

Wavelet methods for the detection of anomalies and their application to network traffic analysis

D.W. Kwon*, K. Ko†, M. Vannucci*‡, A.L.N. Reddy§, and S. Kim‡

March 24, 2005

Abstract

Here we develop an integrated tool for online detection of network anomalies. We consider statistical change point detection algorithms, for both local changes in the variance and for jumps detection, and propose modified versions of these algorithms based on moving window techniques. We investigate performances on simulated data and on network traffic data with several superimposed attacks. All detection methods are based on wavelet packets transformations.

Key words: Change point detection, Network traffic, Statistical hypothesis testing, Wavelet transforms.

1 Introduction

In this paper we investigate the performances of an integrated tool for the detection of network anomalies with the goal of quickly identifying malicious attacks. Detection

*Department of Statistics, Texas A&M University, College Station, TX 77843-3143

†Department of Mathematics, Boise State University, ID

‡Corresponding author, MVANNUCCI@STAT.TAMU.EDU, Ph:(979)845-0805, Fax:(979)845-3144. Supported

by NSF-CAREER award DMS-0093208 and by Task Force at TAMU

§Department of Electrical Engineering, Texas A&M University

of network anomalies is a crucial task in network traffic management. Here we look at a network anomaly as a possible attack by a malicious user. Large scale network attacks cause huge costs and a waste of network resources. Early detection allows quick actions and minimizes network damage. In statistical terms, the detection of an anomaly can be considered as a change point problem. In this paper we consider two kinds of detection methods: Those that detect changes in the local variance of the data and those that detect jumps in the observed data. All statistical methods we consider are wavelet-based. Wavelet transformations have been proven to be a valid tool for the analysis of network traffic, mainly because of their locality and decorrelation properties, see for example Riedi *et al.* [1], Gilbert *et al.* [2], Gilbert [3], Resnick *et al.* [4] and Kim *et al.* [5]. We look at the implementation of the detection methods based on wavelet packet transformations. We explore performances on simulated data. We also analyse the trace data used in Kim *et al.* [5], where the authors propose a novel definition of data correlation for the analysis of traffic packets and classify various types of network attacks as either variance changes or sharp jumps.

For detection we consider the iterated cumulative sums of squares (ICSS) algorithm and the Schwarz information criterion (SIC) algorithm, for the identification of multiple variance change points in sequence data, and the approach suggested by Wang [6] for the detection of sharp jumps and cusps in the data. We explore the implementation of these detection methods based on wavelet packets and assess performances in detecting network traffic attacks in real-time. The ICSS algorithm was originally proposed by Inclán and Tiao [7] while Chen and Gupta [8] suggested the use of the SIC algorithm for change detection. Whitcher *et al.* [9] adapted the ICSS algorithm to discrete wavelet transforms (DWT) and to maximal overlap discrete wavelet transforms (MODWT), also known as “non-decimated”, “translation invariant” or “stationary”. Their work is limited to the detection of variance change points for data that show long-range dependence (LRD). Gabbanini *et al.* [10] extended the ICSS procedure to discrete wavelet packet transforms (DWPT) and maximal overlap discrete wavelet packet transforms (MODWPT). The use of wavelet packets allowed them to analyze a broader class of data than LRD.

Here we exploit the Gabbanini *et al.* [10] method to see how effectively we can detect network traffic anomalies caused by malicious users' network attacks. While Gabbanini *et al.* [10] used only the ICSS algorithm, we implement both the SIC and the ICSS algorithms based on wavelet packets. In addition, we extend the method of Wang [6] to maximal overlap wavelet packets, i.e. MODWPT. In the sequel we will use the term “packet” with two different meanings. In network traffic terminology, data information is partitioned into small “chunks” called packets. The header of the packet contains useful information such as the address (source and destination) and the packet count. In wavelet theory terminology, the term packet indicates the particular frequency band at which the coefficients of a “packet” transform are associated. See section 2 for more details.

The paper is organized as follows. In section 2 we summarize the main concepts of the DWT, DWPT and MODWPT. In section 3 we describe the ICSS and SIC algorithms, for testing and locating multiple variance change points, and the jump detection algorithm of Wang [6]. In Section 4 we describe our implementation of the detection schemes based on wavelet packets and provide step-by-step algorithms. In section 5 we perform a simulation study to compare performances of the ICSS and SIC algorithms. Section 6 deals with the description of the network traces, protocols and processing of the data. We give final remarks in section 7.

2 Wavelet theory

We first provide a brief review of the main concepts in wavelet theory and wavelet transforms. We begin with the exposition of the continuous wavelet transform (CWT) and then describe the standard discrete wavelet transform (DWT) and the maximal overlap discrete wavelet transform (MODWT) and finally introduce wavelet packets transformations (DWPT and MODWPT).

2.1 Basic concepts in wavelets

Wavelets have been very successful as an analytical tool to represent signals, in denoising, data compression and in time-scale analysis of time series, to mention a few of their applications. Furthermore, wavelets enjoy efficient computational schemes for the calculations of the wavelet transforms. Vidakovic [11] provides good references to wavelets and in particular to wavelet methods for statistical analyses.

Using wavelets any function in $L_2(\mathbb{R})$ can be written as a linear combination of the type

$$f(t) = \sum_j \sum_k a_{j,k} \phi_{j,k}(t) = \sum_k a_{j_0,k} \phi_{j_0,k}(t) + \sum_{j \geq j_0} \sum_k b_{j,k} \psi_{j,k}(t)$$

where $a_{j,k} = \langle f(t), \phi_{j,k}(t) \rangle$, $b_{j,k} = \langle f(t), \psi_{j,k}(t) \rangle$ and where $\phi_{j,k}, \psi_{j,k}$ are the so called scaling and wavelet functions, respectively, that satisfy the following conditions:

$$\begin{aligned} \int \phi_{j,k}(t) \phi_{j,k'}(t) dt &= \delta_{k,k'} \\ \int \phi_{j,k}(t) \psi_{j',k'}(t) dt &= 0 \\ \int \psi_{j,k}(t) \psi_{j',k'}(t) dt &= \delta_{j,j'} \delta_{k,k'} \end{aligned}$$

where $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ if $i \neq j$. Several different families of wavelet functions have been defined, each characterized by different properties, such as smoothness, compact support, and so on.

Discrete wavelets are defined as

$$\begin{aligned} \phi_{j,k}(t) &= 2^{j/2} \phi(2^j t - k) \\ \psi_{j,k}(t) &= 2^{j/2} \psi(2^j t - k) \end{aligned}$$

with ϕ satisfying $\phi(t) = \sum_k h(k) \phi_{1,k}$ and

$$\psi(t) = \sqrt{2} \sum_k (-1)^k h(-k+1) \phi(2t-k) = \sqrt{2} \sum_k g(k) \phi(2t-k).$$

The sequences $\{h(k), k \in \mathbb{Z}\}$ and $\{g(k), k \in \mathbb{Z}\}$ are quadrature mirror filters with $g(k) = (-1)^k h(1-k)$.

2.2 DWT

Let $X = (x_1, \dots, x_T)$ be a time series, i.e. a vector of observations from a stochastic process. The DWT is an orthogonal transformation of the data that operates via recursive filters according to the pyramidal algorithm illustrated in Figure 1, Mallat [12]. If $T = 2^J$ the algorithm produces scaling coefficients at a coarsest level J , describing global features of the data, and wavelet coefficients at a number of finer scales $1, \dots, J$ describing local features. We denote with $H = (h_0, \dots, h_{L-1})$ and $G = (g_0, \dots, g_{L-1})$ the scaling and wavelet filters, respectively, and with L the width of the filters. At the first level, $j = 1$, wavelet coefficients $w_{1,t}$ and scaling coefficients $v_{1,t}$ are defined as

$$w_{1,t} = \sum_{l=0}^{L-1} g_l x_{2t+1-l} \bmod T, \quad v_{1,t} = \sum_{l=0}^{L-1} h_l x_{2t+1-l} \bmod T.$$

The wavelet coefficients $w_{2,t}$ and scaling coefficients $v_{2,t}$ at level 2 are computed from the scaling coefficients at level 1 as follows

$$w_{2,t} = \sum_{l=0}^{L-1} g_l v_{1,2t+1-l} \bmod T, \quad v_{2,t} = \sum_{l=0}^{L-1} h_l v_{1,2t+1-l} \bmod T.$$

Similarly, at levels $j = 3, \dots, J$ the wavelet and scaling coefficients are obtained as

$$w_{j,t} = \sum_{l=0}^{L-1} g_l v_{j-1,2t+1-l} \bmod T, \quad v_{j,t} = \sum_{l=0}^{L-1} h_l v_{j-1,2t+1-l} \bmod T.$$

Due to the decimating operator, at level j we have $\frac{T}{2^j}$ scaling and wavelet coefficients. The condition $T = 2^J$ is not strictly required if a partial DWT is performed, i.e. using levels $1, \dots, J_0 < J$. In this case we can relax the condition to $T = K2^{J_0}$, for some positive integer K .

2.3 MODWT

In contrast to the DWT, the maximal overlap wavelet transform (MODWT), Percival and Walden [13], does not decimate the coefficients and therefore the number of scaling and wavelet coefficients at every level of the transform is the same as the number of sample observations. For this reason the MODWT is also called non-decimated DWT. Although it loses orthogonality and efficiency in computation, this transform does not

have any restriction on the sample size and it is shift invariant. Wavelet coefficients, $\tilde{w}_{j,t}$ and scaling coefficients, $\tilde{v}_{j,t}$ at levels $j, j = 1, \dots, J$ are obtained as follows:

$$\begin{aligned}\tilde{w}_{1,t} &= \sum_{l=0}^{L-1} \tilde{g}_l x_{t-l} \bmod T, & \tilde{v}_{1,t} &= \sum_{l=0}^{L-1} \tilde{h}_l x_{t-l} \bmod T \\ \tilde{w}_{2,t} &= \sum_{l=0}^{L-1} \tilde{g}_l \tilde{v}_{1,t-l} \bmod T, & \tilde{v}_{2,t} &= \sum_{l=0}^{L-1} \tilde{h}_l \tilde{v}_{1,t-l} \bmod T \\ \tilde{w}_{j,t} &= \sum_{l=0}^{L-1} \tilde{g}_l \tilde{v}_{j-1,t-l} \bmod T, & \tilde{v}_{j,t} &= \sum_{l=0}^{L-1} \tilde{h}_l \tilde{v}_{j-1,t-l} \bmod T.\end{aligned}$$

The wavelet and scaling filters, \tilde{g}_j, \tilde{h}_j are rescaled as $\tilde{g}_j = g_j/2^{j/2}, \tilde{h}_j = h_j/2^{j/2}$. Non-decimated wavelet coefficients represent differences between generalized averages of the data on a scale $\tau_j = 2^{j-1}$ (or level j).

2.4 Wavelet packet transforms

Wavelet packets, Wickerhauser [14], induce a finer partition of the frequency space, see the right panel of Figure 1. In the discrete wavelet packet transform (DWPT) and the non-decimated version (MODWPT) both scaling and wavelet coefficients are subject to the high-pass and low-pass filtering when computing the next level scaling and wavelet coefficients. With the standard transforms, scaling coefficients identify the frequency band $[0, 1/2^{J+1}]$, with J the coarsest level, while wavelet coefficients at level j describe the frequency band $[1/2^{j+1}, 1/2^j]$. The discrete wavelet packet transforms, DWPT and MODWPT, on the other hand, partition the whole frequency band, $[0, 1/2]$, into equal length frequency bands. For example, at a given level j , we have 2^j frequency partitions with equal length. This finer partition induced by the DWPT implies better decorrelation properties, as exploited in Percival *et al.* [15], Whitcher [16],[17] and Gabbanini *et al.* [10].

As a filtering of the original time series the MODWPT can be written as

$$\tilde{w}_{j,n,t} = \sum_{l=0}^{L-1} \tilde{f}_{j,n,l} x_{(t-l) \bmod T}, \quad (1)$$

for $n = 1, \dots, T$, where

$$\tilde{f}_{j,n,l} = \sum_{k=0}^{L-1} \tilde{f}_{n,k} \tilde{f}_{j-1, \lfloor n/2 \rfloor, l-2^{j-1}k}, 0 \leq l \leq L-1 \quad (2)$$

with

$$\tilde{f}_{n,l} = \begin{cases} \tilde{g}_l & \text{if } n \bmod 4 = 0 \text{ or } 3 \\ \tilde{h}_l & \text{if } n \bmod 4 = 1 \text{ or } 2 \end{cases} \quad (3)$$

and $\tilde{g}_l = (-1)^{l+1} \tilde{h}_{L-l-1}$, and such that $\{\tilde{f}_{1,0,l} = \tilde{g}_l, 0 \leq l \leq L-1\}$ and $\{\tilde{f}_{1,1,l} = \tilde{h}_l, 0 \leq l \leq L-1\}$.

3 Detection methods

In this section we describe two kinds of detection methods: Those that detect changes in the local variance of the data and those that detect jumps in the observed data. In the next section we will discuss our adaption of these methods to wavelet packets and related implementation issues.

3.1 Variance change points detection algorithms

We first summarize the ICSS and SIC detection algorithms for the detection of variance change points and describe a binary segmentation procedure that allows the adaption of these methods to the detection of multiple change points.

The iterated cumulative sums of squares (ICSS) algorithm aims at testing and identifying multiple variance changes in a sequence of independent observations. Null and alternative hypotheses are specified as

$$H_0 : \sigma_1^2 = \sigma_2^2 = \dots = \sigma_T^2 \quad \text{versus} \quad H_a : \sigma_1^2 = \dots = \sigma_k^2 \neq \sigma_{k+1}^2 = \dots = \sigma_T^2.$$

We denote with $C_k = \sum_{t=1}^k x_t^2$ the cumulative sum of squares of a series of uncorrelated random variables $\{x_t\}$ with mean 0 and variances σ_t^2 , $t = 1, \dots, T$. The test statistic is $D = \max(D^+, D^-)$ where

$$D^+ = \max_{1 \leq k \leq T-1} \left(\frac{k+1}{T} - P_k \right)$$

$$\begin{aligned}
D^- &= \max_{1 \leq k \leq T-1} \left(P_k - \frac{k}{T} \right) \\
P_k &= \frac{C_k}{C_T}, \quad k = 1, \dots, T.
\end{aligned}$$

Variance change points are located by looking at $k^* = \operatorname{argmax}_k D$. When the maximum absolute value of D exceeds a certain predetermined value, then we take the point k^* as the change point estimate. Whitcher *et al.* [9] obtained predetermined values for D under the null hypothesis by using Monte Carlo simulation. Inclán and Tiao [7] showed that when the random variables $\{x_t\}$ are independent distributed the asymptotic distribution of D is that one of a Brownian bridge. Whitcher *et al.* [9] suggested to use at least $T = 128$ sample size to conform with this asymptotic approximation.

The Schwarz information criterion (SIC) was suggested by Schwarz [18] and is one of the modifications of Akaike information criterion (AIC) introduced by Akaike [19]. These criteria are useful tools for model selection. Let $\{x_t\}$ be a sequence of independent and identically distributed random variables with probability density function $f(\cdot|\theta)$, where f is a model with K parameters, that is,

$$\text{Model}(k) = \{f(\cdot|\theta) : \theta = (\theta_1, \theta_2, \dots, \theta_K), \theta \in \Theta_k\}$$

$$\text{where } \Theta_k = \{\Theta_k : \theta_{k+1} = \theta_{k+2} = \dots = \theta_K\}, \quad k = 1, \dots, K-1.$$

The *SIC* is defined as $-2 \log L(\bar{\theta}_k) + p \log T$, where $L(\bar{\theta}_k)$ is the maximum likelihood function for the model(k), p is the number of parameters in the model, and n is the total number of samples. We specify the form of $SIC(T)$ and $SIC(k)$ as follows

$$SIC(T) = T \log 2\pi + T \log \hat{\sigma}^2 + T + \log T$$

$$SIC(k) = T \log 2\pi + k \log \hat{\sigma}_1^2 + (T - k) \log \hat{\sigma}_T^2 + T + 2 \log T$$

where

$$\hat{\sigma}^2 = \frac{1}{T} \sum_{i=1}^T (x_i - \bar{x})^2, \quad \hat{\sigma}_1^2 = \frac{1}{k} \sum_{i=1}^k (x_i - \bar{x})^2, \quad \text{and} \quad \hat{\sigma}_T^2 = \frac{1}{(T-k)} \sum_{i=k+1}^T (x_i - \bar{x})^2.$$

Under the same null and alternative hypotheses described above for the case of the ICSS algorithm, the null hypothesis is now rejected based on the principle of minimum

information criterion, that is, we reject if $SIC(T) \geq \min_{2 \leq k \leq T-2} SIC(k)$ and estimate the change point as \hat{k} such that

$$SIC(\hat{k}) = \min_{2 \leq k \leq T-2} SIC(k).$$

Notice that we can only detect change points that occur between the second and $(T-2)^{th}$ point.

The SIC algorithm does not require knowledge of the distribution of the test statistic. A modification of the method, more robust to data fluctuation, introduces a significant level α and its corresponding critical value C_α so that the null hypothesis is rejected if $SIC(T) \geq \min_{2 \leq k \leq T-2} SIC(k) + C_\alpha$. The value C_α can be determined such that

$$1 - \alpha = P \left[SIC(T) < \min_{2 \leq k \leq T-2} SIC(k) + C_\alpha | H_0 \right],$$

see Chen and Gupta [8].

3.1.1 The binary segmentation procedure

Methods described above were designed for location of single change points. In the application section we will use the binary segmentation procedure to test and locate multiple change points. At the first stage of the procedure we test the null hypothesis for the whole data. If we do not reject H_0 we declare that there is no change point in the whole sequence, otherwise we divide the data into two sub-sequences as determined by the change point located. At the second stage we test the two sub-sequences and repeat the above procedure until we do not find any further change point. Several candidate change points may result from this procedure. At the third stage we check these points as follows. For a given possible change point we determine the sub-sequence between the previous possible change point and the next change point and repeat the test. If we still reject H_0 we keep this point as a change point, otherwise we remove it from the list of candidates. This confirmatory step helps to reduce masking effect and to get more reliable change point estimates. Inclán and Tiao [7] describe this procedure in detail.

3.2 Multiple jumps detection: the Wang’s method

Wang’s algorithm [6] enables us to detect sudden jumps and sharp cusps in a time series by using discrete wavelet transforms. The idea is simple to understand: A sudden jump affects the magnitudes of wavelet coefficients, thus one can set a threshold level to identify the location at which the jump occurs. Wang suggested to apply the DWT to the data and use the universal threshold of Donoho *et al.* [20]

$$\begin{aligned} \text{Universal threshold } \lambda &= \hat{\sigma} \sqrt{2 \log n} \\ \hat{\sigma} &= 1.4826 \cdot \text{MEDIAN}[|d^{J-1} - \text{MEDIAN}(d^{J-1})|] \end{aligned}$$

where d^{J-1} is the vector of the finest wavelet coefficients of the wavelet transform and $\hat{\sigma}$ is the MAD estimate. Points above the threshold in absolute value are declared jump points.

4 Detection schemes

We implement the detection methods previously described using wavelet packet transformations. We use a moving window approach so that the methods can be used for online detection. We indicate these modified procedures as MWICSS (moving window ICSS), MWSIC (moving window SIC) and MWWJ (moving window Wang’s jump detection). Whitcher *et al.* [9] suggested that the sample size for the ICSS algorithm be at least 128 for better approximation. In the next section we investigate performances for several different window sizes. We use the same window lengths for the MWSIC, for a better comparison. For the MWWJ algorithm we also try smaller sizes.

Having chosen the length of the window, the data sequence is examined for change points by sliding the window along the data one point at the time and recording all change points detected. For all detection tests we use a 0.05 significance level. Detected points indicate network anomalies. We declare an anomaly to be a potential attack if it is detected by our procedures in a number of consecutive windows. In other words, we look at the detection frequency as the number of times the anomaly is detected and declare an attack if this exceeds a preselected threshold value. Our moving window

procedure and the calculation of the detection frequency is explained in Figure 2, where we use a square symbol to indicate whether the point is detected in a particular window. With a preselected threshold of 6 or higher the point in the figure would be declared an attack. The choice of the threshold implies a trade-off between fast detection and false alarms. Specifically, we want to detect changes as fast as possible after they occur but also want to avoid false alarms. As the threshold value increases we are able to avoid more and more false alarms but with an increase in the detection delay. In the analyses reported here we aimed at decreasing the detection delay for a given false alarm level and look at the mean delay as a performance measure for online detection.

We now give step-by-step descriptions of the implementations of the detection procedures we propose.

4.1 Procedure for variance change detection

In a generic window of size m we test for variance change points as follows.

- Step I: We apply the DWPT and MODWPT. The maximum level of the transforms depends on the length of window. Whitcher *et. al.* [9] recommend to use at least 128 data points to implement the variance change test. Moreover, we want to apply to the coefficients the Ljung-Box test for autocorrelation with maximum lag 10 (see step II). We therefore compute wavelet transforms up to level 4.
- Step II: The application of the MWICSS and MWSIC algorithms to test for variance changes requires uncorrelated data. We therefore choose the DWPT packet with highest P-value among those packets of the tree for which the null hypothesis of the Ljung-Box test for autocorrelation is not rejected. The statistic for this test is defined as

$$Q = m(m + 2) \sum_{k=1}^l \frac{\hat{\rho}^2(k)}{m - k},$$

where $\hat{\rho}^2(k)$ is a squared correlation coefficient at lag k and l is arbitrary chosen (see Ljung and Box, [21]). Here we use a lag of 10, since we use at most 150 data points at a time.

- Step III: We test for variance changes (with either the ICSS or the SIC algorithm) using the coefficients of the DWPT packet selected from Step II. If the null hypothesis that no variance change occurs is rejected then we identify the location of the change point using now the non-decimated wavelet packet coefficients of the packet selected in Step II.
- Step IV: Using the binary segmentation procedure we repeat Steps I-III with subsequent subseries until no further variance change point is found. In the case of the ICSS procedure we also perform the additional confirmatory step on all identified potential change points by using subseries of data between adjacent points, as suggested by Inclán and Tiao [7], see section 3.
- Step V: We record information of the type (t_j, f_j) where t_j is a time location and f_j is its frequency of detection, i.e. how many times a change at that point has been detected by the method up to the window under consideration. We declare a certain time point to be a variance change if its frequency of detection is greater than or equal to a predetermined threshold k . A smaller k implies faster detection but also a larger number of false alarms, see results in Section 6.

4.2 Procedure for jump detection

For jump detection we adapt the procedure suggested by Wang [6] to wavelet packets, specifically to MODWPT coefficients. This allows us to locate the jump points more precisely since the MODWPT is not subsampled.

In a generic window of size m we test for jumps in the data as follows.

- Step I: We apply the MODWPT up to level J .
- Step II: We compute a threshold value λ using the finest wavelet coefficients of the MODWPT (the wavelet coefficients of packet $[1, 1]$) according to the formula given in Section 3.2.
- Step III: We check wavelet coefficients and find those that exceed the threshold value. In general terms, resolution level j identifies the dyadic interval with width proportional to 2^{j-1} . Wang [6] pointed out that jumps are better detected

using relatively narrow widths. In our simulation study we found best detection performances when using the wavelet coefficients at levels 5 and 4. Among all packets at a given level, better performances were obtained at lower frequencies. Results we report here were obtained by considering the locations of the wavelet coefficient of packet [5,1] of the MODWPT for which the absolute value is larger than the threshold value λ . In case we have multiple points as jump points within a given window we choose the closest point to end point of the window. We declare a new jump point if the detected point is at least 20 points away from the jump detected in the previous window.

5 Simulation study

5.1 Purpose of the study

We performed a simulation study to better understand the relative performances of the iterated cumulative sum of squares (ICSS) and the Schwarz Information Criterion (SIC) algorithms. We simulated data and computed mean delays under several different settings. The aim of the study was to assess how two different factors, the window size and the variance ratio, affect the performance of the MWICSS and MWSIC algorithms. We also looked at the robustness of the distributional and model assumptions on the data.

5.2 Simulation Scheme

We simulated normal random sequences of length 250 with one change point in the variance located at point 201. For convenience we set the mean of the data to zero. We used four different variance ratios, one vs. four, four vs. one, one vs. sixteen, and sixteen vs. one. For each variance ratio we replicated the experiment 200 times. We adopted the same detection scheme that we used in the previous section. We looked at three different window sizes, 128, 140, and 150. For window size 128 we used windows sliding from point 74 to 114, from point 62 to 102 for window size 140, and from point

52 to 92 for widow size 150. We set the threshold level to 2, that is we recorded end points of windows where change points were detected for the second time. We measured detection delays as differences between the actual change point (the 201 data point) and the end points. We repeated this scheme for the different variance ratios under investigation. We looked at the mean delays and their standard errors from the 200 experiments as criteria for performance comparison.

5.3 Results for simulations

Results on normal data are reported in Table 1. We repeated the entire simulation with data from a Laplace distribution, see Table 2, and from an AR(1) process with normal errors, see Table 3.

Variance ratio: For increasing variance ratios (1 vs. 4 and 1 vs. 16 variance ratio), both MWICSS and MWSIC can capture change points with mean delay around 17 and 7 points, respectively, away from the end point of the analyzing window. Performances in the case of a one vs. sixteen ratio appear to be better than those for the case of one vs. four ratio. This is an obvious result since a bigger variance change should be easier to detect. In these cases the absolute mean delays are in general quite small. However, when the variance changes from large to small, for example from four to one or from sixteen to one, both algorithms show worse performances, with mean delays almost doubled. A variance change from large to small may take more time to be detected because of the bigger oscillations of the signal in the first part that tend to dominate over the latter part.

Window size: From all three tables we conclude that different window sizes do not affect the detection performance since the variations in detection delays are quite small. Given the reduction in computation time and in cost we suggest to use small window sizes.

MWICSS vs. MWSIC: Both methods show reasonably good performance in the increasing variance ratio cases for both normal and Laplace distributions. In the decreasing variance ratio cases, i.e. four vs. one and sixteen vs. one, we notice that the MWSIC performs better than the MWICSS for the case of a large difference between

the two variance values (16 vs 1). The MWSIC algorithm showed large differences in detection performance according to whether we used the additional checking procedure or not. Results here reported were obtained without this procedure. Similar comments apply to results obtained by generating data from an AR(1) process with normal errors. Here, in addition, we notice an improvement in the standard errors for both methods for the cases four vs. one and one vs. sixteen.

Mean delay: An another goal of the simulation study was to investigate how much we can reduce the detection delay. In the case of increasing variance ratios the best detections were 6-8 data points away from the end of the window. That is, we have to endure a 6-8 delay.

6 Analysis of network data

6.1 Network trace data

Kim *et al.* [5] suggest a new data structure for network anomaly detection. Their data structure is based on the concept of correlation between adjacent sampling periods. They use IP addresses and their packet counts from the packet header data. Their computation procedure intends to convert discrete type information into a continuous signal. Within a given sampling period (e.g. one minute) IP addresses and their packet counts are stored for all traffic flows. An IP address has four fields with word-size of 256 locations, that is, a total of 1024 words. For a given traffic flow its packet count is recorded at the number of each field of IP address. In order to obtain a signal, correlation numbers are computed for the four fields at a given sampling point as follows:

$$C_i(t) = \frac{\sum_{j=0}^{255} [\text{packet count}_j(t-1) \times \text{packet count}_j(t)]}{\sqrt{\sum_{j=0}^{255} (\text{packet count}_j(t))^2}} \quad \text{where } i = 1, \dots, 4.$$

The correlation signal is defined as:

$$S(t) = \alpha_0 + \alpha_1 \left(\sum_{i=1}^4 w_i C_i(t) \right), \quad \text{where } \sum_{i=1}^4 w_i = 1.$$

This linear transformation ensures that the signal lies in the range between zero and one hundred. We illustrate this procedure in Figure 2 with a simple example.

Kim *et al.* [5] analyze internet traffic traces from NLANR (National Laboratory for Applied Network Research). They apply the following sampling scheme: They sampled one minute of traffic to compute their correlation signal and then paused for one minute. The resulting correlation signal consists of 4,302 data points for a 3-day trace. These data were considered as an ambient trace, that is, without noticeable attacks against the network. They then simulated nine kinds of attacks with various behaviors, as motivated by recent SQL Slammer and Code Red attacks. The nine attacks were classified as follows:

(1) **Duration:** The first 6 attacks last for 2 hours, the remaining 3 attacks for 1 hour.

(2) **Persistence:** The first 3 attacks send malicious packets for 3 minutes and pause for 3 minutes. Such pattern is repeated through the attack duration. While the filtering may mitigate the overhead of the attacker's continuing scan traffic, a more sophisticated attacker might have stopped scanning and it may be possible to conceal attacker's intentions through repeating attack and pause periods. The other remaining attacks continue to assault throughout the attack period.

(3) **IP address:** The first attack among every 3 attacks targets a single destination IP address. In a hypothetical situation, the attackers target a famous site such as the White House, CNN or Yahoo, etc. This target may be really one host in case of 32-bit prefix, occasionally aggregated neighboring hosts in case of x-bit prefix. The 2nd attack style imitates from the IP address generation scheme of the notorious Code Red II worm. That is to say, a portion of addresses preserve the class-A and a partition of addresses preserve class-B for the infiltration efficiency. The 3rd type is a randomly generated address that was used for the Code Red I and SQL Slammer worm.

(4) **Protocol:** The 3 major protocols, ICMP, TCP, and UDP, are used in turn.

(5) **Port:** The second port among every 3 attacks targets randomly generated destination ports. It is useful to detect portscan that is used to probe a loosely defensive port. The first port is a representative #80 that stands for the reserved port for well-

known services. The third port is a #1434 that acts for the ephemeral client port, which is used in SQL Slammer worm.

(6) **Size:** There are three different byte counts of packets. The three denominations are random size, 4K bytes and 404 bytes.

The attacks can be described by a 3-tuple (duration, persistency, and IP address). These attacks were superimposed to the ambient traces from NLANR. The ratio of attack and normal traffic is 1:2 in packet counts. The resulting correlation signal is shown in Figure 4. We summarize the features of the nine attacks in Table 7. The first three attacks exhibit variance changes, while the other 6 show also sudden up and down jumps.

Figure 5 shows the sub-sequence of the data corresponding to the second attack. In the same figure we also report autocorrelation functions of the data, of the DWT wavelet coefficients at levels 2 and 3 and of two DWPT packets. This figure clearly shows the additional flexibility of the DWPT versus the DWT at decorrelating data.

6.2 Results

We examined several different combinations of the window size and the wavelet family. We used three different wavelet families, the Haar wavelets, Daubechies wavelets with 2 vanishing moments, and the least asymmetric wavelets with 4 vanishing moments (Daubechies, [22]). In order to reduce the number of false alarms we used the threshold approach as previously described, that is, we considered change points those for which the detection numbers are equal or greater than the threshold value. When computing detection delays we considered a change point successfully detected if a point that falls within 10 time points from the actual change point was detected by our procedure.

We report here results we obtained with Haar wavelets, which showed the best performances. We considered only 8 attacks, that is, 16 change points, among the 9 simulated. We ignored the last attack because of the moving window and threshold approach we adopted. Table 5 reports detection delays for 4 threshold values between 3 and 15. We measured the detection delay as the time difference between the actual change point and the earliest point detected by our procedure. Numbers in the first

column of Table 5 indicate the 16 change points (numbered from 1 to 16) that define starting and ending of the first to the eighth attack. For each threshold value we report results for three window sizes, for both MWICSS and MWSIC. For each combination of the parameters we also give average detection delays and the total number of detection as the total number of points detected.

In Table 6 we report detection delays for the MWWJ algorithm with four different widow sizes. The detection criterion for MWWJ is as follows. We set to 20 the gap size value to decide whether a jump occurs. For a given window size we find all locations at which the absolute value of the MODWPT coefficients exceeds λ (computed using the MODWPT coefficients of the finest level). Then we record the closest location to the end point of the window. We compare this location with the one of the previous window. When the difference between two points is equal to or greater than the predetermined gap size, this new point is declared as a jump point.

As expected, performances of the three different detection methods vary according to the attack type. MWWJ detects all 12 jump-type change points without delay while it shows worse performances in capturing variance change points (first three attacks, see Table 6), particularly for the first attack. Note that the 2nd and 3rd attacks are not “pure” variance change points. Indeed, they contain both a jump in the mean level as well as a variance change. As for the MWICSS and the MWSIC, performances are different for the single attacks. For the 1st, 2nd, and 3rd attacks the two methods show comparable behaviour, with a slight better performance of the MWICSS. For the 4th and 7th attacks the MWSIC does a better job at capturing the starting point while the MWICSS performs better in detecting the end point of the attack. MWSIC shows bad behaviour for the 5th attack, by missing it in most cases, and performs worse than MWICSS in the detection of the 6th attack. The 8th attack is a very difficult case to detect, although MWICSS with a small threshold does a decent job, even if with a considerable detection delay. As a general result, MWICSS may be preferable to MWSIC since it shows smaller mean detection delays. Here MWSIC was performed without the confirmatory step as additional checking procedure previously described because we noticed that including such additional checking would worsen the

performances of the MWSIC method. On the contrary, when used with the MWICSS algorithm the confirmatory step was beneficial.

Plots of Figure 6 give a graphical representation of the performances of the three detection methods. There, each of the two subplots contains a different portion of the signal, displaying 1st, 2nd, 3rd attacks and 4th, 6th and 8th attacks respectively, as representatives of the two different kind of change point, in mean and in variance. Results for MWICSS and MWSIC are for a threshold level 2 and window size 128 (see Table 5), those for MWWJ are for window size 128 (see Table 6). In these plots, the solid circles indicate the real change points, the square rectangles the points detected by the MWICSS, the diamonds those detected by the MWSIC, and the triangles those detected by the MWWJ. Notice how the MWICSS and MWSIC algorithms do a better job at detecting attacks of the first type, that show variance changes. However, there appears to be an asymmetric aspect in the detection of these two methods, in that both the MWICSS and the MWSIC detect the start of the attacks but show a relative large delay in detecting the ending points. In other words, these algorithms seem to be sensitive to the location of the change points and to the variance ratio, as already suggested by the simulation study of the previous section.

For online network attack detection, our results suggest that a simultaneous use of both MWICSS (or MWSIC) and the MWWJ algorithms give best results, allowing the detection of attacks of different types. Indeed, the average detection delay for all methods is 10.63 minutes. In addition, if we consider the starting points of the attacks only, as points of primary interest in network attack detection, the mean detection delay is 1.06 minutes, with a threshold level 2.

7 Concluding Remarks

The main goal of this paper was to develop an integrated tool for the detection of network anomalies and investigate performances using statistical analysis. We have proposed adaptations to wavelet packets of variance change detection methods and of a method for jump detection, and explored their implementation for online detection

of network anomalies. These methods can capture several types of attacks against the network.

References

1. Riedi R, Crouse M, Ribeiro V, Baraniuk R. Network Traffic Modeling using a Multifractal Wavelet Model. *IEEE International Symposium on Digital Signal Processing for Communication Systems (DSPCS)* February 1999.
2. Gilbert AC, Willinger W, Feldman A. Scaling analysis of random cascades, with applications to network traffic. *IEEE Transactions on Information Theory* 1999; **45**(3):971–991.
3. Gilbert AC. Multiscale analysis and data networks. *Applied and Computational Harmonic Analysis* 2001; **10**(3):185–202.
4. Resnick SG, Samorodnitsky G, Gilbert A, Willinger W. Wavelet analysis of conservative cascades. *Bernoulli* 2003; **9**(1):97–135.
5. Kim S, Reddy N, Vannucci M. Detecting Traffic Anomalies through Aggregate Analysis of Packet Header Data. *Proceedings of Networking* 2004; Athens, Greece.
6. Wang Y. Jump and Sharp Cusp Detection by Wavelets. *Biometrika* 1995; **82**(2):385–397.
7. Inclán C, Tiao GC. Use of Cumulative Sums of Squares for Retrospective Detection of Changes of Variance. *Journal of the American Statistical Association* 1994; 89:913–923.
8. Chen J, Gupta AK. Testing and Locating Variance Change-points with Application to Stock Prices. *Journal of the American Statistical Association* 1997; 92:739–747.
9. Whitcher B, Guttorp P, Percival DB. Multiscale Detection and Location of Multiple Variance Changes in the Presence of Long Memory. *Journal of Statistical Computation and Simulation* 2000; **68**(1):65–88.
10. Gabbanini F, Vannucci M, Bartoli G, Moro A. Wavelet Packet Methods for the Analysis of Variance of Time Series with Application to Crack Widths on the Brunelleschi Dome. *Journal of Computational and Graphical Statistics* 2004; **13**(3):639–

658.

11. Vidakovic B. *Statistical Modelling by Wavelets*. Wiley: New York 1999.
12. Mallat SG. A Theory of Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1989; 11:674-693.
13. Percival DB, Walden AT. *Wavelet Methods for Time Series Analysis*. Cambridge University Press: London, 2000.
14. Wickerhauser MV. *Adapted Wavelet Analysis from Theory to Software Algorithms*. A K Peters: Massachusetts 1994.
15. Percival DB, Sardy S, Davison AC. Wavestrapping time series: Adaptive wavelet-based bootstrapping. In *Nonlinear and nonstationary Signal Processing*, Fitzgerald BJ, Smith RL, Walden AT, Young PC (eds). Cambridge University Press: Cambridge, UK 2000.
16. Whitcher B. Simulating Gaussian stationary processes with unbounded spectra. *Journal of Computational and Graphical Statistics* 2001; **10**(1):112-134.
17. Whitcher B. Wavelet-based estimation for seasonal long-memory processes. *Technometrics* 2004; **82**(2):385–397.
18. Schwarz G. Estimating the dimension of a model. *Annals of Statistics* 1978; 6:461-464
19. Akaike H. A new look at the statistical identification model. *IEEE Transactions on Automatic Control* 1974; 19:716–723.
20. Donoho DL, Johnstone IM. Ideal spatial adaptation by wavelet shrinkage. *Biometrika* 1994; **81**(3):425–455.
21. Ljung GM, Box GEP. On a Measure of Lack of Fit in Time Series Models. *Biometrika* 1978; 65:297-304.
22. Daubechies I. *Ten Lectures on Wavelets*. SIAM: Philadelphia, 1992.

Authors' biographies

Deukwoo Kwon is a doctoral student in the Department of Statistics at Texas A&M University.

Kyungduk Ko is Assistant Professor in the Department of Mathematics at Boise State University. At the time of this research he was a doctoral student of Department of Statistics at Texas A&M University.

Marina Vannucci is Associate Professor in the Department of Statistics at Texas A&M University. Dr Vannucci earned her PhD from the University of Florence, Italy. Her research focuses on Bayesian Variable Selection, Classification and Clustering, Nonparametric Functional Estimation, Wavelet Methods in Statistics.

A.L. Narasimha Reddy is Professor in the Department of Electrical Engineering at Texas A&M University. Dr Reddy received his PhD from the University of Illinois at Urbana-Champaign. His research interests include Multimedia, I/O systems, Network QOS and Computer Architecture.

Seongsoo Kim is a doctoral in the Department of Electrical Engineering at Texas A&M University.

variance ratio	method		window size		
			128	140	150
1 vs. 4	MWICSS	mean	16.21	16.72	18.45
		std. err.	9.08	8.84	8.99
	MWSIC	mean	17.86	17.62	18.88
		std. err.	10.04	9.58	9.18
4 vs. 1	MWICSS	mean	32.83	31.76	32.92
		std. err.	6.49	5.60	3.94
	MWSIC	mean	31.48	31.26	31.24
		std. err.	6.50	6.78	6.74
1 vs. 16	MWICSS	mean	6.02	6.24	7.65
		std. err.	4.07	4.16	4.06
	MWSIC	mean	6.06	5.92	7.61
		std. err.	3.73	3.48	3.52
16 vs. 1	MWICSS	mean	35.05	34.88	34.97
		std. err.	4.65	4.84	3.82
	MWSIC	mean	22.24	21.96	23.71
		std. err.	6.17	6.06	5.88

Table 1: Summary of four variance ratios for MWICSS and MWSIC for normal distribution

variance ratio	method		window size		
			128	140	150
1 vs. 4	MWICSS	mean	16.57	17.06	19.71
		std. err.	9.07	8.98	9.55
	MWSIC	mean	19.38	18.60	19.76
		std. err.	10.55	9.51	9.23
4 vs. 1	MWICSS	mean	31.29	29.57	31.78
		std. err.	9.71	12.25	10.28
	MWSIC	mean	28.46	27.12	29.41
		std. err.	8.92	9.16	8.23
1 vs. 16	MWICSS	mean	6.69	6.85	8.46
		std. err.	4.58	4.25	4.54
	MWSIC	mean	7.23	7.05	8.85
		std. err.	5.45	4.93	5.15
16 vs. 1	MWICSS	mean	35.6	35.69	35.58
		std. err.	4.62	4.74	4.71
	MWSIC	mean	21.78	22.10	23.64
		std. err.	6.89	7.03	6.52

Table 2: Summary of four variance ratios for MWICSS and MWSIC for Laplace distribution

variance ratio	method		window size		
1 vs. 4			128	140	150
	MWICSS	mean	17.00	18.02	19.63
		std. dev.	8.77	9.40	9.13
	MWSIC	mean	19.16	19.47	21.33
		std. dev.	10.07	9.74	9.42
4 vs. 1	MWICSS	mean	36.29	30.00	31.25
		std. dev.	3.75	7.69	5.51
	MWSIC	mean	30.62	30.86	32.31
		std. dev.	7.23	8.37	6.94
1 vs. 16	MWICSS	mean	5.90	6.11	7.47
		std. dev.	3.37	3.54	3.36
	MWSIC	mean	6.26	6.10	7.89
		std. dev.	4.06	4.00	3.86
16 vs. 1	MWICSS	mean	34.78	34.57	35.28
		std. dev.	5.42	5.55	4.61
	MWSIC	mean	22.82	23.23	24.68
		std. dev.	6.56	6.54	6.56

Table 3: Summary of four variance ratios for MWICSS and MWSIC for AR(1) with normal errors ($\phi = -0.1$)

	1	2	3	4	5	6	7	8	9
Duration	2h	2h	2h	2h	2h	2h	1h	1h	1h
Persistence	intermittence	intermittence	intermittence	persistence	persistence	persistence	persistence	persistence	persistence
IP	single	semi-random	random random	single single	semi-random	random random	single single	semi-random	random random
Protocol	ICMP	TCP	UDP	ICMP	TCP	UDP	ICMP	TCP	UDP
Port	#80	random	#1434	#80	random	#1434	#80	random	#1434
Size	random	4KB	404B	random	4KB	404B	random	4KB	404B

Table 4: Description of nine simulated attacks

threshold		3						6						
window		128		140		150		128		140		150		
method		ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	
change	1	12	12	12	16	16	18	16	16	16	-	23	-	
	2	65	53	77	113	79	87	71	77	85	101	87	101	
	3	8	8	8	8	10	10	14	14	14	14	14	14	
	4	58	62	58	74	68	68	70	74	74	104	84	114	
	5	4	5	5	7	4	9	7	8	8	19	8	19	
	6	65	70	70	64	78	74	79	80	77	74	82	92	
	7	12	11	16	11	14	13	20	14	24	14	26	16	
	8	29	35	29	51	23	27	35	47	35	71	33	61	
	9	45	105	49	33	52	75	55	-	57	77	63	127	
	point	10	1	-	1	-	2	-	5	-	12	-	6	-
		11	38	16	20	28	18	102	52	88	50	104	52	111
		12	8	51	4	81	2	53	13	73	9	93	14	87
		13	-	9	-	7	-	9	-	41	-	45	-	47
		14	20	14	24	22	20	26	30	30	30	28	30	34
		15	64	-	110	-	116	-	-	-	-	-	-	-
		16	42	-	77	132	-	-	-	-	-	-	-	-
mean delay		31.40	34.69	37.33	46.22	35.86	43.92	35.93	65.64	37.77	74.45	40.31	90.17	
total points detected		107	141	106	137	104	136	55	63	58	66	60	64	

threshold		9						15						
window		128		140		150		128		140		150		
method		ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	ICSS	SIC	
change	1	27	31	27	-	33	-	-	-	-	-	-	-	
	2	77	97	97	122	93	111	116	118	132	-	138	-	
	3	18	18	18	18	20	20	31	-	-	-	-	-	
	4	80	118	82	118	92	126	94	-	110	-	118	-	
	5	11	22	12	27	12	25	19	-	18	-	18	-	
	6	108	94	88	92	94	100	121	-	126	-	131	-	
	7	26	17	30	17	33	23	-	32	-	28	-	28	
	8	64	51	60	77	58	75	89	-	103	-	109	-	
	point	9	61	-	63	-	69	136	-	-	-	-	-	-
		10	22	-	42	-	20	-	-	-	-	-	-	-
		11	62	101	63	120	19	127	-	-	-	-	-	-
		12	20	85	16	101	47	99	-	-	-	-	-	-
		13	-	52	-	64	-	99	-	-	-	-	-	-
		14	44	78	36	37	-	38	52	108	105	-	107	126
		15	-	-	-	-	-	-	-	-	-	-	-	-
		16	-	-	-	-	-	-	-	-	-	-	-	-
mean delay		47.69	60.30	48.77	62.60	49.17	78.08	71.15	86.00	99.00	28.00	108.17	77.00	
total points detected		54	61	57	66	58	64	32	39	28	12	29	13	

Table 5: Detection delays for MWICSS and MWSIC

window		100	128	140	150
change	1	-	16	16	16
	2	-	-	-	-
	3	6	6	6	6
	4	24	27	30	33
	5	7	7	7	7
	6	63	92	97	100
	7	0	0	0	0
	8	0	0	0	0
	9	0	0	0	0
	10	1	1	1	1
point	11	0	0	0	0
	12	0	0	0	0
	13	0	0	0	0
	14	0	0	0	0
	15	0	0	0	0
	16	0	0	0	0
mean delay		7.14	10.53	11	11.36
total points detected		27	28	28	23

Table 6: Detection delays for MWWJ

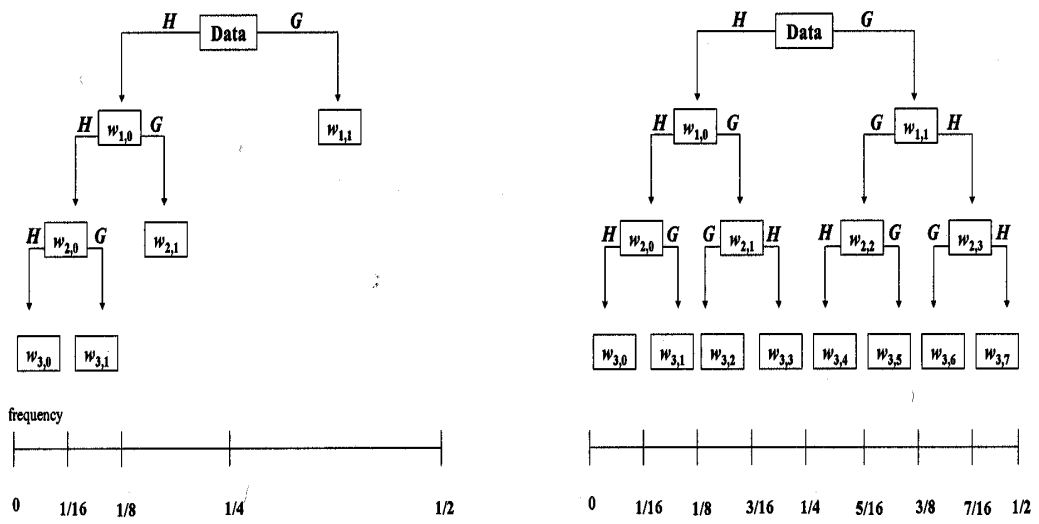
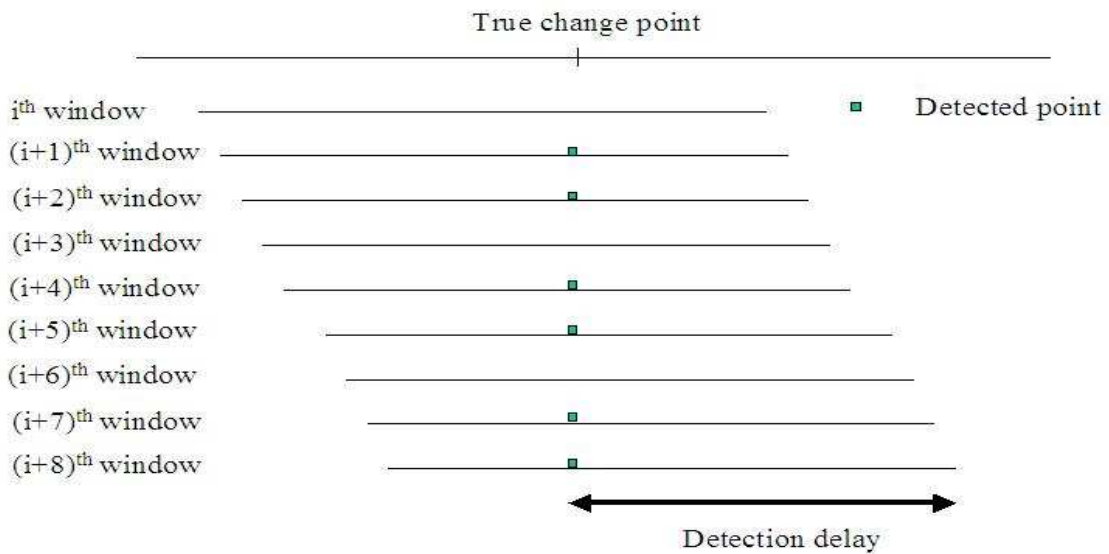


Figure 1: DWT and DWPT



Decision delay: the time of which detection frequency is equal to 6 (if k is set to 6) – true change point

Figure 2: Schematic representation of the moving window and detection frequency procedures

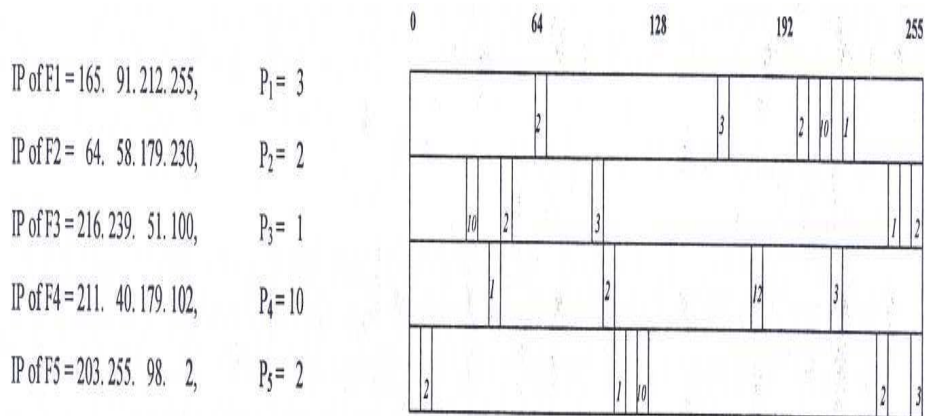


Figure 3: Data structure for computing the correlation signal (from Kim *et al.* (2004))

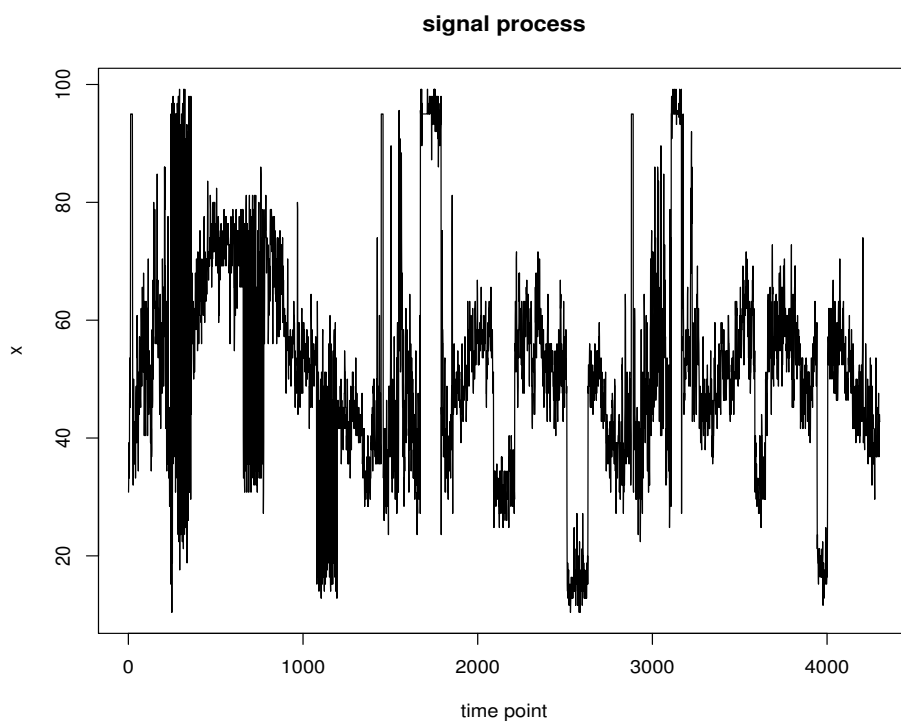


Figure 4: Correlation signal

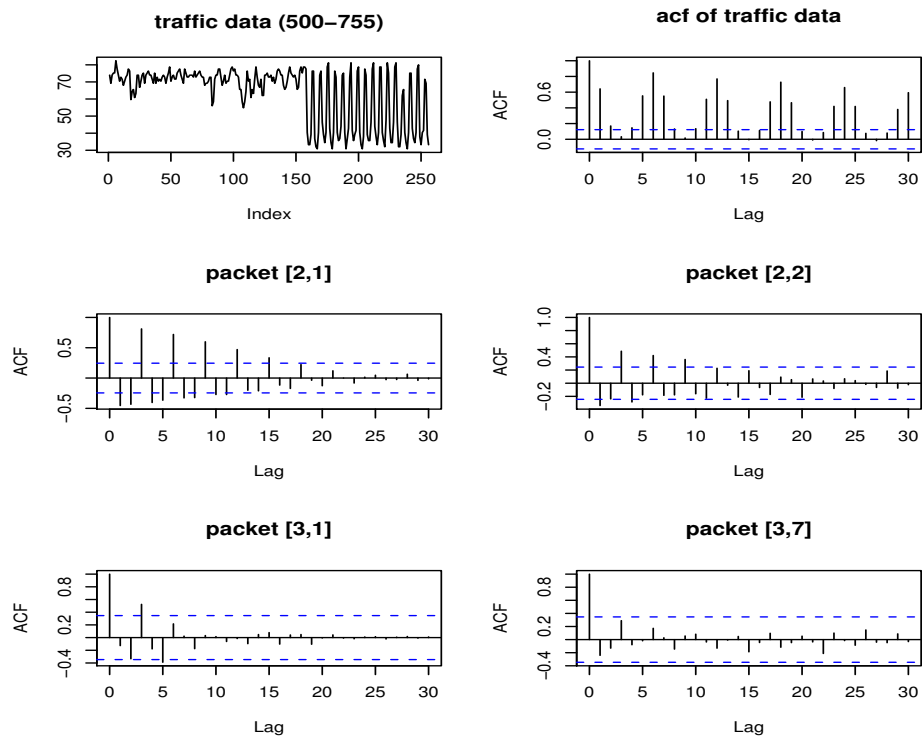


Figure 5: Attack n.2 with autocorrelation functions of the data, of the DWT wavelet coefficients at levels 2 and 3 and of two DWPT packets.

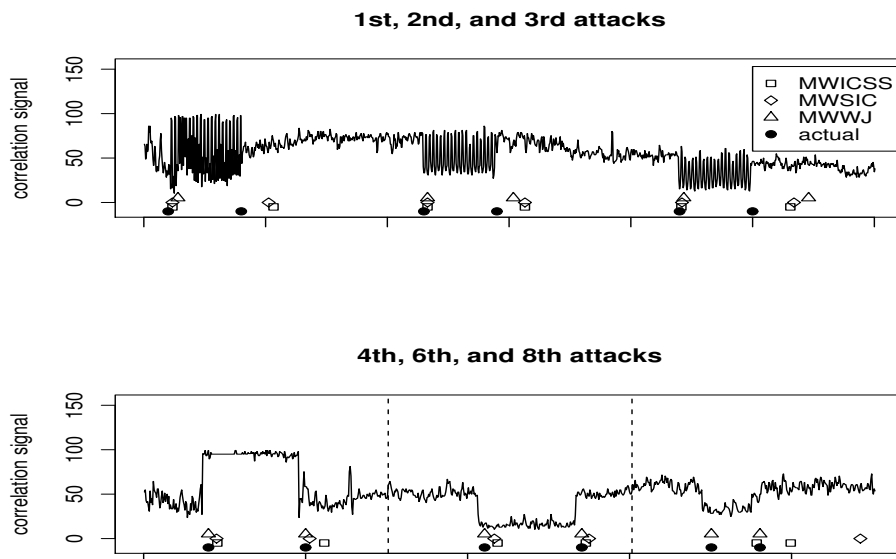


Figure 6: Performances of the three algorithms