Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

# WAVE-CUSUM: Improving CUSUM performance in network anomaly detection by means of wavelet analysis

C. Callegari*, S. Giordano, M. Pagano, T. Pepe

Dept. of Information Engineering, University of Pisa, Via Caruso 16, 56122 Pisa, Italy

ARTICLE INFO

*Article history:*
Received 23 December 2011
Received in revised form
2 April 2012
Accepted 6 May 2012

*Keywords:*
Network security
Intrusion detection system
Network anomaly detection
CUSUM
Wavelet analysis

ABSTRACT

The increasing number of network attacks causes growing problems for network operators and users. Thus, detecting anomalous traffic is of primary interest in IP networks management and many detection techniques, able to promptly reveal and identify network attacks, mainly detecting Heavy Changes in the network traffic, have been proposed. Among these, one of the most promising approach is based on the use of the CUSUM (CUmulative SUM). Nonetheless, CUSUM performance is strongly affected by its sensitivity to the presence of seasonal trends in the considered data.

For this reason, in this paper we propose a novel detection method based on the idea of performing a pre-processing stage of the data by means of wavelets, aimed at filtering out such trends, before applying the CUSUM algorithm.

The performance analysis, presented in the paper, demonstrates the efficiency of the proposed method, focusing on the performance improvements due to the pre-processing stage.

## 1. Introduction

In the last few years the Internet has experienced an explosive growth. Along with the wide proliferation of new services, the quantity and impact of attacks have been continuously increasing. The number of computer systems and their vulnerabilities have been rising, while the level of sophistication and knowledge required to carry out an attack have been decreasing, as much technical attack know-how is readily available on Web sites all over the world.

As a consequence, many research groups have focused their attention on developing novel detection techniques, able to promptly reveal and identify network attacks, mainly detecting Heavy Changes (HCs) in the traffic volume (Brutlag, 2000; Lakhina et al., 2004; Zhang et al., 2005; Thottan and Ji, 2003).

Nevertheless the seasonal nature of the Internet traffic, characterized by cyclic variation (e.g., daily and weekly trends), makes somehow difficult to distinguish a network anomaly from a "normal" variation of the distribution of the traffic, taking to systems that are strongly affected by high percentages of false positives. For such a reason particular attention has to be devoted to the development of methods able to correctly filter out the seasonality of the data so as to reduce the number of false alarms.

To this aim, in this work, we propose to use one of the most promising techniques for detecting changes in the traffic volume, namely the CUSUM (CUmulative SUM) algorithm (Salem et al., 2010), combined with a pre-filtering stage, realized by means of the wavelet transform.

The main idea of the CUSUM (Basseville and Nikiforov, 1993) is to detect changes in the distribution of a given time

* *Corresponding author.*
E-mail addresses: christian.callegari@iet.unipi.it (C. Callegari), stefano.giordano@iet.unipi.it (S. Giordano), michele.pagano@iet.unipi.it (M. Pagano), teresa.pepe@iet.unipi.it (T. Pepe).

0167-4048/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2012.05.001

series, and is applied in the anomaly detection field, considering that the distribution of the traffic descriptors should change between before and during the occurrence of a network anomaly. Hence, it is clear that the method performance is strongly affected by the cyclic variation of the Internet traffic (as an example, let us consider that the traffic distribution change between day and night could deceive the method).

To solve such an issue, in this paper, we propose to combine a "classical" CUSUM based approach together with wavelet analysis. In more detail the latter is used to filter out the seasonal trends in the network traffic before applying the "real" anomaly detection algorithm, based on the CUSUM method.

It is worth noticing that the aim of our work is to demonstrate how the proposed pre-filtering stage can lead to some improvements in the anomaly detection performance. In this paper we have chosen the CUSUM method because it has been shown in the literature to be promising, but in general we can think of also applying wavelet pre-filtering to other detection method.

In the paper we also face the problem of scalability. Indeed "classical" approaches based on the analysis of the single traffic flows result to be definitely infeasible in real backbone networks, even using dedicated hardware. Moreover the use of standard aggregation techniques (e.g., ingress link aggregation) has been demonstrated to be ineffective in such a context. For such reason in our method we make use of the sketches, probabilistic data structures that allow us to randomly aggregate the traffic flows. In more detail, in this paper we have chosen to use the reversible sketches that are able to also identify the traffic flows (inside an aggregate) responsible for the detected anomalies, overcoming the limitations of "classical" sketch.

The experimental results, obtained by testing our system over real traffic traces, collected in the Internet2/Abilene backbone network, demonstrate the efficiency of our method, which outperforms the "classical" method in terms of both correct detection rate and false positive rate.

The remainder of this paper is organized as follows: next section discusses some related works, while Section 3 gives a detailed description of the proposed architecture. The in Section 4 we discuss the experimental results and, finally, Section 5 concludes the paper with some final remarks.

## 2. Related work

Due to their properties, both Wavelet transform and CUSUM algorithm are quite "classical" approaches to detect irregular patterns in traffic traces.

Given the vast amount of literature on anomaly detection, in the following we only review some of the most relevant works, referring to (Thottan et al., 2010) for a more complete overview on the topic.

The first detailed work on the study of network anomalies by means of wavelet analysis is represented by Barford et al. (2002), where the authors apply general wavelet filters to the data strings of Internet traffic measurements, seen as a generic signal. The authors propose a platform for network measurement, called IMAPIT (Integrated Measurement Analysis Platform for Internet Traffic). Even if this work has inspired us the idea of performing a pre-filtering stage, before applying the CUSUM algorithm, the detection stage is based on a completely different approach.

An important work (Donoho et al., 2002) discusses the application of the wavelet transform to detect anomalies, by monitoring the access link connecting a given site to the Internet. The idea is to characterize the packets generated by the keystrokes in a SSH connection. The authors also discuss some evasion techniques. Even if quite interesting the work just focuses on a very particular kind of Internet traffic, while our work is much more "general".

A framework for real time wavelet-based analysis of network traffic anomalies, called Waveman, is then proposed in Huang et al. (2008). The system is based on the application of signal processing techniques in IDSs (Intrusion Detection Systems) and works on two distinct metrics, namely percentage deviation and entropy. The aim of the work is to evaluate the performance of various wavelet functions in detecting different types of anomalies like Denial of Service (DoS) attacks and port-scans. The tests are only performed on a few minutes of traffic leading up to the attack, the attack itself, and a few minutes of normal traffic after the attack. This experimental methodology strongly affects the general applicability of the proposed work.

Recent work (Lu and Ghorbani, 2009) proposes a new network signal modeling technique for detecting network anomalies, combining wavelet approximation and system identification theory. The system is based on the selection of 15 network flow-based features modeling the normal daily network traffic; wavelet approximation and the ARX (Auto Regressive eXogenous) system prediction technique are used together to detect network attacks. The authors also perform the comparison of four different wavelet basis functions and a completed analysis for the full 1999 DARPA network traffic dataset is carried out. The results of this work have been used in our method for selecting the best mother wavelet function to be used.

Finally it is worth noticing that the combined use of wavelet transform and the sketches to detect network anomalies has only been applied in a couple of works. In more detail, in (Pukkawanna and Fukuda, 2010) the authors apply the method already described in (Barford et al., 2002) to the time series given by the temporal evolution of the single buckets of a sketch.

Regarding the CUSUM algorithm, it has been widely used in the field of network anomaly detection. This is strongly justified by the fact that being a sequential analysis technique, typically used for monitoring change detection, the application to network anomaly detection is quite straightforward. Nevertheless, the original version of the algorithm imposes several constraints that are usually not respected in the field of anomaly detection (see next Section for more details). Hence the non parametric version of the CUSUM is usually applied (Tartakovsky et al., 2006).

To cite some of the most interesting works, in Wang et al. (2002) the authors propose to aggregate the whole traffic in one flow, and to use a non parametric version of CUSUM for detecting TCP SYN flooding, while in Siris and Papagalou

(2004) the authors present a comparative analysis of two anomaly detection algorithms (adaptive threshold and CUSUM) for the detection of TCP SYN flooding. The result of the comparison shows that CUSUM is more efficient for detecting low intensity attacks than an adaptive threshold.

One of the most recent works in the field is (Salem et al., 2010), where the authors first use sampling to reduce the amount of data and to discard unpredictable variations of legitimate traffic, then use the sequential MNP-CUSUM over sketch for anomaly detection. This work, that presents very good results, has been used in this paper as a benchmark for results comparison.

After an extensive survey, to the best of our knowledge, there is no work that presents a combined use of wavelet analysis and CUSUM algorithm. Indeed, the only work that combines wavelet and CUSUM is represented by Carl et al. (2006), where the authors apply wavelets for detecting change-points in the CUSUM statistics (and not for traffic signal decomposition). Hence the method is based on a completely opposed approach with respect to our technique.

## 3. System architecture

In this section we detail the architecture of the proposed novel Intrusion Detection System (see Fig. 2), together with the theoretical background information, necessary to understand the proposed approach. Note that we focus on the useful details only, referring the reader to the provided references for a complete description.

### 3.1. System input

First of all the input data are processed by the data formatting module. This module is responsible of reading the Netflow (Claise, 2004) traces (corresponding to the traffic gone through a given router over 5 min time bins) and of transforming them in ASCII data files, by means of the Flow-Tools (Flow-Tools Home Page).

We describe the streaming data, by using the most general model proposed in the literature: the Turnstile Model (Muthukrishnan, 2003). According to this model, the input data are viewed as a stream that arrives sequentially, item by item. Let $I = \sigma_1, \sigma_2, \ldots, \sigma_n$ be the input stream.

Each item $\sigma_t = (i_t, c_t)$ consists of a *key*, $i_t \in (1, \ldots, N)$, and a *weight*, $c_t$.

In our case, the key is represented by the destination IP address and the weight by the number of received bytes, thus the output of this first block is given by text files containing on each line an IP address (the key $i_t$) and the number of bytes received by that IP in the last time bin (the weight $c_t$). The number of distinct files is equal to the number of analyzed time bins and will be denoted as $N$ in the following.

Note that the modularity of the system allows great flexibility. Indeed, instead of considering the number of bytes received by a given IP, the system administrator can easily choose of using another traffic descriptor that better allows her to detect the different attacks.

### 3.2. Sketch module

After the data have been correctly formatted, they are passed as inputs to the hash functions responsible for the construction of the 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.

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, Schweller et al. (2004) proposes a novel algorithm (namely the reversible sketch) 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 amount of memory occupied by the corresponding data structures.

In our system, each file, corresponding to a time bin, is used to build a distinct reversible sketch table. In more detail, we have used $D = 16$ distinct hash functions, which give output in the interval $(0, 1, \ldots, w - 1)$, that means that the resulting sketch tables will be $\in \mathbb{N}_{D \times w}$, where $w = 512$ in our case. As far as the hash functions are concerned, we have

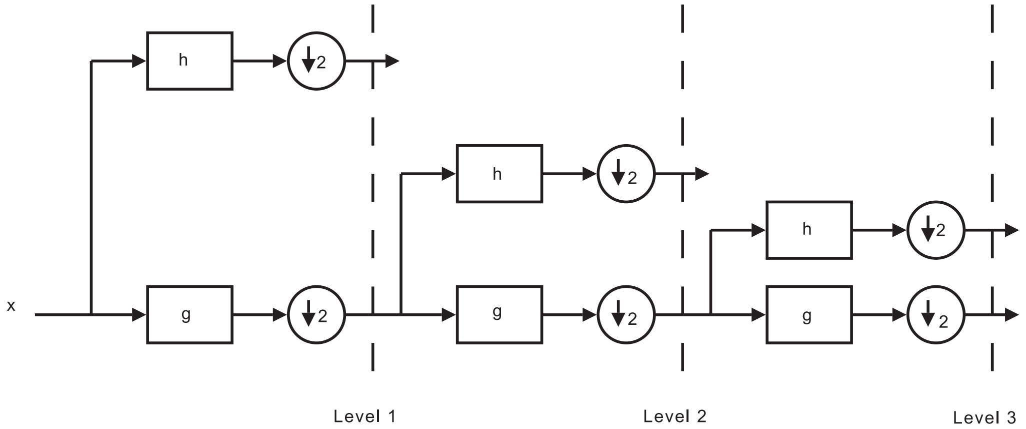

**Fig. 1 – Wavelet decomposition.**

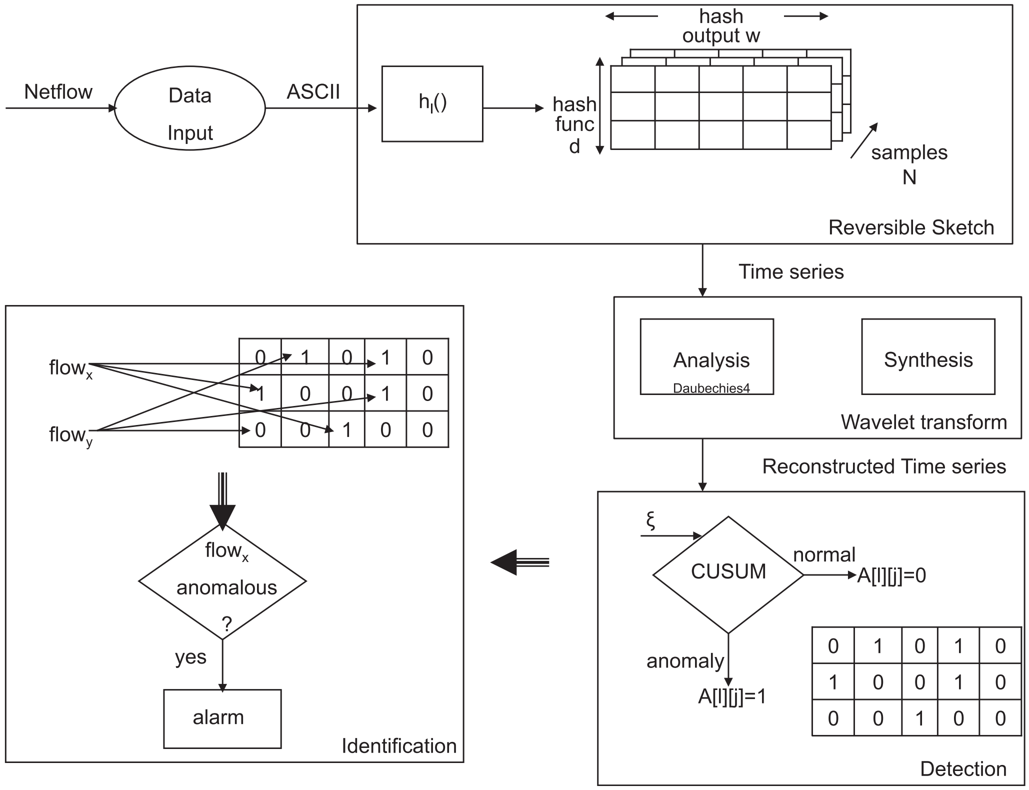

**Fig. 2 – System architecture.**

chosen to use functions belonging to the 4-universal hash family[1] (Thorup and Zhang, 2004), obtained as:

$$h(x) = \sum_{i=0}^{3} a_i \cdot x^i mod\, p\, mod\, w \tag{1}$$

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

In this way, given that we had $N$ distinct time bins, we obtain $N$ distinct sketch tables $T^n_{D \times w}$ (where $n \in$ (1, 2, …, $N$) is the time bin) that contain in each bucket the quantity of traffic (bytes) received by the related aggregate (given by the IP addresses whose hash function collides in that bucket). Hence, considering the temporal evolution of each bucket $T_{lj}$ of the sketch table, we obtain $d \cdot w$ time series of $N$ samples $T_{lj}[n]$.

### 3.3. *Wavelet module*

The time series generated by the sketch module are given as input to the wavelet module, aimed at filtering out the daily and weekly trends, which lead to very poor performance of the CUSUM algorithm (see the comparison results).

The Wavelet decomposition (Daubechies, 1992) is based on the representation of any finite-energy signal $x(t) \in L^2(\mathbb{R})$ by means of its inner products $\{x_{m,n}\}_{m,n \in \mathbb{Z}}$ with a set of functions, $\{\psi_{m,n}(t)\}_{m,n \in \mathbb{Z}}$, which are scaled and translated versions of an adequately chosen *mother wavelet* $\psi(t)$:

$$\psi_{m,n}(t) = a_0^{-m/2} \psi\left(a_0^{-m} t - n b_0\right)$$

In this work, we will consider the well-known Daubechies bases family of compactly-supported mother wavelets, introduced by the Belgian mathematician Ingrid Daubechies in 1988.

The multiresolution analysis represents the theoretical framework for the efficient calculation of the wavelet decomposition (Mallat, 1989). Let $x = (x_1, x_2, \ldots)$ denote the approximation of a finite-energy signal $x(t)$ at a given resolution; the wavelet coefficients $\{x_{m,\,n}\}$ at lower resolutions can be obtained considering the filter bank shown in Fig. 1, where the coefficients $h$ and $g$ depend on the chosen mother wavelet. In particular, the outputs of the high-pass filter $h$ give the detail coefficients (at the given resolution), while the outputs of the low-pass filter give an approximation at a lower resolution, which is further decomposed in a similar way.

In our case, the wavelet decomposition is performed on six steps (namely we obtain six levels of transformed coefficients) by using Daubechies-4 as mother wavelet. Note that this mother wavelet has been chosen because it is the most widely used in the literature and in Lu and Ghorbani (2009) it is demonstrated to be the one that offers the best performance. The signal is then reconstructed by using the coefficients from

[1] A class of hash functions $H$: (1, …, $N$) → (1, …, $w$) is a $k$-*universal hash* if for any distinct $x_0, \ldots x_{k-1} \in$ (1, …, $N$) and any possible $v_0, \ldots v_{k-1} \in$ (1, …, $w$):

$$\Pr_{h \in H} = \Pr\{h(x_i) = v_i;\ \forall i \in (1, \ldots, k)\} = \frac{1}{w^k}.$$

levels one to five, and inserting null coefficients in the sixth level, which corresponds to the low-pass component of the signal according to the notation used in Fig. 1. The result of this synthesis operation (as shown in Section 4) is the original signal without the seasonal behavior.

### 3.4. Detection module

The reconstructed time series obtained after the wavelet analysis and synthesis, are given in input to the detection module, where a non parametric version of the CUSUM (namely, the MNP-CUSUM) algorithm is performed.

Let us suppose to have a time series, given by the samples $x_n$ from a process, then the goal of the algorithm is to detect with the smallest possible delay a change in the distribution of the data.

Note that the assumption of the method, that is the knowledge of the two distributions before and after the change, implies that CUSUM is only able to decide between two simple hypotheses. But, in case of network anomalies we cannot suppose that the distribution after the change is known (usually neither the distribution before the change is known). This implies the need of using the non parametric version of the algorithm (Tartakovsky et al., 2006), which is based on a different definition of the test statistics $S_n$. In more detail in this paper we have used the multichart non parametric CUSUM (MNP-CUSUM), in which $S_n$ is defined as:

$$\begin{aligned} S_0 &= x_0 \\ S_{n+1} &= S_n + x_n - (\mu_n + c \cdot \sigma_n) \end{aligned} \tag{2}$$

where $\mu_n$ and $\sigma_n$ are the mean value and the standard deviation until step $n$, while $c$ is a tunable parameter of the algorithm.

The rational behind the CUSUM algorithm is that $S_n$ decreases before the change, and increases linearly with a positive slope after the change, until the increment reaches the threshold $\xi$ when the alarm is raised.

In our system the quantity $\mu_n$ and $\sigma_n$ have been estimated by using the EWMA (Exponentially Weighted Moving Average) algorithm, while the value of the parameter $c$ has been set equal to 0.5 (also note that the algorithm has experimentally shown to be robust to the choice of this parameter).

An anomaly is thus detected at a given time bin, if the test statistics starts increasing with a positive slope and the increment exceeds the threshold $\xi$. Note that, in the experimental tests the threshold $\xi$ has been set by means of Monte Carlo Simulation.

The output of this phase is a binary matrix ($A[d][w]$), for each time bin, that contains a "1" if the time series corresponding to a given 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, in our tests $H = (D/2) + 1$). If so, the mediator reveals an anomaly, otherwise the matrix $A$ is discarded.

### 3.5. Identification phase

In case the voting system outputs the presence of an anomaly in a given time bin, the system applies the reversible sketch algorithm to the sketch table given by the value of all the time series in that time bin for identifying IP addresses responsible for the anomalies (see Schweller et al. (2004) for the algorithm details).

## 4. Experimental results

A well-known serious issue in testing IDSs is represented by the lack of complete datasets provided with a ground truth. Indeed, the only one (DARPA IDEVAL) dates back to 1999 (Lippmann et al., 2000) and it is not representative of real traffic (Mahoney and Chan, 2003). Thus, it is a common choice to use a real dataset and to synthetically add some anomalies (Lakhina et al., 2005).

The proposed system has been tested using a publicly available dataset, composed of traffic traces collected in the Abilene/Internet2 Network (The Internet2 Network), a hybrid optical and packet network used by the U.S. research and education community.

The used traces consist of the traffic related to nine distinct routers, collected in one week, and are organized into 2016 files, each one containing data about 5 min of traffic. To be noted that the last 11 bits of the IP addresses are anonymized for privacy reasons; nevertheless we have more than 220,000 distinct IP addresses.

Since the data provided by the Internet2 project do not have a ground truth file, we are not capable of saying *a priori* if any anomaly is present. Because of this reason we have performed a manual verification of the data (according to the method presented in Lakhina et al. (2005)), analyzing the traces for which our system reveals the biggest anomalies. Moreover we have synthetically added some anomalies in the data, so as to be able to correctly interpret the offered results (more details in Appendix A).

As already stated in the previous section, we have considered as input to the system the number of bytes received by a given IP address. This choice is supported by the obtained experimental results. Nevertheless, it is possible to feed the system with another metric, just simply modifying the first block, if the "new" metric can result more suitable for detecting some attacks.

Before analyzing the performance of the system in terms of detected anomalies, in the first three figures we empirically show the reason for which the proposed method is able to outperform the "classical" CUSUM methods. Regarding the "classical" method, used as a benchmark for performance comparison, we have used the method described in Salem et al. (2010) that represents, to the best of our knowledge, the most recent version of a CUSUM based anomaly detection system. As far as the system parameters are concerned, to

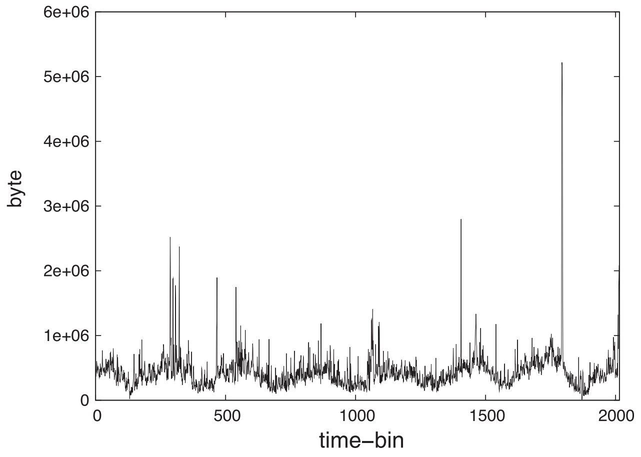

**Fig. 3 – Original signal (traffic aggregate over one week).**

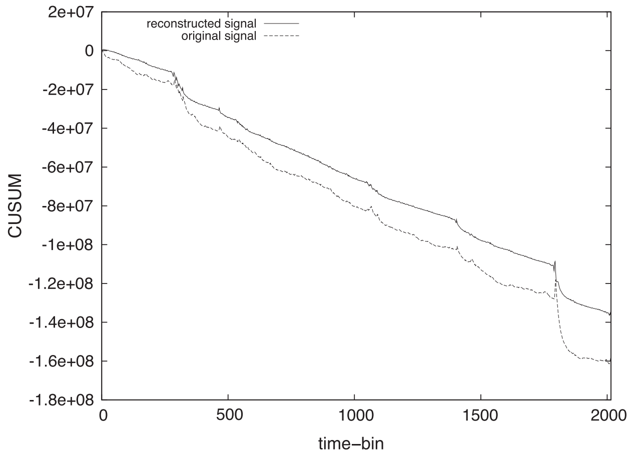

**Fig. 5 – CUSUM.**

allow a fair comparison, we have used the settings suggested in Salem et al. (2010).

Fig. 3 shows the temporal evolution of one of the buckets of the sketch over one week, before performing wavelet analysis. In more detail, it represents the number of bytes received by an IP aggregate and, as can be clearly observed, presents the "typical" seasonality of the network traffic, with a daily and a weekly trend. It is easy to understand that the seasonality in the data can make difficult the detection of some anomalies, which can be masked by such trend.

Fig. 4 shows the same traffic aggregate, after wavelet analysis. As described in the previous section the signal has been reconstructed after that the wavelet coefficients corresponding to the low frequencies (level six) have been put to zero. The result is that the daily as well as the weekly trends of the signal have "disappeared", making intuitively easier the detection of the anomalies.

This intuition is confirmed by the plot of the CUSUM statistics in Fig. 5, where we can easily see that the statistics related to the original signal (dotted line) is much more "rugged" than the one related to the reconstructed signal (solid line). Hence, considering that (as stated in the theoretical section on CUSUM) an anomaly is revealed if the CUSUM statistics starts increasing and exceeds a given threshold, our method seems to be more robust to signal noise than the "classical" method; the following tables show that this intuition is confirmed by the experimental results.

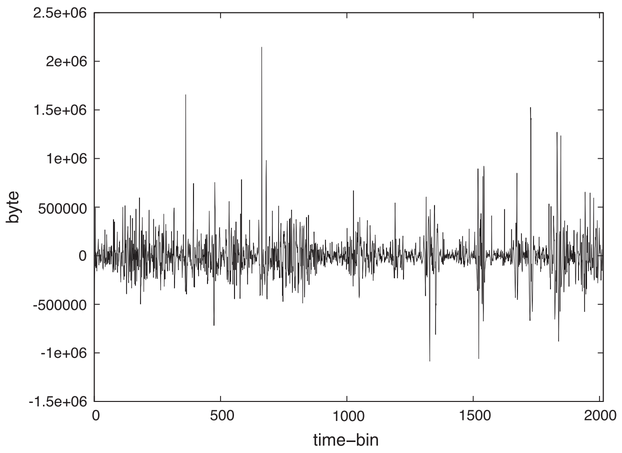

**Fig. 4 – Reconstructed signal.**

Since, given the nature of the dataset, we cannot plot a ROC (Receiver Operating Characteristic) curve, in the following tables we report the total number of detected anomalies (i.e., the number of detections, correct detections – both of synthetic and already present anomalies – plus the false positives) and the number of synthetic anomalies detected by the system. Note that the tables have been obtained varying the values of the threshold $\xi$. The real values of such thresholds (set by Monte Carlo simulation) are not reported since are not significant in themselves, just consider that the first values (namely $\xi_1$) always corresponds to the highest threshold value for which the system is able to detect all the 154 synthetic anomalies.

Hence, the system is always able to obtain a 100% detection rate (revealing all the 154 synthetic anomalies), but the performance can be strongly different depending on the total number of detected anomalies that has a direct impact on the number of false alarms.

To assess the validity of the proposed system we have carried out three distinct sets of simulations, in which we have injected, in the original traces, anomalies of low, medium and high intensity/volume. Note that in all the three cases, the added anomalies reflect the nature of real anomalies, already present in the original data.

Tables 1 and 2 respectively present the performance of our system and of the classical CUSUM system, when facing the

**Table 1 – Experimental results: our method – low volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_1$ | 703 | 154 |
| $\xi_2$ | 412 | 137 |
| $\xi_3$ | 348 | 94 |
| $\xi_4$ | 297 | 67 |
| $\xi_5$ | 226 | 9 |
| $\xi_6$ | 189 | 7 |
| $\xi_7$ | 102 | 6 |

**Table 2 – Experimental results: "classical" method – low volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_1$ | 963 | 154 |
| $\xi_2$ | 806 | 19 |
| $\xi_3$ | 778 | 17 |
| $\xi_4$ | 740 | 16 |
| $\xi_5$ | 710 | 14 |
| $\xi_6$ | 623 | 13 |
| $\xi_7$ | 595 | 12 |

**Table 4 – Experimental results: "classical" method – medium volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_1$ | 751 | 154 |
| $\xi_2$ | 612 | 23 |
| $\xi_3$ | 598 | 17 |
| $\xi_4$ | 571 | 16 |
| $\xi_5$ | 537 | 15 |
| $\xi_6$ | 514 | 13 |
| $\xi_7$ | 491 | 11 |

detection of low volume anomalies. We can easily notice that the performance is strongly different; indeed our system, when detecting all the 154 synthetic, also detects 549 additional anomalies (703 total anomalies minus the synthetic ones), while the "classical" system detects 809 additional anomalies. In this case, to really evaluate the performance, we have performed a manual verification of the dataset, checking the additional detections of the system. From that, we can conclude that, about 150 anomalies of the 549 detected by our system are real anomalies and 150 are suspicious activities. That means that our system has a false positive rate between the 12% and 20%, when the detection rate (over the synthetic anomalies) is 100%, while for the classical system the false positive rate raises to about 25%–33%. Note that the "classical" system does not detect all the anomalies already present in the data, while detecting some more "false" anomalies. Moreover we can notice from Table 1 that we can obtain a negligible false alarm rate, when the detection rate is about 89%, while it is not possible to lower the false alarm rate without significantly worsen the detection rate for the classical system.

Tables 3 and 4 represent the same comparison for medium volume anomalies. Also in this case, we can easily see that the two systems present strongly different performance. Indeed, in this case our system is able to achieve an almost ideal behavior obtaining the 100% of correct detection in correspondence of a negligible false alarm rate, while the "classical" system, when achieving the 100% of detection rate also has a false alarm rate between 22% and 30%. Also, we can notice that our system results to be robust to small variations in the values of the threshold, while the "classical" method performance abruptly change when varying the threshold.

Finally, Tables 5 and 6 show the performance of the two systems in the detection of high volume anomalies. Note that in this case, in our system, $\xi_0$1 is the highest value for which the system detects all the synthetic anomalies plus the about 150 anomalies already present in the original data. For these results, the considerations done in the previous two cases are still valid. Indeed our system is able to detect all the anomalies with a negligible false alarm rate, while with the "classical" system we must accept a high false alarm rate, also when not detecting all the anomalies.

**Table 3 – Experimental results: our method – medium volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_1$ | 308 | 154 |
| $\xi_2$ | 256 | 117 |
| $\xi_3$ | 205 | 80 |
| $\xi_4$ | 182 | 75 |
| $\xi_5$ | 121 | 16 |
| $\xi_6$ | 80 | 7 |
| $\xi_7$ | 60 | 6 |

**Table 5 – Experimental results: our method – high volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_0$ | 303 | 154 |
| $\xi_1$ | 262 | 154 |
| $\xi_2$ | 127 | 19 |
| $\xi_3$ | 124 | 17 |
| $\xi_4$ | 85 | 16 |
| $\xi_5$ | 39 | 14 |
| $\xi_6$ | 33 | 13 |

**Table 6 – Experimental results: "classical" method – high volume anomalies.**

| Threshold | Total anomalies | Synthetic anomalies |
|---|---|---|
| $\xi_1$ | 554 | 154 |
| $\xi_2$ | 424 | 27 |
| $\xi_3$ | 418 | 14 |
| $\xi_4$ | 382 | 12 |
| $\xi_5$ | 365 | 10 |
| $\xi_6$ | 340 | 9 |
| $\xi_7$ | 317 | 8 |

For sake of brevity we do not show here the results of the identification module, given that they do not depend on the use of wavelet analysis and the performance of the reversible sketch are well-known in the literature (Schweller et al., 2004).

## 5. Conclusions

In this paper, we have proposed a novel anomaly detection method, based on a combined use of wavelet analysis and the CUSUM algorithm. In more detail wavelet analysis is used to filter the seasonality from the traffic aggregates so as to improve the performance of the CUSUM based anomaly detection techniques.

The experimental tests have demonstrated the efficiency of the proposed solution for different intensity of the anomalies. The advantages are both in terms of false alarm rate as well as of robustness to the changes of the threshold value.

## Acknowledgments

This work was partially supported by DEMONS, a research project supported by the European Commission under its 7th Framework Program (contract-no. 257315). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DEMONS project or the European Commission.

## Appendix A

The dataset has been realized by adding some synthetic anomalies to the Abilene/Internet2 Network traffic traces. In more detail we have used the traffic traces collected in one week and associated to the nine routers of the backbone networks.

Given such traces we have added anomalies that can be associated to DoS and DDoS attacks, either spanning a single or multiple time bins. To introduce anomalies that are plausible in the dataset, we have at first performed a manual verification of the data (according to the method presented in Lakhina et al. (2005)), analyzing the traces for which our system reveals the anomalies.

Then we have computed the average volume $V$ of the detected anomalies (in terms of associated traffic) and we have synthetically produced 154 distinct anomalies, spanning either a single or multiple (up to three) time bins. These anomalies are represented by $n$ traffic flows ($n$ is randomly chosen between 1 and 5), characterized by a volume of traffic given by $V$ multiplied by a scaling factor $k$ that randomly assumes a value in the range [0.5, 0.9] for the low volume anomaly case, [0.9, 1.1] for the medium volume anomaly case, and [1.1, 2.5] for the high volume anomaly case.

Eventually, the obtained anomalies have been randomly inserted in the traffic traces (according to a uniform distribution).

The presented results represent the average of ten independent runs.

## REFERENCES

Barford P, Kline J, Plonka D, Ron A. A signal analysis of network traffic anomalies. In: Internet measurement workshop; 2002. p. 71–82.

Basseville M, Nikiforov IV. Detection of abrupt changes: theory and application. NJ, USA: Prentice-Hall, Inc.; 1993.

Brutlag JD. Aberrant behavior detection in time series for network monitoring. In: Proceedings of the 14th USENIX conference on system administration. Berkeley, CA, USA: USENIX Association. p. 139–46. URL, http://portal.acm.org/citation.cfm?id=1045502.1045530; 2000.

Carl G, Brooks RR, Rai S. Wavelet based denial-of-service detection. Computers and Security 2006;25(8):600–15.

Claise B. Cisco systems NetFlow services export version 9, RFC 3954 (informational). URL, http://www.ietf.org/rfc/rfc3954.txt; Oct. 2004.

Daubechies I. Ten lectures on wavelets, no. 61 in CBMS-NSF series in applied mathematics. Philadelphia: SIAM; 1992.

Donoho DL, Flesia AG, Shankar U, Paxson V, Coit J, Staniford S. Multiscale stepping-stone detection: detecting pairs of jittered interactive streams by exploiting maximum tolerable delay. In: Proc. of the 5th international symposium on recent advances in intrusion detection (RAID). Springer; 2002. p. 17–35.

Flow-Tools Home Page, http://code.google.com/p/flow-tools/.

Huang C-T, Thareja S, Shin Y-J. Wavelet-based real time detection of network traffic anomalies. International Journal of Network Security 2008;6:309–20.

Lakhina A, Crovella M, Diot C. Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM; 2004. p. 219–30.

Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. SIGCOMM Computer Communication Review 2005;35(4):217–28.

Lippmann R, Haines J, Fried D, Korba J, Das K. The 1999 DARPA off-line intrusion detection evaluation. Computer Networks 2000;34(4):579–95.

Lu W, Ghorbani AA. Network anomaly detection based on wavelet analysis. EURASIP Journal on Advances in Signal Processing 2009;2009:4:1–4:16.

Mahoney MV, Chan PK. An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. In: Proceedings of the international symposium on recent advances in intrusion detection (RAID). Springer-Verlag; 2003. p. 220–37.

Mallat S. Multifrequency channel decompositions of images and wavelet models. IEEE Transactions on Acoustics, Speech and Signal Processing 1989;37(12):2091–110.

Muthukrishnan S. 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. p. 413.

Pukkawanna S, Fukuda K. Combining sketch and wavelet models for anomaly detection. In: Intelligent computer communication and processing (ICCP), 2010 IEEE international conference on; 2010. p. 313–9.

Salem O, Vaton S, Gravey A. A scalable, efficient and informative approach for anomaly-based intrusion detection systems: theory and practice. International Journal of Network Management 2010;20:271–93, http://dx.doi.org/10.1002/nem.748.

Schweller R, Gupta A, Parsons E, Chen Y. Reversible sketches for efficient and accurate change detection over network data streams. In: Proceedings of the ACM SIGCOMM conference on Internet measurement, IMC '04. New York, NY, USA: ACM. p. 207–12, http://doi.acm.org/10.1145/1028788.1028814; 2004.

Siris VA, Papagalou F. Application of anomaly detection algorithms for detecting SYN flooding attacks. In: Global telecommunications conference; 2004.

Tartakovsky AG, Rozovskii BOL, Zek RUBB, Kim H. Detection of intrusions in information systems by sequential change-point methods. Statistical Methodology 2006;3:252–93.

The Internet2 Network, http://www.internet2.edu/network/.

Thorup M, Zhang Y. 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. p. 615–24.

Thottan M, Ji C. Anomaly detection in IP network. In: IEEE trans. signal processing, vol. 51; 2003. p. 2191–204.

Thottan M, Liu G, Ji C. Anomaly detection approaches for communication networks. London: Springer; 2010.

Wang H, Zhang D, Shin KG. Syn-dog: sniffing syn flooding sources. In: IEEE ICDCS; 2002. p. 421–8.

Zhang Y, Ge Z, Greenberg A, Roughan M. Network anomography. In: IMC; 2005.

**Christian Callegari** received his Laurea degree in Telecommunication Engineering “cum laude” on October 2004 from the University of Pisa. In 2005 he obtained the qualification to practice the profession of Engineer and he is a member of the IEEE Communication Society. In 2008 he obtained a Ph.D. in Information Engineering from the University of Pisa. He is currently a Postdoctoral fellow in the Telecommunication Network research group at the department of Information Engineering of the University of Pisa. His research and professional areas of interest are Network Security, Traffic Classification, Traffic Engineering, MPLS architecture and Network Simulation.

**Stefano Giordano** received the Laurea degree “cum laude” in Electronics Engineering and the Ph.D. degree in Information Engineering from the University of Pisa in 1990 and 1994 respectively. He worked with Consorzio Pisa Ricerche since 1990 in the field of Telecommunication Networks participating and coordinating several research activities. Since the end of 2001 he is associate professor at the Department of Information Engineering of the University of Pisa. His research and professional areas of interest are Broadband Communications, Telecommunication Networks Analysis and Design, Simulation of Communication Networks and Systems, Multimedia communications.

**Michele Pagano** received Laurea (cum laude) in Electronics Engineering in 1994 and a Ph.D. in Electronics Engineering in 1998, both from the University of Pisa. Since 2007, he is an associate professor at the University of Pisa. His research interests are related to statistical characterization of traffic flows and to network performance analysis, mainly in the framework of architectures able to support Quality of Service. A new research field is represented by network security issues, mainly in the framework of Intrusion Detection. He has co-authored around 100 papers published in international journals and presented in leading international conferences.

**Teresa Pepe** received her Laurea degree in Telecommunication Engineering “cum laude” on September 2008 from the University of Pisa. In 2009 she obtained the qualification to practice the profession of Engineer. On November 2008 she won a grant funded by the MIUR for a Ph.D. position in Information engineering at the Department of Information Engineering of the University of Pisa. She is actually a Ph.D. Student in the Telecommunication Network research group of the Department of Information Engineering at the University of Pisa. Her research and professional areas of interest are Network Security and Traffic Classification.