by the Rényi entropy of order α, where $\alpha \geq 0$ and $\alpha \neq 1$:

$$H_\alpha(X) \;=\; \frac{1}{1-\alpha} \log_2 \left(\sum_{l=1}^{L} p_l^\alpha \right) \tag{3}$$

which is, in general, a non-increasing function in α, apart from the case of uniform distribution, for which $H_\alpha(X) = \log_2 L \;\; \forall \alpha$.

Also in this case, the contributions due to the different events depend on the value of the exponent α. Indeed, as α approaches zero, (3) weighs all possible events more and more equally, regardless of their probabilities and in the limit for $\alpha \to 0$, the Rényi entropy is just the logarithm of the size of the support of X. On the contrary, as $\alpha \to \infty$, the Rényi entropy is increasingly determined by the events with highest probability.

In more detail, several special cases of the Rényi entropy are well–known in the literature:

- H_0 is the Hartley entropy of X:
$$H_0(X) = \log_2 L$$
- the limiting value of H_α as $\alpha \to 1$ is the standard Shannon entropy
$$H_1(X) = H(X) = -\sum_{l=1}^{L} p_l \log_2 p_l$$
as for the Tsallis entropy (apart from the different multiplicative constant)
- H_2 is the collision entropy, sometimes just called "Rényi entropy"
$$H_2(X) = -\log_2 \sum_{l=1}^{L} p_l^2 = -\log_2 \mathbb{P}(X=Y)$$
where X and Y are iid RVs
- as $\alpha \to \infty$, H_α converges to the min-entropy
$$H_\infty(X) = \min_l(-\log p_l) = -\log \max_l p_l$$
and indeed $H_\infty(X)$ is the *smallest* entropy measure in the family of Rényi entropies

Finally, note that Tsallys and Rényi entropies depend from the probability distribution through the same quantity and hence the following relation holds

$$(1-q) H_q(X) = \log_2 \left(1 + (1-q) S_q(X)\right)$$

and it can be shown that $H_q(X)$ is an increasing function of $S_q(X)$. However, they have different properties; for instance, unlike the Tsallis entropy, the Rényi entropy is extensive as the traditional Shannon entropy.

D. Kullback–Leibler divergence

The Kullback–Leibler divergence (KL), also known as information divergence, information gain, or relative entropy, is a "measure" of the difference between two probability distributions P and Q [29].

In case of discrete probability distributions, the KL divergence of Q from P is given by

$$D_{\mathrm{KL}}(P\|Q) = \sum_{l=1}^{L} p_l \, \log \frac{p_l}{q_l} \tag{4}$$

and it is defined only if $q_l = 0$ implies $p_l = 0 \;\forall l$ (absolute continuity).

From an information theory point of view, $D_{\mathrm{KL}}(P\|Q)$ is the amount of information lost when Q is used to approximate P; in other words, it measures the expected number of extra bits required to code samples from P using a code optimized for Q rather than the code optimized for P. It can be easily related to Shannon entropy; indeed, denoting by U the uniform distribution over the L values assumed by X, we have

$$H(X) = \log_2 L - D_{\mathrm{KL}}(P\|U)$$

It is easy to show that

$$D_{\mathrm{KL}}(P\|Q) \geq 0$$

and equality holds iff $P = Q$ almost everywhere, in accordance with the intuitive idea of distance between distributions; however, KL is not a metric in the space of probability distributions since it is not symmetric[1]

$$D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$$

and does not satisfy the triangle inequality.

It is worth mentioning that, as for Shannon entropy, the family of Rényi divergences provide generalizations of the KL. Indeed, the Rényi divergence of order α (or α-divergence) of a distribution P from a distribution Q is defined to be

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \left(\sum_{l=1}^{L} \frac{p_l^\alpha}{q_l^{\alpha-1}} \right)$$

when $0 < \alpha < \infty$ and $\alpha \neq 1$. As expected, the limit $\alpha \to 1$ gives the KL.

E. Jensen–Shannon divergence

The Jensen–Shannon divergence (JS) is another popular method of measuring the similarity between two probability distributions [30] and can be interpreted as a symmetrized and smoothed version of KL. It is defined by[2]

$$D_{\mathrm{JS}} = \tfrac{1}{2} D_{\mathrm{KL}}\left(P\|M\right) + \tfrac{1}{2} D_{\mathrm{KL}}\left(Q\|M\right) \tag{5}$$

where M is the average of the two distributions, i.e.

$$M = \frac{1}{2}(P + Q)$$

It can be shown that, using the base 2 logarithm, the JS is bounded by 1:

$$0 \leq D_{\mathrm{JS}}(P\|Q) \leq 1$$

F. Sketch

Sketches are a family of data structures that use the same underlying hashing scheme for summarising data. They differ in how they update hash buckets and use hashed data to derive estimates [19].

Specifically, the sketch data structure is a two-dimensional $D \times W$ array $T[d][w]$, where each row d ($d = 1, \ldots, D$) is associated with a given hash function h_d. These functions give an output in the interval $(1, \ldots, W)$ and these outputs are associated to the columns of the array. As an example, the element $T[d][w]$ is associated to the output value w of the d hash function.

The input data are viewed as a stream that arrives sequentially, item by item, according to the Turnstile Model [31]. Let $I = \sigma_1, \sigma_2, \ldots$ be the input stream, then each item $\sigma_k = (i_k, c_k)$ consists of a *key*, i_k (e.g., IP addresses, L4 ports), and a *weight*, c_k (e.g., number of bytes or packets in a flow). When new data arrive, the sketch is updated as follows:

$$T[d][h_d(i_k)] \leftarrow T[d][h_d(i_k)] + c_k \tag{6}$$

[1]Kullback and Leibler themselves actually defined the divergence as $D_{\mathrm{KL}}(P\|Q) + D_{\mathrm{KL}}(Q\|P)$, which is symmetric

[2]Note that JS can be generalized for the comparison of more than two distributions, but this goes beyond the goal of our theoretical background

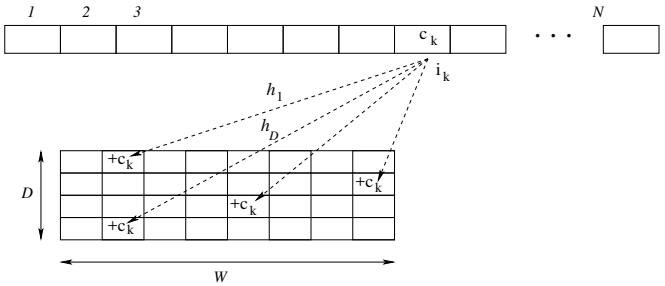


Figure 1: Sketch: Update Function

The update procedure is realised for all the different hash functions as shown in Figure 1.

To be noted that, given the use of hash functions, it is possible to have some *collisions* in the sketch table. In this work we have taken advantage of this fact, indeed having collisions allows us to randomly aggregate the traffic flows, namely all the IP flows that collide in the same bucket will be considered as an aggregate.

From the anomaly detection point of view, aggregation performed by means of probabilistic data structures as sketches has proven to lead to better performance with respect to "classical" aggregation strategies (e.g., ingress/egress router) [23].

However, sketch data structures have a major drawback: they are not reversible. That is, a sketch cannot efficiently report the set of all keys that correspond to a given bucket of the sketch.

To overcome such a limitation, [32] proposes a novel algorithm for efficiently reversing sketches, focusing primarily on the k-ary sketch. The basic idea is to hash "intelligently" by modifying the input keys and/or hashing functions so as to make possible to recover the keys with certain properties like big changes without sacrificing the detection accuracy.

In more detail the update procedure for the k-ary sketch is modified by introducing modular hashing and IP mangling techniques.

The modular hashing works partitioning the n-bit long hash key x into q words of equal length n/q, that are hashed separately using different hash functions, h_{di} ($i = (1, \ldots, q)$). Let us consider that the output of each function is m-bit long. Finally, these outputs are concatenated to form the final hash value (as depicted in Figure 2).

$$\delta_d(x) = h_{d1}(x)|h_{d2}(x)|\ldots|h_{dq}(x) \tag{7}$$

Since the final hash value consists of $q \times m$ bits, it can assume $W = 2^{q \times m}$ different values.

Note that the use of the modular hashing can cause a highly skewed distribution of the hash outputs. Consider, as an example, our case in which IP addresses are used as hash keys. In network traffic streams there are strong spatial localities in the IP addresses since many IP addresses share the same prefix. This means that the first octets (equal in most addresses) will be mapped into the some hash values increasing the collision probability of such addresses.

To effectively resolve this problem, the *IP mangling* technique has to be applied before computing the hash functions.

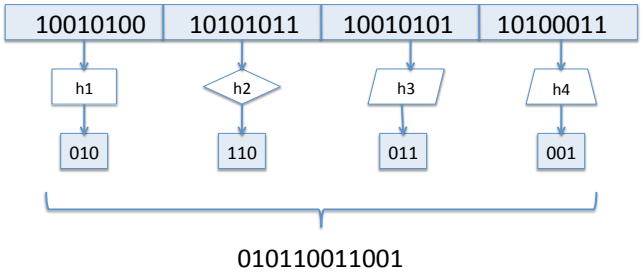


Figure 2: Modular Hashing

By using such technique the system randomizes, in a reversible way, the input data so as to remove the correlation or spatial locality.

The other key point introduced in [32] is the algorithm for reversing the sketch, named *reverse hashing*, which exploits the properties of modular hashing and IP mangling. For the sake of brevity, we skip the discussion of this algorithm, referring the reader to [32] for all the details.

IV. System Architecture

In this section we describe the architecture of the proposed system (depicted in Figure 3), detailing the functionalities of each system block.

A. System Input

First of all the input data are processed by a module called, in Figure 3, System Input, which is responsible of parsing the input data so as to extract the proper traffic features. In more detail, this module is responsible of reading the network traffic (e.g., NetFlow traces [33], pcap [34]) and of parsing it (e.g., by using the Flow-Tools [35], in case of NetFlow data), so as to produce plain ASCII files containing the input data. In our implementation, such data are formatted according to the most general data streaming model, that is the Turnstile Model (see Section III).

As already mentioned, according to this model, the input data are viewed as a stream that arrives sequentially, item by item. Let $I = \sigma_1, \sigma_2, \dots$ be the input stream, then each item $\sigma_k = (i_k, c_k)$ consists of a *key*, i_k, and a *weight*, c_k.

In the Turnstile model, the arrival of a new data item causes the update of an underlying function $U[i_k] = U[i_k] + c_k$, which represents the sum of the *weights* of a given *key* over the time.

This model is very general and can be used in different scenarios. As an example, in the context of network anomaly detection, the key can be defined using one or more fields of the packet header (IP addresses, L4 ports), or entities (like network prefixes or AS number) to achieve a higher level of aggregation, while the underlying function can then be the total number of bytes or packets in a flow, or flows per OD pair.

In our system, yet maintaining the idea of the underlying function U and of the total sum, such functions are in fact realised in the subsequent module, responsible for the construction of the sketches, as described in the following subsection.

From the practical perspective, in our implementation we have in input pcap data, measuring the traffic gone through a given router, collected over fifteen minutes time-bins. Thus, this module will output a distinct file for each considered time-bin (let us assume we have N distinct time-bins), each file containing a list of keys observed in that time-bin (e.g., the list of destination IP addresses) and the associated weights (e.g., the number of bytes received by that IP address).

Note that the modularity of the system allows great flexibility. Indeed, the system administrator can easily choose which traffic descriptor has to be used to better allows her to detect the different attacks.

B. Sketch Computation

After the data have been correctly formatted, they are passed as inputs to the module responsible for the construction of the reversible sketch tables.

Hence, referring to Figure 3, the block "Hashing H^1" is responsible for the construction of the reversible sketches (as already stated, each file, corresponding to a distinct time-bin, is used to build a distinct sketch).

As far as the hash functions are concerned, we have used 4-universal hashes[3] [36], obtained as:

$$h(x) = \sum_{i=0}^{3} a_i \cdot x^i \ mod\ p\ mod\ W \tag{8}$$

where the coefficients a_i are randomly chosen in the set $[0, p-1]$ and p is an arbitrary prime number (we have considered the Mersenne numbers).

Note that, given that in our system we need to compute the entropy associated to a given traffic aggregate, maintaining a simple counter in each bucket of the sketch is not enough. Hence, instead of having a "standard" two-dimensional array, as described so far, in our system we have implemented a novel 3D data structures $T[d][w][l]$, in which the third dimension is used to store histograms.

In more detail, a second hashing scheme, H^2 in the figure, independent of the first one and still realised with 4-universal hash functions, is used to map each input weight c_k to a given histogram bin. We considered L bins for each traffic descriptor and selected the bin associated to each key i_k as the hash of the corresponding weight c_k, realising a *random* histogram (the impact of using *random* histograms will be discussed in Section VI-B).

Formally, for each new data, the update procedure of the sketch is described by

$$T[d][h_d^1(i_k)][h_d^2(c_k)] \leftarrow T[d][h_d^1(i_k)][h_d^2(c_k)] + 1 \tag{9}$$

Note that in our implementation, both the hashing schemes H^1 and H^2 are given by D distinct hash functions, which

[3] A class of hash functions $H : (1, \dots, N) \rightarrow (1, \dots, W)$ is a *k-universal hash* if for any distinct $x_1, \dots x_k \in (1, \dots, N)$ and any possible $v_1, \dots v_k \in (1, \dots, W)$:

$$P(h(x_i) = v_i; \forall i \in (1, \dots, k)) = \frac{1}{W^k}$$

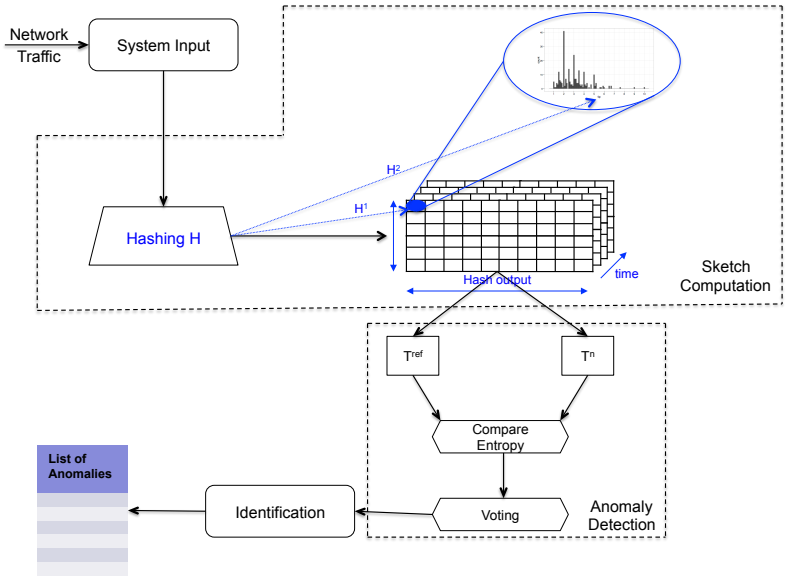


Figure 3: System Architecture

give output in the interval $[1; W]$ and $[1; L]$, respectively. This results in sketches that are $\in \mathbb{N}_{D\times W\times L}$, where D, W, and L can be varied.

At this point, given that we had N distinct time-bins, we have obtained N distinct sketches $T^n_{D\times W\times L}$, where $n \in [1, N]$ is the sequence number of the time-bin.

It is important to highlight that, apart from the effectiveness of performing a random aggregation, with respect to "classical" aggregation techniques – as already discussed – the use of sketches also has two additional advantages. First of all, it allows to maintain low and constant the system complexity when the amount of traffic changes. Indeed the sketch dimensions do not depend on the quantity of processed traffic. Second, as described in [16], [17], entropy does not always allow us to discriminate two (also very different) histograms (as an example, think of two histograms that are scrambled versions of the same histogram). Hence, an attacker could realise a "mimicry" attack, in which after having estimated the traffic distribution, it creates an attack such that the associated histogram, yet very different from the reference one, leads to the same (or very similar) entropy value (as discussed in [17]). In our case, given that the hashing scheme used to construct the sketch introduces some randomness (and it is in general unknown), such an attack is unfeasible.

C. Anomaly Detection

Once the sketches have been constructed they are passed in input to the block that is responsible for the actual anomaly detection phase. As depicted in Figure 3, two distinct sketches are considered in such a block: the reference sketch T^{ref}, which is the last observed non anomalous sketch, and the current sketch T^n.

At this point the system performs one of the following actions:

- compute and compare the entropy: for each bucket of the current sketch $T^n[d][w][\cdot]$ the system computes the entropy (by using one of the entropy definitions provided in Section III) associated to the stored histogram and computes the difference between such a value and the entropy associated to the same bucket in the reference sketch
- compute the "distance" between the sketches: for each bucket of the current sketch $T^n[d][w][\cdot]$ the system computes the "distance" (by using one of the divergence definitions provided in Section III) between the stored histogram and the correspondent histogram of the reference sketch

Thus such a value (either the entropy difference or the distance) is compared with a threshold to decide if there is an anomaly or not.

The output of this phase is a binary matrix ($A \in \mathbb{N}_{D\times W}$), for each time-bin, that contains a "1" if the corresponding sketch bucket is considered anomalous at that time-bin , "0" otherwise.

Note that, given the nature of the sketches, each traffic flow is part of several random aggregates (namely D aggregates), corresponding to the D different hash functions. This means that, in practice, any flow will be checked D times to verify if it presents any anomaly (this is done because 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 to the matrix A. The algorithm simply verifies if at least H rows of A contain at least a bucket set to "1" (H is a tunable parameter, set to $\lceil D/2 \rceil$ in the experimental section). If so, the system reveals an anomaly, otherwise the matrix A is discarded.

D. Identification

In case the voting procedure outputs the presence of an anomaly in a given time-bin, the system applies the reversible sketch algorithm to the sketch table in that time-bin for identifying IP addresses responsible for the anomalies.

Hence the output is represented by a list of anomalies and responsible flows.

V. Dataset: MAWI Traffic Traces

Our system has been extensively tested and evaluated using the traffic traces of the MAWI (Measurement and Analysis on the WIDE Internet) dataset [37], which consists of packet traces from the MAWI archive (sample-points B and F), publicly available at [38]. Each trace in this database is a pcap file containing the traffic captured over 15 minutes in a specific day, since 2001 to now, 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 are essential for evaluating new approaches. Although also for the MAWI archive, an exact description of the attacks is not available, the data set presents two important features that made it suitable for the performance evaluation procedure. First of all the traffic mixture is representative of the current mixtures of network services and applications, being collected in a real operating network, and then, in the framework of the successive project MAWILab [37], 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.

In more detail, the traces classification has been obtained combining the output of four anomaly detectors [39]. 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.

The most widely used performance indicators are the ROC curve, which plots the detection probability P_D against the false alarm probability P_{FA}, and the associated AuC (Area under the Curve), when varying the detection threshold. 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 [40], 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 detector": the number of unrevealed flows labeled as "anomalous" and detected at least by two of the four detectors used in MAWI classification
- "fn 3 detector": the number of unrevealed flows labeled as "anomalous" and detected at least by three of the four detectors used in MAWI classification
- "fn 4 detector": the number of unrevealed flows labeled as "anomalous" and detected by all 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)
- "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.

VI. Performance Evaluation

In this section we describe the experimental results obtained testing our system over the MAWILab dataset. It is important to highlight that, since we have focused on volume anomalies, we have taken into consideration, as traffic descriptors, the number of flows with the same destination IP address (hence referring to the Turnstile model presented in Section IV, for each item the key is given by the destination IP address and the weight is given by the number of associated flows) and the quantity of traffic received by each IP address expressed either in bytes or in packets (again referring to the Turnstile model, for each item the key is still given by the destination IP address, while the weight is given by the number of associated bytes or packets, respectively).

In the first three subsections, we present some preliminary analysis results aimed at : i) understanding if taking into consideration different traffic descriptors and different definitions of the entropy takes to different system performance, ii) comparing random hisogram to "standard" ones, and iii)

correctly dimensioning the system. Then, the subsequent three subsections detail the results achieved in terms of ROC curve and AuC, when taking into consideration bytes, flows, and packets.

A. Preliminary Analysis: Entropy Analysis

Before actually discussing the system performance, we have carried out several experimental tests to evaluate the differences when applying our method to different traffic descriptors and using different definitions of entropy (namely Shannon – H, Rényi with parameter α – H_α, and Tsallis with parameter q – S_q).

Figure 4 represents the scatter plot of H computed over the same traffic aggregates, when taking into consideration flows and bytes (Figure 4a), packets and bytes (Figure 4b), and packets and flows (Figure 4c). In more detail, we consider one row of the sketch and each point represents one bucket for the chosen time-bin and its coordinates are given by the values of the entropy associated to the histogram of the first descriptor (x axis) and second one (y axis). In this preliminary comparison about the information associated to different features, we have used the Shannon entropy, being the most "classical" definition.

The basic idea is that two variables that present, in the scatter plot, a linear pattern should take to the same system performance. It is important to highlight that very different performance can be offered also by strongly correlated variables provided that they do not have a linear scatter plot.

As it can be clearly seen from the figures, the scatter plots show that the couples of variables do not have a clear linear relationship. This result indicates, as will be confirmed in the next subsections, that the three different traffic descriptors can lead to completely different results.

The subsequent Figure 5 presents the scatter plot computed between the different definition of entropy (H and H_3 in Figure 5a, H and S_3 in Figure 5b, and H_3 and S_3 in Figure 5c), so as to evaluate if, as expected from the theory, using different definitions of entropy actually makes sense in the anomaly detection field. Note that, for sake of brevity and given that in these graphs we do not want to evaluate the impact of the parameters α and q, we have arbitrarily set them to a value equal to 3. Also in these graphs, the lack of *linearity* in the plots is an indication that the use of the three different families of entropy should take to different performances.

Moreover, from the scatter plots, we can infer an additional information: a monotonic curve indicates that the order of the buckets (if ranked according to the associated entropy) does not change between the two used definitions. Hence, from the practical point of view, this means that the ranking of the buckets remains the same when using Tsallis and Renyi (as expected from the theory), while it changes if using Shannon and Renyi or Tsallis. This gives us a further indication about the fact that different entropy definitions will take to different system performance.

A similar conclusion can be drawn observing Figure 6 and Figure 7, where we show the scatter plot between H_2 and H_4, and S_2 and S_4, to evaluate if different values of parameters α and q are worth being studied.

Statistics	Byte	Flow	Packet
min	46	1	1
Max	131524478	3941	161619
50th Percentile	122	1	1
75th Percentile	382	2	4
95th Percentile	2178	4	15
99th Percentile	52474	18	18

Table I: Features Statistics

B. Preliminary Analysis: Random Histogram

In this second analysis, some experimental tests have been carried out to evaluate the usage of the hashing scheme H^2 to realise the *random* histograms stored in each bucket of the sketch. Indeed, a more straightforward choice would be to realise "standard" histogram, simply dividing the samples of interest in L distinct equal bins (in this analysis we use $L = 64$ for sake of clarity, the impact of L will be discussed in the next subsection). Nonetheless, simply realising the histograms in such a way we would obtain too dense histograms (in Figure 8 we show as an example the histogram computed over the Byte feature contained in one of the MAWILab traces, using Shannon entropy). This is due to the distribution of the considered traffic features, whose main statistics are shown in Table I.

Moreover, only statistics related to the training set (i.e., in our case the time-bin used to build the reference sketch) may be available in advance, so even a non-uniform quantization does not solve the problem.

Instead, using the hashing scheme H^2 we obtain *random* histograms, which are much more "sparse" and hence more significant for our purpose (in Figure 9 we show the *random* histogram computed over the same values used in Figure 8 and the same number L of bins).

C. Preliminary Analysis: System Dimensioning

This third analysis has been carried out to correctly dimension the main structures used in the systems (namely the sketch). As far as the number of rows (D) and columns (W) are concerned, based on previous works (e.g., [40]), they have been set to $D = 16$ and $W = 512$.

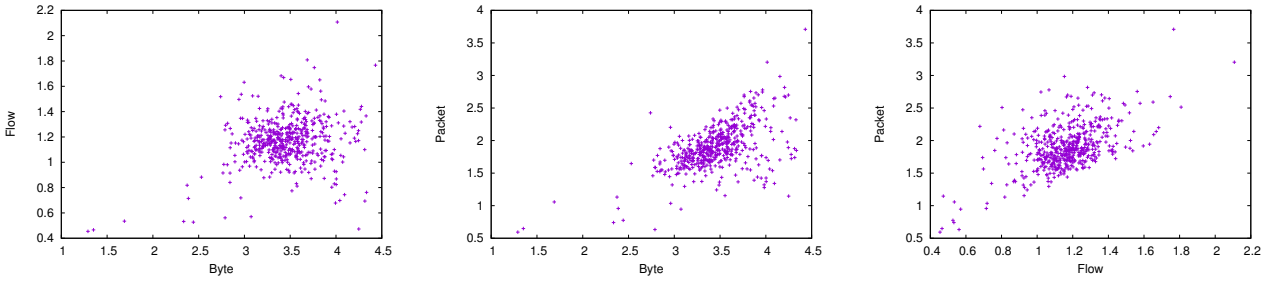
Instead, regarding the number of histogram bins (L), we have carried out some experimental tests. Figure 10 shows the ROC curves achieved when varying L in the range $\{32, 64, 96, 128\}$ (for sake of simplicity we only report the results achieved when using Shannon Entropy and byte as traffic feature, but comparable results have been achieved with other "configurations").

As it clearly appears, the best results are obtained when $L = 96$.

D. Experimental Results: Byte

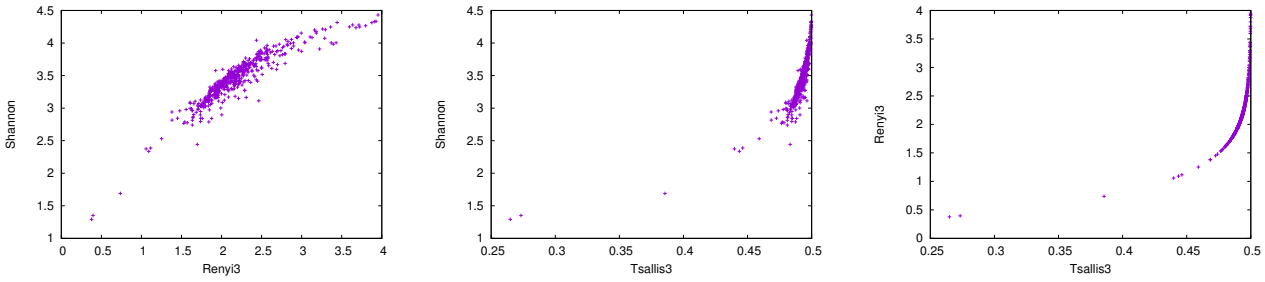
In this subsection we show the actual performance, in terms of detection probability P_D and false alarm probability P_{FA}, of our system, when taking in consideration the bytes.

First of all, in Figure 11 we present the performance with Shannon entropy when varying the interpretation of the



(a) Flow vs. Byte (b) Packet vs. Byte (c) Packet vs. Flow

Figure 4: Scatter Plot: different descriptors



(a) H vs. H_3 (b) H vs. S_3 (c) H_3 vs. S_3

Figure 5: Scatter Plot: different definitions of entropy

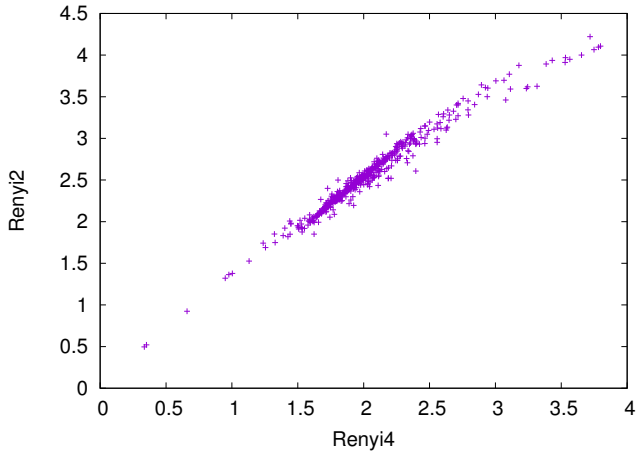


Figure 6: Scatter Plot: H_2 vs. H_4

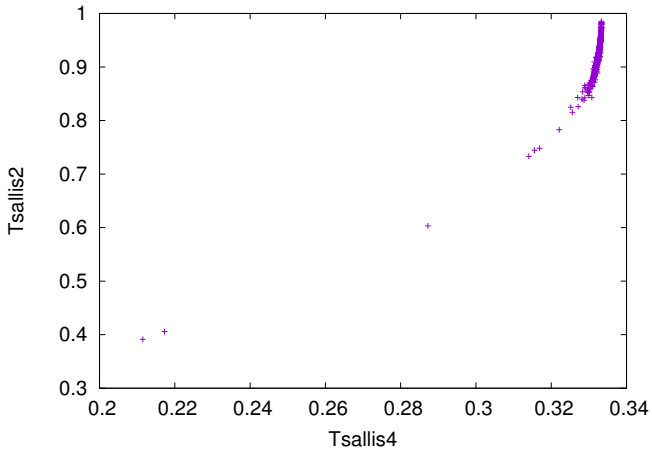


Figure 7: Scatter Plot: S_2 vs. S_4

MAWILab labels, as discussed in Section V. The plots clearly show that the system performance are strongly influenced by the different interpretation of the labels, going from unacceptable (e.g., "fn attack") to very good (e.g., "fn unknown 4 detector"). This variability, that is already known in the literature [40], can be justified by the very little number of flows belonging to some categories (e.g., "fn attack") compared with the total number of anomalies. In any case, the lack of effectiveness in detecting known attacks is not a major concern for the applicability of the anomaly detection systems. Indeed, anomaly-based IDSs are typically used in conjunction with misuse-based systems, which are effective in revealing known attacks, but are unable to find the unknown ones (for which the signatures are not present yet!). The latter are instead well detected by the proposed algorithms.

Given that, in the following we will only focus on the performance for the "fn unknown 4 detector" case.

Figures 12, 13, and 14 respectively present the ROC curves when using "standard" entropy definitions, H_α, and S_q. We can notice that among Shannon entropy, Kullback-Leibler divergence, and Jensen-Shannon divergence, only Shannon entropy takes to good results, while both KL and JS lead to unacceptable results, being all the ROC curves in Figure 12 close (if not even below) the diagonal. Instead, using H_α the system offer good performance, as demonstrated in Figure 13, but the performance strongly improve when using S_q. Indeed

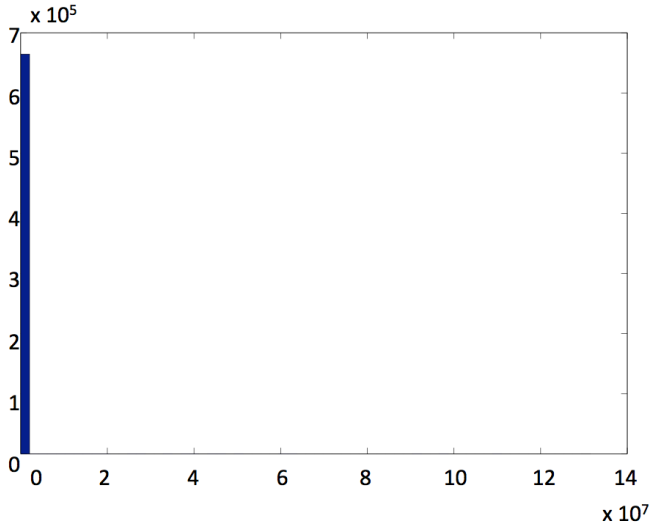


Figure 8: "Standard" Histogram (Byte, 64 bins)

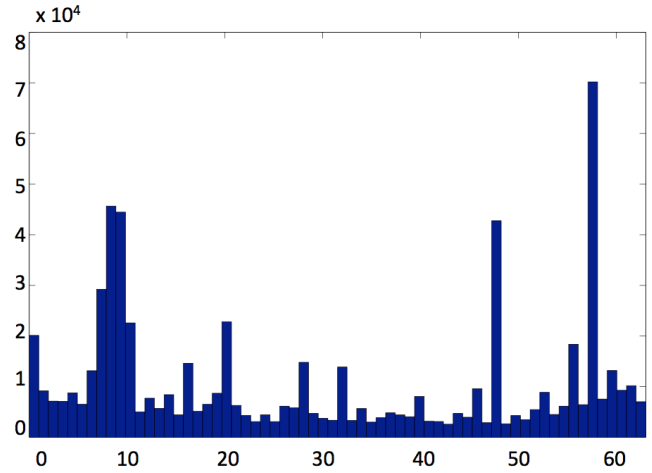


Figure 9: Random Histogram (Byte, 64 bins)

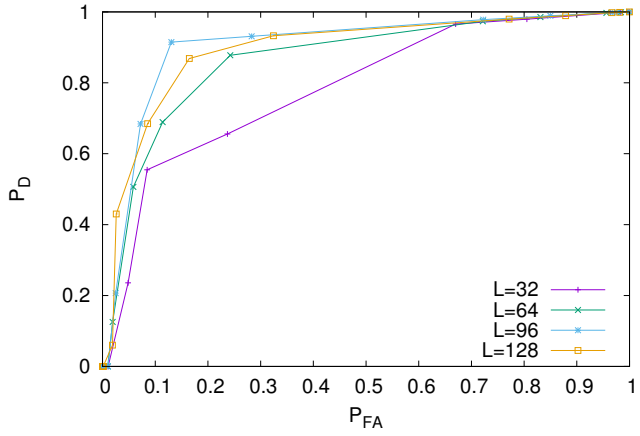


Figure 10: ROC Curve: Varying L

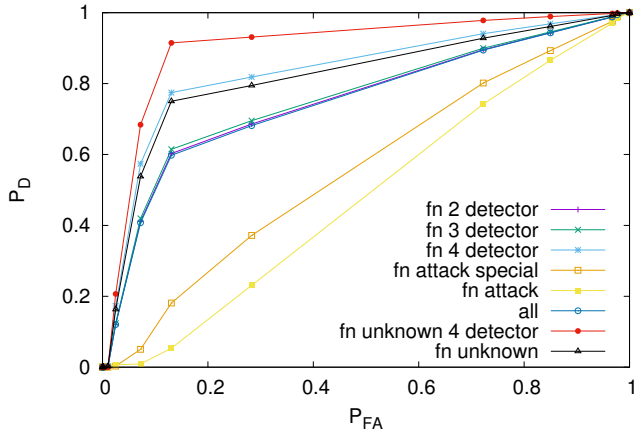


Figure 11: ROC Curve: H with different interpretations of the MAWILab labels (Byte)

from Figure 14, we can see that the system achieves very good performance, especially when $q = 6$ (additional results, not shown for sake of brevity, indicate that the performance do not significantly improve further increasing the value of q). This result is justified by the fact that Tsallis entropy better characterises highly probable events when q is positive and "big" (see Section III for the theoretical discussion), and a volume anomaly can be considered as a highly probable event if we consider the histogram representing the number of received flows. It is worth noting that, because of this aspect, we have not taken into consideration $q < 0$.

All the results are summarised in Table II, where, for sake of completeness, we report the values of the AuC, for all the discussed methods and for all of the different MAWILab labels. From the table we can easily conclude that the best performance (over the "fn unknown 4" label), for each considered "family" of entropy definition, are obtained by H, H_2, and S_6, which are further compared, in terms of ROC curve, in Figure 15.

In conclusion, for the byte case, we can see that the system is able to achieve very good results ($P_D > 90\%$ with $P_{FA} < 10\%$)

E. Experimental Results: Flow

Similarly to what done in the previous subsection (and taking into account the obtained results), we present the system performance in the case "fn unknown 4 detector", when taking into consideration the number of flows as traffic descriptor.

Figure 16 presents the ROC curves obtained when using "standard" entropy definitions. Such plots clearly indicate that the best performance are obtained when using H, while the use of both KL and JS takes to unacceptable performance, being the ROC curve very close to the diagonal.

Instead, Figure 17 presents the ROC curves obtained when using H_α and varying the value of α between 2 and 4. By observing the graphs, it is clear that α does not have a strong impact on the system performance, and that H_2 behaves slightly better than the other cases.

Similarly, Figure 18 presents the ROC curves obtained when using S_q and varying the q parameter. In this case the parameter has been varied between 2 and 6, with the best

Method	AuC							
–	all	fn 2	fn 3	fn 4	attack	attack sp.	unknown	unknown 4
H	0.749288	0.751888	0.758025	0.835591	0.4803	0.557955	0.819242	0.90413
KL	0.522163	0.522252	0.521702	0.522083	0.551005	0.510057	0.526626	0.531105
JS	0.521666	0.521904	0.521302	0.526757	0.548744	0.494368	0.531675	0.543652
$\mathbf{H}_2$	0.674507	0.676392	0.68072	0.738476	0.482242	0.541467	0.723197	0.790217
$\mathbf{H}_3$	0.634436	0.635968	0.639801	0.697944	0.459855	0.498696	0.684096	0.745809
$\mathbf{H}_4$	0.594717	0.595974	0.599049	0.641632	0.451493	0.496209	0.630735	0.679397
$\mathbf{S}_2$	0.716298	0.718115	0.722697	0.794849	0.518546	0.549366	0.777345	0.855448
$\mathbf{S}_3$	0.706611	0.708557	0.713177	0.809879	0.502209	0.466369	0.794499	0.878781
$\mathbf{S}_4$	0.701317	0.702265	0.704969	0.781248	0.547067	0.509702	0.77139	0.840795
$\mathbf{S}_5$	0.742962	0.743929	0.747218	0.844888	0.57774	0.50111	0.831412	0.91621
$\mathbf{S}_6$	0.744033	0.74574	0.750237	0.848218	0.539046	0.49014	0.83692	0.922063

Table II: AuC Values (Byte)

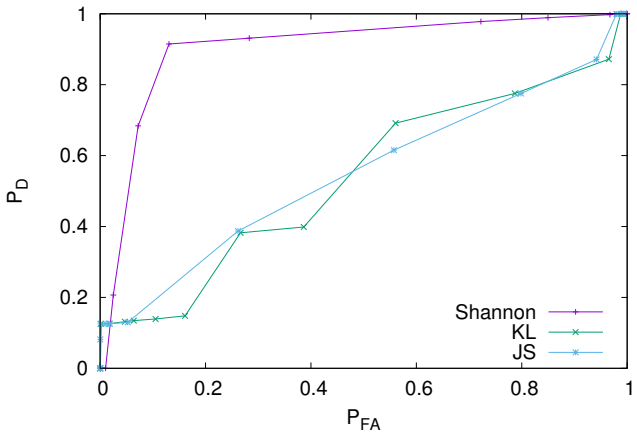


Figure 12: ROC Curve: "standard" entropy (Byte)

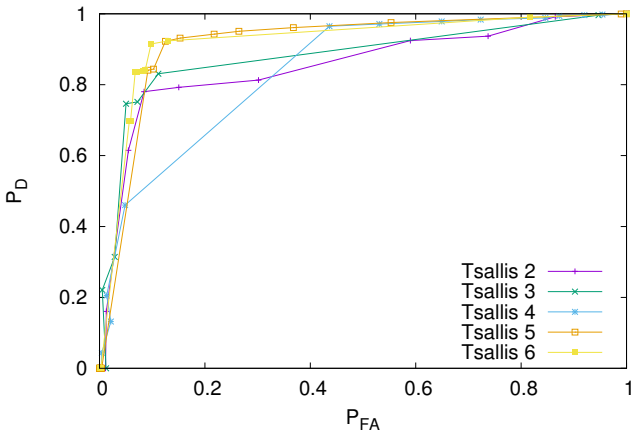


Figure 14: ROC Curve: S_q (Byte)

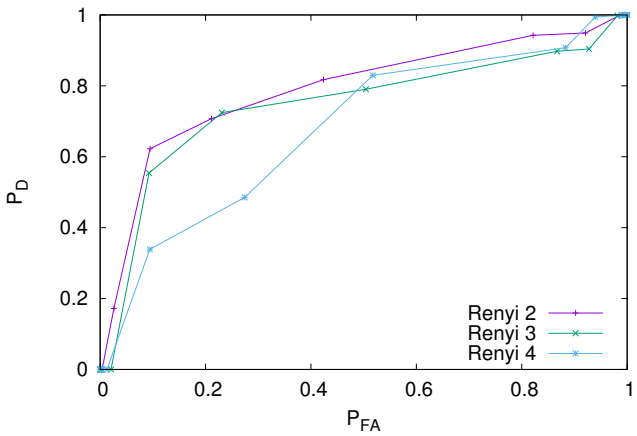


Figure 13: ROC Curve: H_α (Byte)

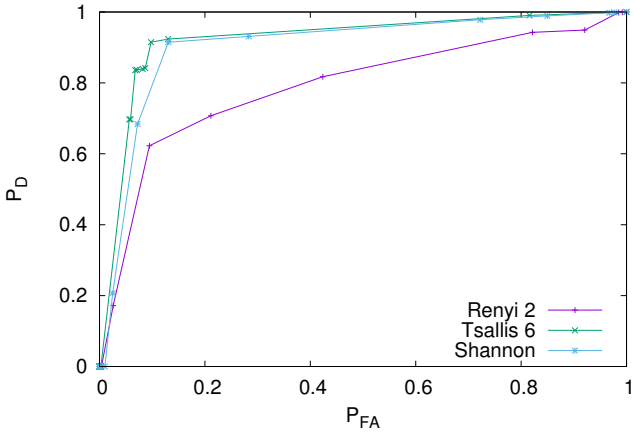


Figure 15: ROC Curve: best cases (Byte)

performance offered when $q = 6$. This fact is justified, as previously discussed, by the fact that Tsallis entropy better characterises highly probable events when q is positive and "big".

To allow an easier comparison among the different methods, in Table III we report, for each of the considered system settings, the value of the AuC. From the table it appears that the best performance, for considered "family" of entropy definition, are obtained by H, H_2, and S_6, which are further compared in terms of ROC curve, in Figure 19.

It is important to highlight that the system, with its best settings (i.e., S_6), is able to achieve very good results with more of 85% of detection in correspondence of P_{FA} around 15%.

F. Experimental Results: Packet

Similarly to what done in the previous subsections, we present the system performance when taking into consideration

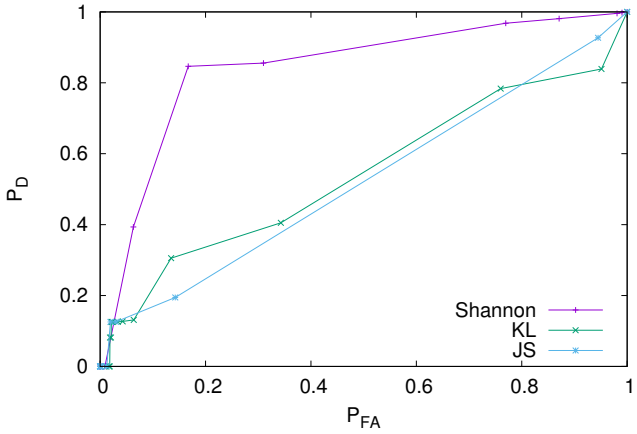


Figure 16: ROC Curve: "standard" entropy (Flow)

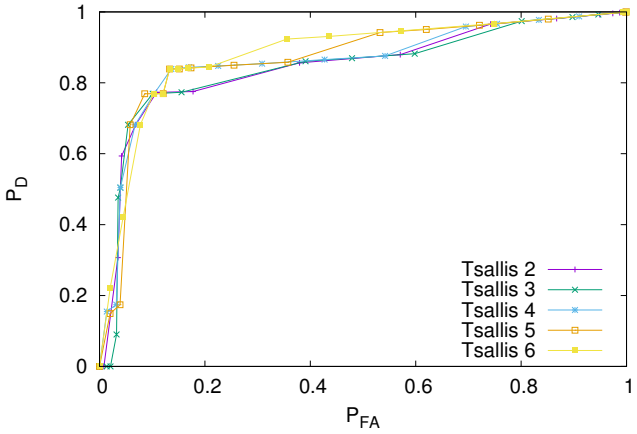


Figure 18: ROC Curve: S_q (Flow)

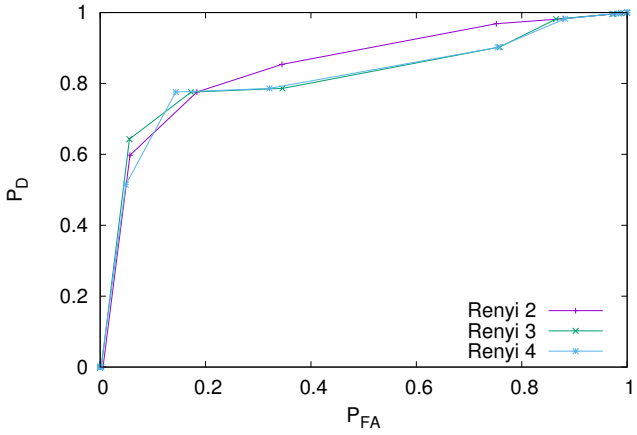


Figure 17: ROC Curve: H_α (Flow)

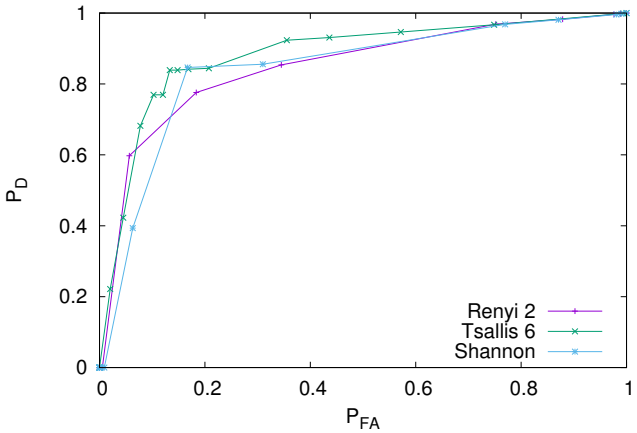


Figure 19: ROC Curve: best cases (Flow)

the packets as traffic descriptor.

Figure 20 presents the ROC curves obtained when using "standard" entropy definitions, and, as in he previous cases, the only acceptable performance are offered by Shannon entropy.

Then, Figures 21 and 22 present the ROC curves obtained when using H_α (varying the value of α between 2 and 4) and S_q (the value of q between 2 and 6). By observing the graphs, it is clear that neither α nor q have a strong impact on the system performance, and that H_2 and S_5 behaves slightly better than the other cases in each respective "entropy family".

As for the other cases, in Table IV we report all of the values of the AuC. From the table we can easily see that the best performance are offered by Shannon, H_2, and S_5, which are further compared in Figure 23

It is important to notice that, also in this case, the system is able to offer very good performance with more of 90% of detection in correspondence of P_{FA} around 5% (S_5 case).

Method	AuC
H	0.842355
KL	0.542945
JS	0.522932
H_2	0.848728
H_3	0.817902
H_4	0.814845
S_2	0.848961
S_3	0.845637
S_4	0.862569
S_5	0.870584
S_6	0.882147

Table III: AuC Values (Flow)

Method	AuC
H	0.918374
KL	0.543908
JS	0.464775
H_2	0.897934
H_3	0.88578
H_4	0.893729
S_2	0.929691
S_3	0.923326
S_4	0.931093
S_5	0.931339
S_6	0.927684

Table IV: AuC Values (Packet)

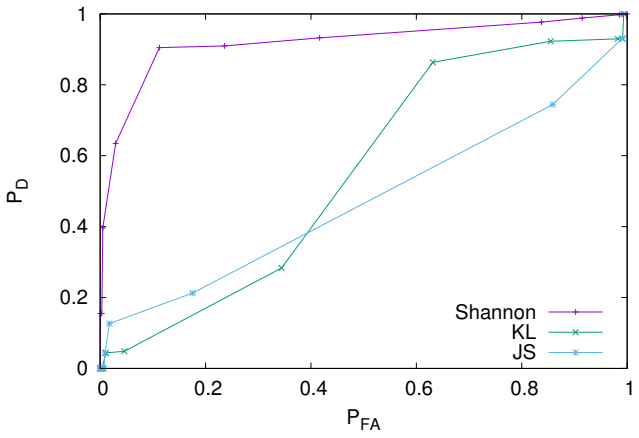


Figure 20: ROC Curve: "standard" entropy (Packet)

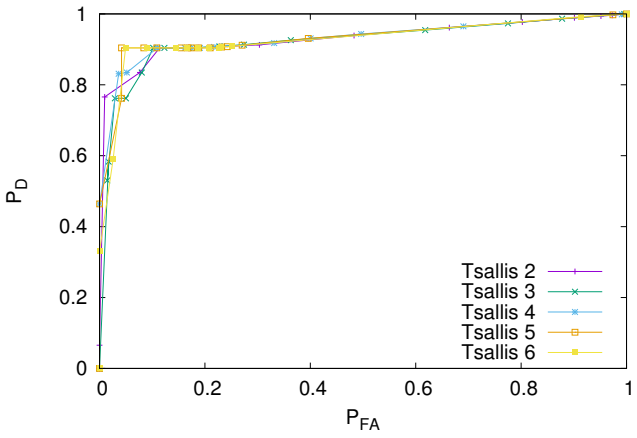


Figure 22: ROC Curve: S_q (Packet)

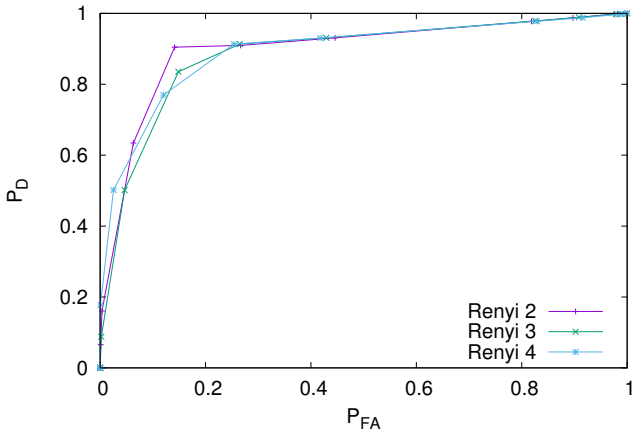


Figure 21: ROC Curve: H_α (Packet)

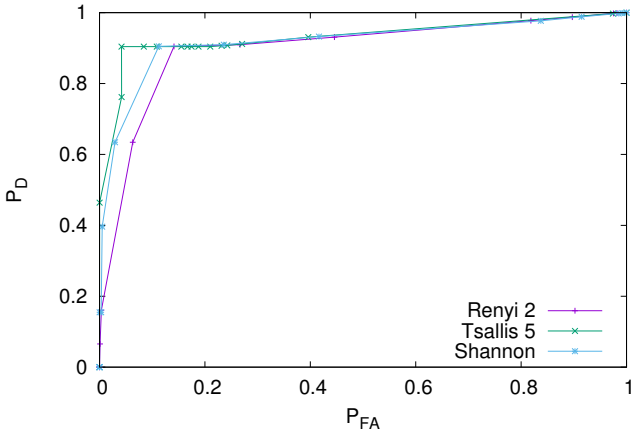


Figure 23: ROC Curve: best cases (Packet)

VII. Conclusions

In this paper we have presented a novel anomaly detection system that leverages on the estimation of the different kinds of entropy, associated to the descriptors of traffic aggregates, obtained through random data structures. In more detail, the system at first relies on a modified version of the reversible sketches (extended to the three-dimensional case) to randomly aggregate the traffic, so as to simultaneously address scalability issues and improve the detection performance.

Then it computes the entropy associated to one of the different traffic descriptors by using different measures of entropy (namely Shannon, Kullback-Leibler, Jensen-Shannon, Rényi, and Tsallis).

The extensive evaluation phase, carried over the MAWILab traffic traces, has demonstrated that the use of entropy definitions different from the "standard" Shannon entropy takes to improve the system performance. In more detail, the experimental results suggest that Tsallis entropy (with a relatively high value of q, i.e., $q = 5, q = 6$) offers the best performance. Indeed, in those cases, the system is able to achieve very good results with more than 90% of detection in correspondence of a false alarm probability around 5%. Finally, it is worth noticing that the offered performance are almost independent of the chosen traffic descriptor.

Acknowledgment

This work was partially supported by Multitech SeCurity system for intercOnnected space control groUnd staTions (SCOUT), a research project supported by the FP7 programme of the European Community.

References

[1] C. Callegari, S. Giordano, and M. Pagano, "Entropy-based network anomaly detection," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2017, pp. 334–340.

[2] A. Wagner and B. Plattner, "Entropy based worm and anomaly detection in fast ip networks," in *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*, June 2005, pp. 172–177.

[3] C. Callegari, S. Giordano, and M. Pagano, "On the use of compression algorithms for network anomaly detection," in *2009 IEEE International Conference on Communications*, June 2009, pp. 1–5.

[4] A. S. S. Navaz, V. Sangeetha, and C. Prabhadevi, "Entropy based anomaly detection system to prevent ddos attacks in cloud." *CoRR*, vol. abs/1308.6745, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1308.html#NavazSP13

[5] F. Ghaffari and M. Abadi, "Droidmalhunter: A novel entropy-based anomaly detection system to detect malicious android applications," in *Computer and Knowledge Engineering (ICCKE), 2015 5th International Conference on*, Oct 2015, pp. 301–306.

[6] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *IEEE 2nd International Forum on Research and Technologies for Society and Industry (RTSI 2016)*, 2016.

[7] A. Lakhina, "Diagnosing network-wide traffic anomalies," in *In ACM SIGCOMM*, 2004, pp. 219–230.

[8] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006, pp. 147–152.

[9] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "Improving pca-based anomaly detection by using multiple time scale analysis and kullbackleibler divergence," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1731–1751, 2014.

[10] D. F. S.D. Donald, R.V. McMillen and J. McEachen, "Modeling network conversation flux for patternless intrusion detection," in *Proc. 6th WSEAS International Multiconference onCircuits, Systems, Communications and Computers (CSCC 2002)*, 2002, pp. 441–446.

[11] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, ser. SP '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 130–. [Online]. Available: http://dl.acm.org/citation.cfm?id=882495.884435

[12] A. Ziviani, A. T. A. Gomes, M. L. Monsores, and P. S. S. Rodrigues, "Network anomaly detection using nonextensive entropy," *IEEE Communications Letters*, vol. 11, no. 12, pp. 1034–1036, 2007. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2007.070761

[13] B. Tellenbach, M. Burkhart, D. Sornette, and T. Maillart, "Beyond shannon: Characterizing internet traffic with generalized entropy metrics," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, ser. PAM '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 239–248. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00975-4_24

[14] C. F. L. Lima, F. M. Assis, and C. P. de Souza, "A comparative study of use of shannon, rényi and tsallis entropy for attribute selecting in network intrusion detection," in *Measurements and Networking Proceedings (M N), 2011 IEEE International Workshop on*, Oct 2011, pp. 77–82.

[15] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, p. 2367, 2015. [Online]. Available: http://www.mdpi.com/1099-4300/17/4/2367

[16] L. Zhang and D. Veitch, "Learning entropy," in *NETWORKING 2011 - 10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 9-13, 2011, Proceedings, Part I*, 2011, pp. 15–27. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20757-0_2

[17] lker Ozcelik and R. R. Brooks, "Deceiving entropy based dos detection," *Computers & Security*, vol. 48, pp. 234 – 245, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016740481400159X

[18] B. K. Subhabrata, E. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: Methods, evaluation, and applications," in *In Internet Measurement Conference*, 2003, pp. 234–247.

[19] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58 – 75, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/\\B6WH3-4BM8Y1G-1/2/71b7980bb85b570bc57ee73f8afcd62f

[20] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature," in *ACM SIGCOMM*, 2005.

[21] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Detecting anomalies in backbone network traffic: A performance comparison among several change detection methods," *International Journal of Sensor Networks*, vol. 11, no. 4, pp. 205–214, 2012.

[22] O. Salem, S. Vaton, and A. Gravey, "A scalable, efficient and informative approach for anomaly-based Intrusion Detection Systems: theory and practice," *International Journal of Network Management*, 2010.

[23] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "When randomness improves the anomaly detection performance," in *Proceedings of 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, 2010.

[24] R. Schweller, A. Gupta, E. Parsons, and Y. Chen, "Reversible sketches for efficient and accurate change detection over network data streams," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 207–212. [Online]. Available: http://doi.acm.org/10.1145/1028788.1028814

[25] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[26] J. Havrda and F. Charvát, "Quantification method of classification processes. Concept of structural a-entropy," *Kybernetika (Prague)*, vol. 3, pp. 30–35, 1967.

[27] C. Tsallis, "Possible generalization of boltzmann-gibbs statistics," *Journal of Statistical Physics*, vol. 52, no. 1, pp. 479–487, 1988. [Online]. Available: http://dx.doi.org/10.1007/BF01016429

[28] ——, "Entropic nonextensivity: a possible measure of complexity," *Chaos, Solitons and Fractals*, vol. 13, pp. 371–91, 2002.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177729694

[30] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, Jan 1991.

[31] S. Muthukrishnan, "Data streams: algorithms and applications," in *Proceedings of the annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003, pp. 413–413.

[32] R. Schweller, A. Gupta, E. Parsons, and Y. Chen, "Reversible sketches for efficient and accurate change detection over network data streams," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 207–212. [Online]. Available: http://doi.acm.org/10.1145/1028788.1028814

[33] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3954.txt

[34] "TCCPDUMP and LIBPCAP Home Page," http://www.tcpdump.org (accessed on November 2016).

[35] "Flow-Tools Home Page," http://www.ietf.org/rfc/rfc3954.txt (accessed on November 2016).

[36] M. Thorup and Y. Zhang, "Tabulation based 4-universal hashing with applications to second moment estimation," in *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004, pp. 615–624.

[37] "MAWILab," http://www.fukuda-lab.org/mawilab/ (accessed on November 2016).

[38] "MAWI Working Group Traffic Archive," http://mawi.wide.ad.jp/mawi/ (accessed on November 2016).

[39] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," *ACM CoNEXT*, 2010.

[40] C. Callegari, A. Casella, S. Giordano, M. Pagano, and T. Pepe, "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, October 14-16, 2013*, 2013, pp. 350–358.