Model Order Selection and Eigen Similarity based Framework for Detection and Identification of Network Attacks

Thiago P. B. Vieira[a], Danilo F. Tenório[a], João Paulo C. L. da Costa[a,b,c], Edison P. de Freitas[d], Giovanni Del Galdo[b,c], Rafael T. de Sousa Júnior[a]

[a] *Department of Electrical Engineering, University of Brasilia (UnB), 70910-900, Brasília-DF, Brazil*
[b] *Institute for Information Technology, Ilmenau University of Technology, Ilmenau, Germany*
[c] *Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany*
[d] *Graduate Program in Electrical Engineering, Federal University of Rio Grande do Sul (UFRGS), 90035-190, Porto Alegre, Brazil*

Abstract

Novel schemes for attack detection are crucial to identify adaptive malicious traffic coming from sources that are quickly mobilized by attackers in high throughput communication networks. In this context, signal processing techniques have been applied to attack detection due to their capability to detect anomalies that are previously unknown, i.e. blind detection. This paper proposes a signal processing framework for the detection and identification of network attacks using concepts of model order selection (MOS), eigenvalues and similarity analysis. In order to validate the proposed framework, we consider a network traffic sample that contains malicious activity such as denial of service (DoS) and port probing. We propose to model the network traffic as a superposition of components, namely, user's operations (legitimate traffic), network service operation not related to the user (noise) and the malicious activity. The experiments performed in a real network show that the proposed blind detection approach achieves satisfactory levels of accuracy in terms of timely detection and identification of TCP/UDP ports under attack.

Keywords: Network Attack Detection, Model Order Selection, Eigen Analysis, Similarity Analysis

1. Introduction

Traditionally, cyber defense methods can be effective against ordinary and conventional types of attacks, yet may fail against innovative malicious techniques [1]. In order to be able to detect and avoid novel attacks and their variations, it is necessary to develop or improve techniques to achieve efficiency on resource consumption, processing capacity and response time. Moreover, it is crucial to obtain high detection accuracy and capacity to detect variations of malicious patterns. Recently, signal processing schemes have being applied to the detection of malicious traffic in computer networks [2, 3, 4, 5, 6, 7], showing advances in network traffic analysis.

Information security may consist of both technical and procedural aspects. The former includes equipment and security systems, while the latter corresponds to security rules and recommendations. Intrusion detection and intrusion prevention systems are security systems used, respectively, to detect (passively) and prevent (proactively) threats to computer systems and computer networks. Such systems can work in the following fashions: signature-based, anomaly-based or hybrid [3, 8].

In the context of anomaly-based schemes, this work proposes an approach for the detection of malicious traffic in computer networks. Inspired by [5, 6], this work models the network traffic using a signal processing formulation as a composition of three components: legitimate traffic, malicious traffic and noise, taking into account the incoming and outgoing traffic in certain types of network ports (TCP or UDP). To the best of our knowledge there is no similar model in the literature. The proposed technique is based on eigenvalue analysis, model order selection (MOS) and similarity analysis. In contrast to [5, 6, 7], MOS and eigenvalue analysis are applied to detect time frames under attack. In addition, we also evaluate the accuracy of the proposed framework. Additionally, this proposed approach has its accuracy evaluation based on eigen similarity analysis for extracting detailed information about accurate time and network ports under attack.

The performed experiments show that synflood, fraggle and port scan attacks can be detected accurately and with great detail in an automatic and blind fashion, applying signal processing concepts for traffic modeling and through approaches based on MOS and eigen similarity analysis. The main contributions of the proposed framework are the capability to blindly detect time frames under network attack via MOS and eigen analysis, and the detailed identification of the network attack via eigen similarity analysis.

This paper is organized as follows. In Section 2, related works are discussed. Section 3 presents the data model. Section 4 describes the proposed framework for blind and automatic detection of malicious traffic. Section 5 discusses the experimental validation and presents the experimental results. Section 6 draws the conclusions and the suggestions for future work. The Appendix A presents mathematical concepts of examples of state-of-the-art MOS schemes.

2. Related Works

Several methods have been proposed for the identification and characterization of malicious activity in computer networks. Classical methods typically employ data mining [9, 10] and regular file analysis [11] to detect patterns that indicate the presence of specific attacks in network traffic.

Data mining is often used to describe the process of extracting useful information from large databases. Multiple methods of data mining are used in [9] to analyze data flow in a network with the aim of identifying characteristics of malicious traffic in large scale environments. Researchers have applied data mining techniques in log analysis [10] to improve intrusion detection performance. However, data mining techniques used so far in network analysis require prior collection of large data sets, which is a limitation of several schemes for online analysis [3].

Regular file analysis [11] consists of traffic analysis for detecting known patterns that indicate the presence of attacks, applying statistical analysis to the study of collected traffic. An essential feature of this method is that it depends

on prior knowledge of the details of the attacks to be identified, and also depends on previous log collection for traffic analysis and false positives reduction.

Principal Component Analysis (PCA) is a statistical technique commonly used for dimensionality reduction. It uses an orthogonal transformation to convert a set of correlated variables into a set of linearly uncorrelated variables, where the first principal components have the largest variance. PCA has been used in attack detection [12]. However, PCA requires human intervention in order to identify abnormalities based on the eigenvalues profiles, if used without complementary techniques. Besides being prone to higher errors and false positives, such human intervention makes PCA not useful for real time applications. Therefore, in order to automate the analysis of eigenvalues profile, model order selection (MOS) schemes should be incorporated.

Signal processing techniques have been successfully applied to network anomaly detection [2]. Lu and Ghorbani [2] proposed a network anomaly detection model based on network flow, wavelet approximation, and system identification theory. However, their work did not address problems without significant outliers, such as port scan attacks. Zonglin *et al* [4] proposed a method to detect traffic anomaly with correlation analysis, where the correlation between traffic signals and the predicted traffic signals are used to reveal anomalies. Zonglin *et al.* [4] evaluated the correlation analysis for anomaly detection, but the work was not applied to probing and denial of service (DoS) attack detection, simultaneously.

The data collected from honeypot systems, such as captured traffic and operating system logs, can be analyzed to obtain information about attack techniques, general trends of threats and exploits. Blind automatic detection of malicious traffic techniques have been developed for honeypots in [5, 6]. However, traffic on honeypot is simpler than real network traffic, because there are no running legitimate applications, due to the fact that honeypots emulate behavior of a host within a network to deceive and lure attackers [13]. Since honeypots do not generate legitimate traffic, the amount of data captured in honeypots is significantly lower in comparison to a Network Intrusion Detection System (NIDS), which captures and analyzes the largest possible amount of

network traffic [5]. MOS for blind identification of malicious activities in honeypots was proposed by us in [5], which evaluated criteria for selecting the model order, through simulations and comparing the order of the resulting model with the true model order.

Lee *et al.* [14] proposed OverSampling PCA (osPCA), which allows one to determine the anomaly of the target instance according to the variation of the resulting dominant eigenvector obtained by similarity analysis and over sampling. In contrast to Lee *et al.*, the framework applies MOS for detection of time frames under attack and similarity analysis to extract details for detection of time and ports under attack. Additionally, Lee *et al.* only evaluated their proposed scheme for covariance analysis, while we adopted an analysis based on sample covariance of zero mean variables and sample covariance of zero mean and unitary standard deviation variables, for DoS and probing attacks, respectively.

The proposed framework does not require either a significant amount of logs to detect attacks, nor prior data collection, in order to make comparisons and evaluate the existence of malicious traffic. The proposed solution is automatic and blind for detection of time frames under probing and DoS attacks through MOS and eigen analysis. Moreover, we apply eigen similarity analysis to identify details of time and ports under network attacks.

3. Data Model

In this paper, scalars are denoted by italic letters (*a, b, A, B,* α*,* β), vectors by lowercase bold letters (**a, b**), matrices by uppercase bold letters (**A, B**), and $a_{i,j}$ denotes the (*i, j*) elements of the matrix **A**. The superscripts $^{\mathrm{T}}$ and $^{-1}$ are used for matrix transposition and matrix inversion, respectively. We define the operator diag$(\cdot)$ that returns the vector of the main diagonal of a given matrix, the operator $\rightarrow$, which denotes the deletion of a given element from a set and the operator #, that returns the rank of a matriz, and the operator $\sim$ that sorts the elements of a vector in ascending order.

This section presents details of the simulated scenario along with a description of the dataset model as a signal superposition of legitimate traffic, noise and malicious traffic. Subsection 3.1 describes the environment and scenario adopted in order to reproduce DoS and probing attacks. Subsection 3.2 presents how network traffic can be modeled as signal superposition, and Subsection 5.1 details the traffic of synflood, fraggle and port scan attacks.

3.1. Analyzed Scenario and Data Collection

The environment of the analyzed scenario is composed of two computers and one router with access to the Internet and to an internal network, where the simulation of legitimate traffic, noise, DoS and port scan attacks are performed. During the traffic generation, one computer assumes the role of the attacker, while the other is the victim, according to scenario represented by Figure 1.

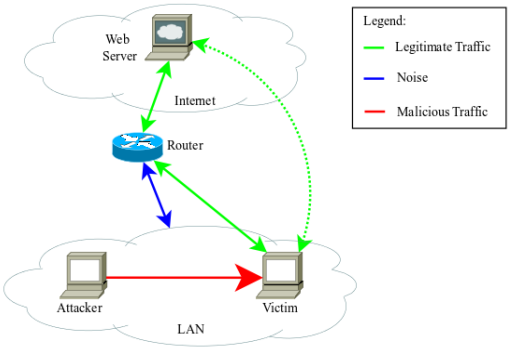

Figure 1: Scenario to reproduce legitimate traffic, noise, DoS and port scan.

Note that the set of network traffic is modeled as legitimate, noise and malicious traffic, where the victim performs legitimate activities, that can be characterized by web access. In many organizations this type of traffic is predominant, since most of corporate services are web-based, such as: web pages, customized web-based systems and cloud services. It is possible to characterize the traffic of a DHCP service as an example of noise associated with the transport layer. For malicious traffic, three types of networks attacks are evaluated: synflood, fraggle and port scan. These attacks are reproduced using well-known security

tools, such as Nmap[1] to port scan, Metasploit[2] to synflood and Hping[3] to lead the fraggle attack.

A network traffic log is commonly formed by timestamp, protocol, source IP address, source port, destination IP address, destination port and additional information, according to the type of transport protocol used. The following TCP traffic log is presented in order to exemplify the collected data:

```
21:00:34.099289 IP 192.168.1.102.34712 > 200.221.2.45.80: Flags
[S], seq 2424058224, win 14600, options [mss 1460, sackOK,TS
val 244136 ecr 0,nop,wscale 7], length 0
```

and the following to exemplify UDP traffic log:

```
21:24:42.484858 IP 192.168.1.102.68 > 192.168.1.1.67: BOOTP/DHCP,
Request from 00:26:9e:b7:82:be, length 300
```

In the proposed framework, the goal is to detect the anomalies only taking into account the traffic profile, i.e., specific information such as origin IP or day and time of the attack are not considered. Therefore, from the entire log information, we just consider the timestamp (for sequencing), port type and port number.

3.2. Modeling Data

By modeling the dataset as a signal superposition, the network traffic ($\mathbf{X}$) can be characterized as a mixture of three components: legitimate traffic ($\mathbf{S}$), noise ($\mathbf{N}$) and malicious traffic ($\mathbf{A}$), according to the following expression:

$$\boldsymbol{X}^{(q)} = \boldsymbol{U}^{(q)} + \boldsymbol{N}^{(q)} + \boldsymbol{A}^{(q)}, \tag{1}$$

[1] http://nmap.org
[2] http://www.metasploit.com
[3] http://hping.org

where q represents the q-th time frame, which is a time grouping of network traffic. The matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$ consists of M rows and N columns. Each row represents a communication port (TCP port or UDP port), and each column represents time bins having a approtiate size, such as one minute. Each element $x_{m,n}^{(q)}$ stands for the number of times that the port m appears at the n-th minute, at the q-th time frame.

The legitimate traffic $\boldsymbol{U}^{(q)}$ is characterized by the traffic from user's operations. When a user accesses a web page, for example, there is the corresponding TCP/IP traffic to request the page, as well as there is the traffic required to domain name resolution. Figure 2 depicts an example of the legitimate traffic obtained during experiments.

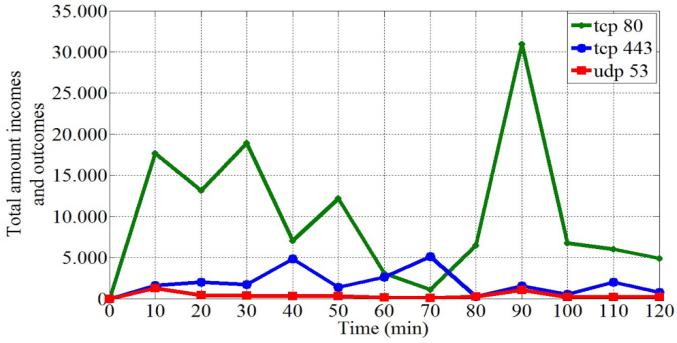

Figure 2: Traffic from user's operations, that can be characterized by web access, traffic of well-known applications or network protocols.

The traffic that is not associated with user's operations and with malicious traffic is modeled as noise $\boldsymbol{N}^{(q)}$. The automatic acquisition service of logical IP network address (DHCP) is an example of noise. Independently of any user operation, the machine receives an IP address, since it is configured to perform a DHCP address request. Figure 3 depicts an example of noise in a network traffic, represented by traffic to ports 67 and 68.

The traffic coming from a malicious activity, such as a synflood or fraggle attack, is represented by the matrix $\boldsymbol{A}^{(q)}$. For this work we only consider the traffic from port scanning and flood attacks.

We define that if the obtained $\#\boldsymbol{A}^{(q)} \neq 0$, then there is malicious traffic in

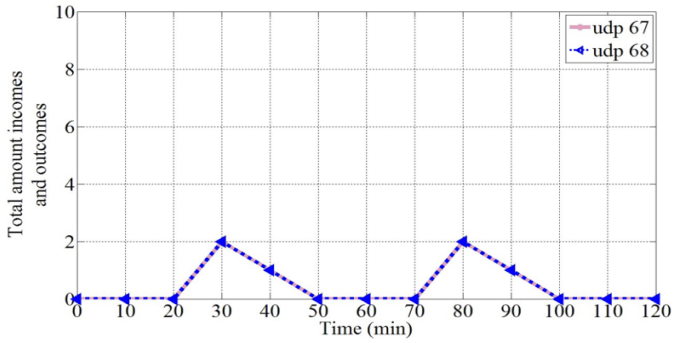

Figure 3: Network traffic of user independent operations for network management.

the evaluated time frame q, on the other hand, if the $\#\boldsymbol{A}^{(q)} = 0$, then there is no malicious traffic. This paper shows how to detect the $\#\boldsymbol{A}^{(q)}$, given only the matrix $\boldsymbol{X}^{(q)}$, in order to identify malicious network traffic.

3.3. Synflood, Fraggle and Port scan

The network attacks evaluated by this work are: synflood, fraggle and port scan. The first two attacks can be qualified as DoS attacks, while the last one can be qualified as probing or port scanning attack.

With respect to the synflood attacks, the attacker sends a large quantity and concurrent successive SYN requests to a target, in order to consume resources and cause a DoS. Figure 4 depicts an example of a synflood attack carried out in a real computer network. In an interval of ten minutes, more than 210,000 packets are sent as a synflood attack. This network traffic behavior can be considered an abnormal behavior of network traffic, especially since it is concentrated in a short period of time and presents similar outstanding traffic during the time under attack.

With respect to the fraggle attack, large packets with UDP echo segments are sent to the broadcast address of a network. Every packet is modified to have the source address of the victim, in order to implement the source address spoofing technique. Therefore, each host receives a huge amount of requests UDP echo and all of them replies to the IP address of the victim, causing a packet flooding aiming a DoS. This attack can affect the entire network, since all hosts receive

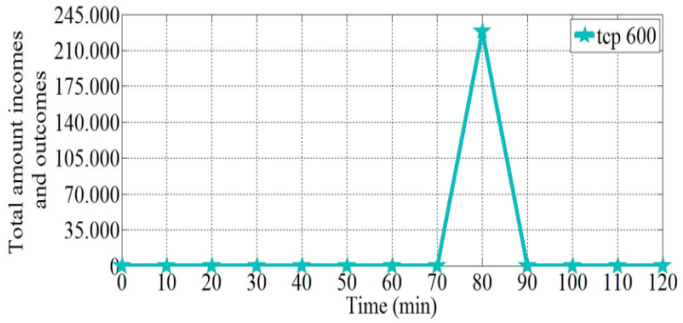

Figure 4: A large quantity of SYN requests to a target, in order to cause a DoS.

several requests UDP echo and respond with the ICMP protocol, therefore each host acts as an amplifier of the attack. This last part of the fraggle attack is not taken into account in this work, because the victim receives ICMP (network layer) packets originated from the hosts that are attacked with flooding packet UDP echo. Figure 5 depicts an example of the fraggle attack in a real computer network.

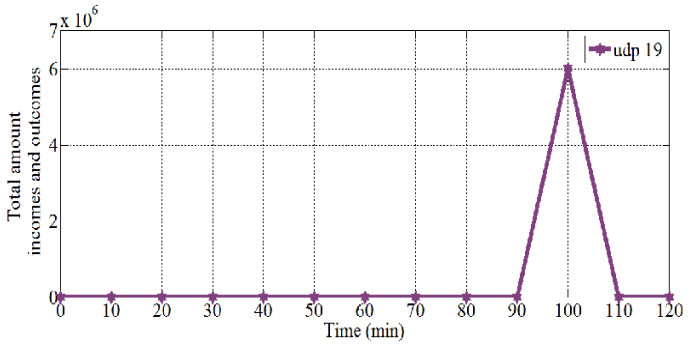

Figure 5: Large amount of "UDP echo" requests and replies, causing packet flooding.

More than 6,000,000 malicious packets can be counted in an interval of ten minutes, which can be considered an abnormal network traffic, especially due to the concentrated traffic in a short period of time and due to the similarity of the outstanding traffic.

Port scan is the attempt to establish a connection to TCP and UDP ports to identify what services are running or are in the listening state. There are several

available port scanning techniques, including: TCP SYN scan, TCP ACK scan and UDP scan. This work evaluates the use of TCP SYN scan and UDP scan.

In TCP SYN scan, a SYN packet is sent to the destination and two types of responses may occur: SYN/ACK or RST/ACK. In the first case, the destination port is in the listening state, in the second case, the destination port is not listening. At the end of each port scanning, a RST/ACK packet is sent by the system that is performing the port scan. Therefore, a full connection or a complete three-way handshake is never established, which makes the detection of the attack sender more difficult, and requires approaches able to identify probing attacks without connection stablishment. The UDP scan technique sends UDP packets to the destination port, and if it responds with a *ICMP port unreachable* message, then it indicates that the scanned port is closed. On the other hand, if a message is not received, then the port is considered as open.

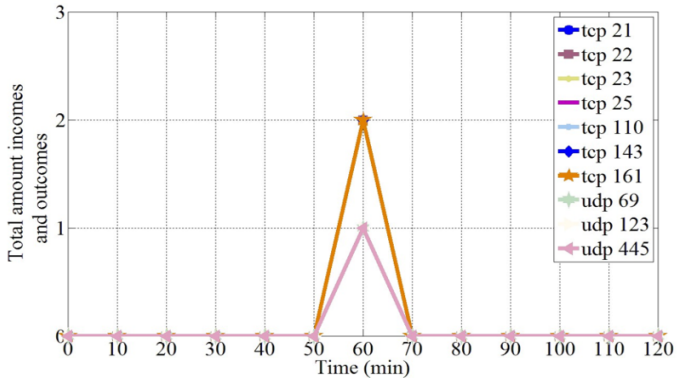

Figure 6: Connection attempts in order to identify active ports.

Figure 6 depicts an example of the port scan attack in a real computer network. Note that the traffic is composed of two packets for each TCP port and one UDP packet to each port. The incoming and outgoing packets analysis, for each port, shows the high correlation and similarity of TCP and UDP traffic during the simulated port scan attack.

4. Proposed Framework for Detection and Identification of Network Attacks

This section describes the proposed technique to detect synflood, fraggle and port scan, according to Figure 7, which represents the overview of the proposed framework for detection and identification of network attacks. In Subsection 4.1 we present the steps for extraction of the largest eigenvelue for each q-th time frame. Next, in Subsection 4.2, we show how to apply the eiganvalues on the MOS scheme in order to detect the attack. In Subsection 4.3, we present the eigenvalue analysis to identify the time frames detected as under attack, and the Subsection 4.4 describes the similarity analysis evaluated for detailed attack identification.

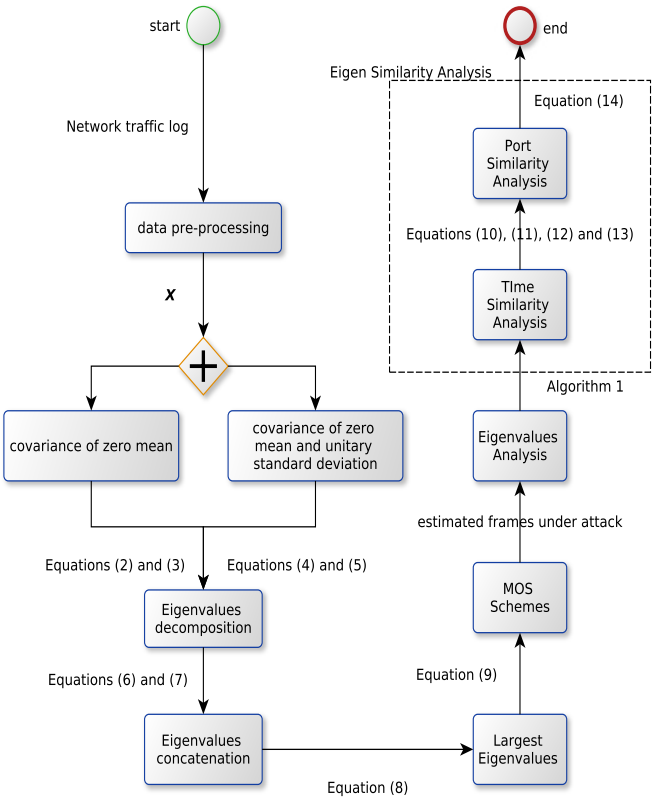

Figure 7: Overview of The Framework for Detection and Identification of Network Attacks.

4.1. Largest Eigenvalue by Time Frames

The proposed attack detection algorithm starts by the data pre-processing of a network traffic log containing IP, ports and timestamp of senders and receivers. During this step, the desired information is extracted in order to classify and count packets according to the origin and destination ports, and subsequently this information is grouped by minutes and by time frames.

With the data grouped into Q time frames, the framework considers the time variations of the matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$, with $q = 1, \ldots, Q$, in order to detect the attack.

According to DoS and port scan attacks' behavior, DoS attacks and port scan attacks can be characterized as covariance aware attack [15] and correlation aware attack [1], respectively. These characteristics are substantiated by the results obtained through the analysis based on sample covariation of zero mean variables and on covariance of zero mean and unitary standard deviation variables, described in Section 5, which shows that the main components of DoS attacks are dominated by the variables with more variance and that the traffic associated with port scan attack does not generate many logs, however, it presents high covariance of zero mean and unitary standard deviation variables.

Therefore, to detect DoS attacks, it is necessary to calculate the sample covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ of the zero mean samples given by

$$\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}. \tag{2}$$

The set of obtained vectors $\boldsymbol{y}_m^{(q)}$ composes the zero mean matrix $\boldsymbol{Y}^{(q)}$, then the sample covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ can be calculated as follows

$$\hat{\boldsymbol{R}}_{yy}^{(q)} = \frac{1}{N} \boldsymbol{Y}^{(q)} \boldsymbol{Y}^{(q)^{\mathrm{T}}}. \tag{3}$$

For the detection of the port scan attack, the main components are not dominated by the variables with large variance. Moreover, the portscan traffic presents a highly correlated network traffic. In order to exploit such structure,

we compute the sample covariance $\hat{\boldsymbol{R}}_{zz}^{(q)}$ whose variables have zero mean and unitary standard deviation as follows

$$\boldsymbol{z}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}}. \tag{4}$$

The set of vectors $\boldsymbol{z}_m^{(q)}$ composes the matrix $\boldsymbol{Z}^{(q)}$, then the sample covariance matrix $\hat{\boldsymbol{R}}_{zz}^{(q)}$ can be calculated via

$$\hat{\boldsymbol{R}}_{zz}^{(q)} = \frac{1}{N}\boldsymbol{Z}^{(q)}\boldsymbol{Z}^{(q)^{\mathrm{T}}}. \tag{5}$$

Once the $\hat{\boldsymbol{R}}_{yy}^{(q)}$ and $\hat{\boldsymbol{R}}_{zz}^{(q)}$ have been obtained for DoS and port scan attack detection, respectively, and since the next steps are the same for both sample covariance matrices, we refer to $\hat{\boldsymbol{R}}_{yy}$ and $\hat{\boldsymbol{R}}_{zz}$ as a matrix $\boldsymbol{C}$. Therefore, the following step of the algorithm is the eigenvalue decomposition (EVD), calculated according to (6), in order to obtain the vector of eigenvalues $\boldsymbol{e}^{(q)}$ associated with each matrix, according to (7).

$$\boldsymbol{C}^{(q)} = \boldsymbol{V}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{V}^{(q)^{\mathrm{T}}}, \tag{6}$$

$$\boldsymbol{e}^{(q)} = \mathrm{diag}(\boldsymbol{\Lambda}^{(\mathrm{q})}), \tag{7}$$

where the operator $\mathrm{diag}(\cdot)$ extracts the main diagonal of a matrix.

The eigenvalues should be sorted in descending order, i.e., $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > ... > \lambda_m^{(q)}$. Therefore, the largest eigenvalue of the q-th time frame evaluated for the attack detect is given by $\lambda_1^{(q)}$.

The concatenation of the eigenvalues vector $\boldsymbol{e}^{(q)}$ for $q = 1, \ldots, Q$ is repre-

sented by

$$\boldsymbol{E} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}. \tag{8}$$

Note that since $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, then the first line of the matrix $\boldsymbol{E}$ contains the largest eigenvalues of each q-th time frame, which is the Greatest Eigenvalue Time Vector (GETV) [7], denoted as

$$\boldsymbol{e}_{\max} = \boldsymbol{E}\{:, 1\} = [\lambda_1^{(1)}, \lambda_1^{(2)} ... \lambda_1^{(Q)}] \tag{9}$$

4.2. MOS Schemes

Traditionally the MOS schemes are applied for the eigenvalues of the vector $\boldsymbol{e}^{(q)}$. However, the goal here is to detect the variations of the eigenvalues for different values of q. Therefore, instead of using a certain q, the proposed approach applies MOS schemes for a vector of the largest eigenvalues of each q-th time frame, in order to identify variations and estimate the model order $\hat{d}$, which is the estimated number of time frames under attack. Therefore, $\boldsymbol{e}_{\max}$ is sorted in descending order, producing $\sim \boldsymbol{e}_{\max}$, that is used as input parameter for MOS schemes, according to $\hat{d} = \text{MOS}(\sim \boldsymbol{e}_{\max})$. Note that some MOS schemes may also require the number of minutes that compose a time frame, as $\hat{d} = \text{MOS}(\boldsymbol{e}_{\max}, Q)$. For more information about MOS, we refer to Appendix A.

In our previous work [7], the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE schemes are evaluated for synflood and port scan attack detection, showing that EDC and EFT are effective for detecting this kind of attacks. The present work extends that evaluation to also analyse the effectiveness of the listed MOS schemes for fraggle attack detection, as shown in Section 5.

4.3. Eigenvalue Analysis

After applying the MOS schemes to the vector $\sim \boldsymbol{e}_{\max}$, we obtain the estimate of the $\#\boldsymbol{A}$. For instance, in the case of fraggle, synflood and portscan, if $\hat{d} = 1$, , then $\#\boldsymbol{A} = 1$, which means that during the during the Q time frames one attack is present. However, if $\hat{d} = 0$, then $\#\boldsymbol{A} = 0$, and this means that none of these attacks are present. Note that $\hat{d}$ can be greater than 1, indicating the presence of more than one attack.

In Subsection 4.2, we obtained only if $\hat{d} = 1$ or $\hat{d} = 0$. However, if $\hat{d} = 1$, the MOS schemes do not provide any information about the q-th time frame under attack. The identification of the q-th time frame under attack can be carried out through a eigenvalues analysis.

The largest eigenvalue analysis for estimating the q-th time frames that are under attack can be expressed according to Algorithm 1, where $\hat{\boldsymbol{q}}_{\max} \in \mathbb{R}^{\hat{\mathrm{d}}}$ denotes a vector of the q-th time frames under attack, which is the q-th indexes corresponding to the $\hat{d}$ largest eigenvalues of $\boldsymbol{e}_{\max}$. Algorithm 1 initially identifies the largest value of $\boldsymbol{e}_{\max}$, according to Line 2 of Algorithm 1, and its correspondent index, according to Lines 4 and 5 of Algorithm 1. Subsequently, the largest value is removed of $\boldsymbol{e}_{\max}$, according to Line 8 of Algorithm 1, and a new iteration is performed until $\boldsymbol{e}_{\max} = \{\}$.

Algorithm 1 Detection of Time Frames Under Attack

1: **loop** $f = 1$ until $f == \hat{d}$
2: $\quad q_{\text{value}} = \arg\max_{\lambda} \ \boldsymbol{e}_{\max}$
3: $\quad$ **loop** $i = 1$ until $i == Q$
4: $\quad\quad$ **if** $\boldsymbol{e}^{(i)}_{\max} == q_{\text{value}}$ **then**
5: $\quad\quad\quad \hat{\boldsymbol{q}}^{(f)}_{\max} = i$
6: $\quad\quad$ **end if**
7: $\quad$ **end loop**
8: $\quad \boldsymbol{e}_{\max} \rightarrow \hat{\boldsymbol{q}}^{(f)}_{\max}$
9: **end loop**

After the estimation of the $\hat{\boldsymbol{q}}_{\max}$ time frames under attack, it is necessary to obtain more details of the detected attacks, such as the n-th minutes when the attacks happened and the m-th network ports that were attacked. To deal with

this problem, the adoption of a similarity analysis between legitimate traffic and the traffic of time frames estimated as under attack is evaluated, analysing the effectiveness of cosine similarity to highlight abnormalies inserted by network traffic attacks.

4.4. Eigen Similarity Analysis

Cosine similarity calculates the cosine of the angle between two vectors, which represents the similarity of values between the selected vectors. Therefore, cosine similarity can be used to evaluate the variation of the most significant eigenvectors of $\boldsymbol{V}^{(q)}$ against the the most significant eigenvectors of time frame detected as under attack, to analyse similarity changes into the most significant eigenvectors caused by the insertion of anomalous traffic [14].

This subsection describes the proposed eigen similarity analysis for detailed attack identification, in complement to the attack estimation carried out through MOS schemes and eigenvalue analysis. In Subsection 4.4.1 we present the eigen similarity analysis for identification of time under attack. Next, in Subsection 4.4.2, we show how to apply the eigen similarity analysis in order to identify network ports under attack.

4.4.1. Time Similarity Analysis

For eigen similarity analyis, we evaluate the cosine similarity in order to identify lacks of similarity between legitimate and malicious traffic, as follows

$$s_n = \frac{|\boldsymbol{v}^{(q)} \cdot \boldsymbol{v}_{(n)}|}{\|\boldsymbol{v}^{(q)}\| \|\boldsymbol{v}_{(n)}\|}, \tag{10}$$

where s_n denotes the absolute similarity degree of the n-th minute, $\boldsymbol{v}^{(q)}$ is the most significant eigenvectors of a selected set of minutes without network attack, and $\boldsymbol{v}_{(n)}$ is the most significant eigenvectors obtained after append the target n-th minute of traffic to be performed the DoS and port scan attack identification.

The most significant eigenvector $\boldsymbol{v}^{(q)}$, of a time frame q without attack, can be derivated from (6) and selected according to the eigenvector of the largest eigenvalue $\lambda_1^{(q)}$, which is the principal component of the evaluated matrix. The

same calculation shall be performed in order to obtain the target eigenvectors $\boldsymbol{v}_{(n)}$, calculated from a time frame without attack plus minutes of a time frame estimated as under attack, to evaluate the occurrence of network attacks.

The reference eigenvectors $\boldsymbol{v}^{(q)}$ is calculated from the traffic whithout attack, from a time frame q composed of Q minutes of legitimate network traffic. For the detailed attack identification, each $\boldsymbol{x}_{(n)}^{(\hat{q})}$ vector of each n-th minutes of the estimated $\hat{\boldsymbol{q}}_{\max}$ time frames shall be individually appended into $\boldsymbol{X}^{(q)}$, as represented by

$$\boldsymbol{X}_n = \{\boldsymbol{X}^{(q)} | \boldsymbol{x}_{(n)}^{(\hat{q})}\}. \tag{11}$$

The resultant $\boldsymbol{X}_{(n)}$ is necessary to obtain $\boldsymbol{v}_{(n)}$, through (6), for calculating the similarity degree s_n, ranging from 0 to 1, for each n-th minute. The s_n denotes the absolute similarity degree of the n-th minute in comparison to a well-known traffic without attack, detected through MOS schemes and eigenvalue analysis.

The incremental approach for similarity analysis is based on the incremental appending of network traffic into $\boldsymbol{X}^{(q)}$, where the first evaluation is based on (11) and the subsequent evaluations is based on (12), incrementally appending each n-th minute until $n = N$.

$$\boldsymbol{X}_n = \{\boldsymbol{X}_n | \boldsymbol{x}_{(n)}^{(\hat{q})}\}, \tag{12}$$

Figure 8 ilustrates the network traffic selection for the incremental approach of eigen similarity analysis, where the $\boldsymbol{X}^{(1)}$ is chosen as reference for similarity analysis of the m-th minutes of the time frame $q = 3$, where one network attack was previously detected.

The eigen similarity analysis starts at $\boldsymbol{x}_{(41)}^{(3)}$ and is incrementally performed until $\boldsymbol{x}_{(60)}^{(\hat{3})}$, in order to calculate the s_n. We assume that $s_n < l$ means an attack identification, according the anomaly on similiarity of s_n to a defined limiar l. Therefore, after obtaining the most significant eigenvector $\boldsymbol{v}^{(q)}$ and the target eigenvectors $\boldsymbol{v}_{(n)}$ for eigen similarity analyis, the s_n is calculated

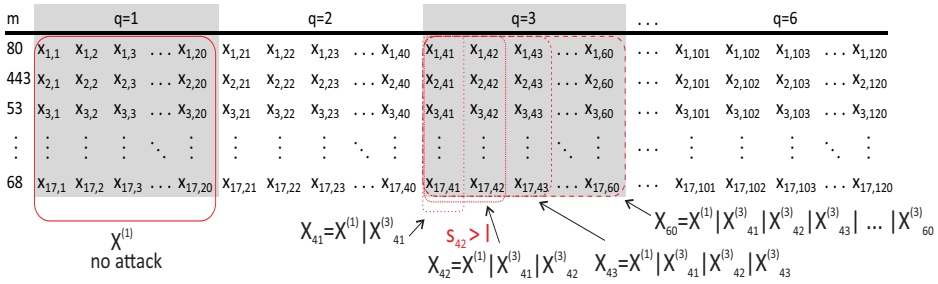

Figure 8: Traffic selection for incremental approach.

according to (10).

If $s_n = 1$, then the two eigenvectors are completely similar and no anomaly is detected. Smaller values of s_n mean less similarity and can indicate an anomaly, according to a defined threshold l, assuming that if $s_n < l$, then a network attack is identified during the n-th minute. Therefore, the s_n of each n-th minute shall be compared with the threshold l to evaluate if a attack was identified, according to

$$\hat{\boldsymbol{n}}_{(\boldsymbol{n})} = \begin{cases} 1, & \text{if } s_n < l \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\hat{\boldsymbol{n}}_{(n)}$ denotes a vector of n-th minutes detected as under attack.

The eigen similarity analysis can also be applied in an individual fashion, where each n-th minute must be individually appended into $\boldsymbol{X}^{(q)}$, as shown by Figure 9.

The incremental and the individual approaches can be combined to obtain the incremental individualized approach, where each minute is incrementally appended into the selected $\boldsymbol{X}^{(q)}$ for obtaining $\boldsymbol{v}_{(n)}$ to similarity analysis of the n-th minute, until detect the first n-th minute under attack. Subsequently, $\boldsymbol{X}_n$ becomes the new reference of traffic without network attack and each subsequent minute must have its similarity individually evaluated, as shown in Figure 10.

This approach of incremental similarity analysis followed by individual anal-

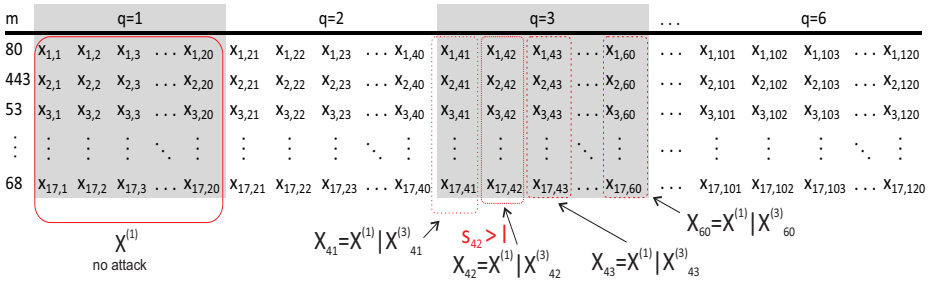

Figure 9: Traffic selection for individual approach.

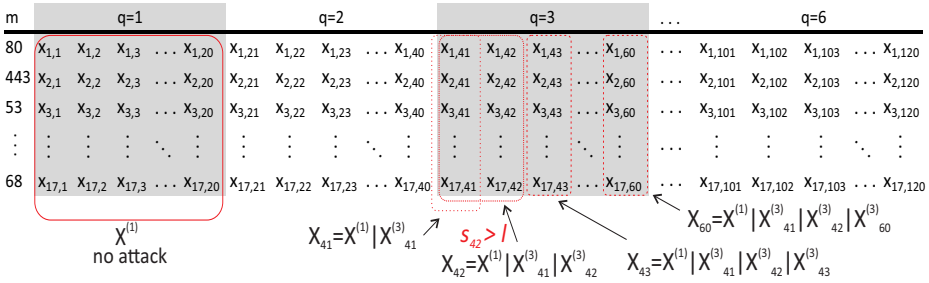

Figure 10: Traffic selection for incremental individualized approach.

ysis after an attack detection allows to identify the attack period, highlighting the first and last time under attack. This identification is possible due to the variation of the most significant eigenvectors, which becomes more significant when compared a traffic under attack against a traffic with no attack, according to results which are discussed in Section 5.

4.4.2. Port Similarity Analysis

Given $\hat{\boldsymbol{n}}$, which is the set of n-th minutes under attack, it is still necessary to obtain more details about the identified network attack, such as the network ports that are attacked during each n-th minute identified as under attack. Hence, it is also applied the cosine similarity analysis to identify variation of the most significant eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port during a n-th minute.

For detection of ports under attack, the $\boldsymbol{v}^{(q)}$ last most significant eigenvectors without attack shall be used as reference for similarity analysis against the $\boldsymbol{v}_{(n)}$ identified as under attack, individually evaluating the cosine similarity of each m-th port of all $\hat{\boldsymbol{n}}$ minutes.

Therefore, $\boldsymbol{v}^{(q)}$ should be calculated from the last $\boldsymbol{X}^{(q)}$ time frame without attack, and $\boldsymbol{v}_{(m,\hat{n})}$ should be calculated from the same traffic appened of all n-th minutes until the identified minute under attack, denoted as $\boldsymbol{X}_n$.

For similarity analyis, each m-th port of the last n-th minute of $\boldsymbol{X}_n$, denoted as $x_{(m,n)}$, shall be individually replaced by the traffic of the evaluated m-th port of the $\hat{n}$-th minute under attack, denoted as $x^{(\hat{q})}_{(m,\hat{n})}$, in order to identify significant variation on similarity caused by the traffic of the m-th port.

This approach for detection of ports under attack via similarity analysis is given by

$$\begin{cases} x_{(m,n)} = x^{(\hat{q})}_{(m,\hat{n})} \\ \\ s_{m,\hat{n}} = \frac{|\boldsymbol{v}^{(q)} \cdot \boldsymbol{v}_{(m,\hat{n})}|}{\|\boldsymbol{v}^{(q)}\| \|\boldsymbol{v}_{(m,\hat{n})}\|}, \end{cases} \tag{14}$$

where $x^{(\hat{q})}_{(m,\hat{n})}$ denotes the m-th port of the selected n-th minute identified as under attack and $x_{(m,n)}$ denotes the m-th port of the last n-th minute of $\boldsymbol{X}_n$, which is used to calculate the $\boldsymbol{v}_{(m,\hat{n})}$ most significant eigenvectors that contains the traffic of the m-th port of the $\hat{n}$-th minute identified as under attack.

Once $\boldsymbol{v}^{(q)}$ and $\boldsymbol{v}_{(m,\hat{n})}$ are obtained, then the $s_{m,\hat{n}}$ similarity degree can be calculated in order to identify if the traffic replacement highlights the adition of anomalous traffic by the evaluated m-th port during the $\hat{n}$-th minute previously identified as under attack.

This procedure should be repeated for each m-th target port of $\hat{\boldsymbol{n}}$, in order to individually identify the network ports under attack during each $\hat{q}$-th time frame.

5. Experiments and Results

This section presents the performed experiments and the acquired results. First, in Section 5.1, the scenario adopted in the experiments is summarized. Then, Section 5.2 shows the results of the largest eigenvalue analysis by time frames. In Section 5.3 are described the results of the evaluated MOS schemes for attack detection. Section 5.4 presents the results of the eigenvalue analysis for identification of time frames under attack, Section 5.5 shows the results of similarity analysis for detailed DoS and port scan identification, and Section 5.6 discuss the computational complexity of the proposed framework.

5.1. Analyzed Scenario

The experiment time is 120 minutes, separated into six time frames, with each time frame corresponding to twenty minutes. Therefore, as the time of each sampling period is one minute, then $N = 20$. For each time frame q, a traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17\times 20}$ was obtained, as well as a covariance $\hat{\boldsymbol{R}}_{yy}^{(q)} \in \mathbb{R}^{17\times 17}$ (calculated via (3)) and a sample covariance matrix $\hat{\boldsymbol{R}}_{zz}^{(q)} \in \mathbb{R}^{17x17}$, assuming that $q = 1, 2, 3, 4, 5$ and 6.

The simulation started at 21:00h, the first time frame was from 21:00h until 21:20h ($q = 1$), the second was from 21:20h until 21:40h ($q = 2$), the third was from 21:40h to 22:00h ($q = 3$), the fourth was from 22:00h until 22:20h ($q = 4$), the fifth was from 22:20h until 22:40h ($q = 5$), and finally, the sixth was from 22:40h until 23.00h ($q = 6$). During the simulation, the victim made legitimate access, and the attacker performed the following attacks: at 21:54h ($q = 3$) was performed a port scan, at the interval ranging from 22:10h to 22:20h ($q = 4$) a synflood attack was simulated, and at the interval from 22:30h to 22:40h ($q = 5$) a fraggle attack was performed.

5.2. Largest Eigenvalues Analysis

For the evaluation of MOS Schemes acuracy for DoS and port scan detection, the framework defines that it is necessary to obtain the largest eigenvalue of each time frame, through eigen decomposition from a covariance of zero

mean variables or covariance matrix of zero mean and unitary standard deviation variables, calculated from the evaluated traffic, as described in Section 4. Through eigenvalue analysis of traffic with DoS or port scan attacks, it is possible to visualize a significant difference between the largest eigenvalues and the remain eigenvalues, which can indicate a relationship between an outlier and time frames under attack.

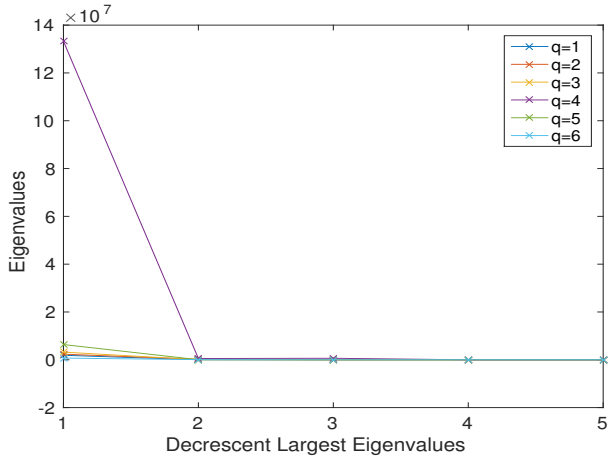

Figure 11: Eigenvalues of the sample covariance matrix (synflood).

Figure 11 depicts the eigenvalues calculated from sample covariance matrix of the network traffic used to evaluate the synflood attack identification. In Figure 11, the largest eigenvalue related to the simulated synflood attack ($q = 4$) stands out significantly from the other eigenvalues.

Figure 12 illustrates the eigenvalues calculated from sample covariance matrix of the matrix used for fraggle attack detection. In Figure 11, the largest eigenvalue related to the simulated synflood attack ($q = 5$) stands out significantly from the other eigenvalues, in accordance with the result shown in Figure 11 for the synflood attack analysis.

Figure 13 depicts the eigenvalues calculated from covariance matrix of zero mean and unitary standard deviation variables, of the network traffic matrix evaluated for port scan detection.

As analyzed for the synflood and fraggle attacks, note that the largest eigen-

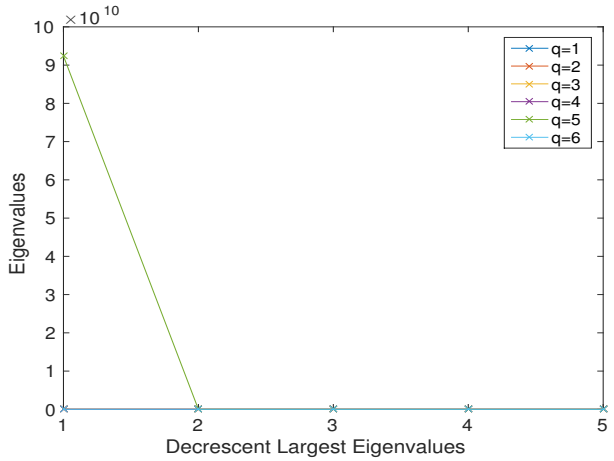

Figure 12: Eigenvalues of the sample covariance matrix (fraggle).

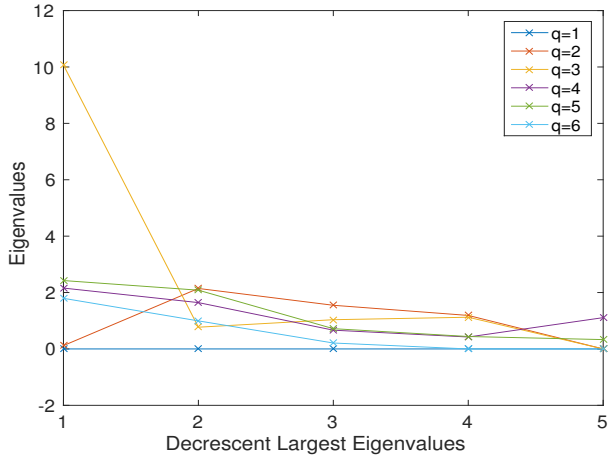

Figure 13: Eigenvalues of the covariance matrix of zero mean and unitary standard deviation (port scan).

485 value, related to this attack ($q = 3$), stands out significantly from the others eigenvalues.

Table 1 presents the values of the largest eigenvalues of each time frame q-th for port scan, synflood and fraggle detection.

In Table 1, note the significant variation of the eigenvalues associated with
490 attacks, in comparison to the others. At $q = 4$, where the synflood attack occurred, the maximum eigenvalue obtained is approximately 21 times larger

Table 1: Largest Eigenvalue related to attacks detection

Time Frame q	**Vectors GETV**			
	Detection of ***synflood/fraggle***	**Detection of** ***synflood***	**Detection of** ***fraggle***	**Detection of** ***port scan***
1	1887545	1887545	1887545	2,0734
2	2341327	2341327	2341327	2,1451
3	3213867	3213867	3213867	10,0718
4	133238294	133238294	731229	2,1620
5	92384021611	6367983	92384021611	2,4253
6	708335	708335	708335	1,7948

than the second one. At $q = 5$, where the fraggle attack occurred, the maximum eigenvalue obtained is about 29,000 times larger than the second one. At $q = 3$, where the port scan attack occurred, the maximum eigenvalue obtained is approximately 4 times larger than the second one. In the last case, for port scan attack detection, although the largest eigenvalue presented no too large variance to the second one, if compared to synflood or fraggle attacks, it clearly deviates from the remaining largest eigenvalues.

These results highlight that all q-th time frames where a network attack was simulated, present high significant variance between the largest eigenvalue and the remaining eigenvalues, obtained from sample covariance matrix, for DoS detection, or from covariance matrix of zero mean and unitary standard deviation variables, for port scan detection. Therefore, we propose to apply the vector of the largest eigenvalues to MOS schemes in order to evaluate their accuracy for identification of time frames under attack, motivated by the fact that it is relevant to apply MOS schemes to automate the attack detection process, taking into account the characteristics of the evaluated eigenvalues.

5.3. MOS Schemes Evaluation

In [7], we evaluate the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE MOS schemes [16, 7] for synflood and port scan attack detection. In this work we extend that evaluation for fraggle attack detection, applying the same schemes to fraggle attack detection over the traffic presented in Section 3, as results shown in Table 2.

Table 2: MOS schemes applied to port scan and DoS detection

Type of analysis q	**MOS schemes (estimated model order** $\hat{d}$**)**						**(d)**
	AIC	**MDL**	**EDC**	**RADOI**	**EFT**	**SURE**	
Detection of synflood (presence of attack)	2	1	**1**	5	**1**	4	**1**
Detection of synflood (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of fraggle (presence of attack)	1	1	**1**	5	**1**	4	**1**
Detection of fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of port scan (presence of attack)	1	1	**1**	1	**1**	9	**1**
Detection of port scan (absence of attack)	0	0	**0**	1	**0**	1	**0**
Detection of synflood/fraggle (presence of attack)	2	2	**2**	5	**2**	5	**2**
Detection of synflood/fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**

Note that $\hat{d} = 1$, if there is attack, while $\hat{d} > 1$ indicates more than one attack. An example of this could be seen for attack detection via EFT for traffic containing synflood and fraggle attacks, showing $\hat{d} = 2$, which indicates the presence of two attacks, as expected by the d real values of Table 2.

In Table 2, two MOS schemes outperforms from the others, EDC and EFT. Efficient Detection Criterion (EDC) and Exponential Fitting Test (EFT) are the most effective schemes, correctly estimating the number of attacks in comparison to the expected values for effetctive attack detection, as defined by the column of real values in Table 2. The AIC and MDL schemes are satisfactory only for port scan detection, however SURE and RADOI schemes did not show effective results for port scan or DoS detection.

Although EDC and EFT presented the same accuracy on the evaluation, the EDC scheme requires less processing time than EFT, which is an important criteria to select EDC as the MOS scheme for DoS and port scan detection on the remain experiments.

According to Table 2, EDC and EFT estimated correctly the number of attacks of a time frame vector, indicating that ocurred $\hat{d}$ network attacks, but not providing additional details, what highlights the necessity of complementary approaches in order to estimate the time and ports under attack. Hence, we

propose apply eigen analysis to estimate the q-th time frames under attack and eigen similarity analysis to estimate the minutes and ports under attack.

5.4. Eigenvalue Analysis

According to the results presented in Section 5.2, the largest eigenvalue stands out significantly from the others eigenvalues of an evaluated q-th time frame. This behavior can also be observed in the largest eigenvalues analysis, according to results presented in Table 1, where it is possible to observe that the $\hat{d}$ largest eigen values of the time frames under attacks stand out significantly from the others largest eigenvalues.

Therefore, we conclude that the $\hat{d}$ largest eigenvalues correspond to the respectives q-th time frames under attack, which is denoted by $\hat{\boldsymbol{q}}_{\max}$ and can be calculated according to Algorithm 1.

5.5. Eigen Similarity Analysis

This paper proposes applying eigen similarity analysis to detect time and ports under attack, from each q-th time frames under attack defined by $\hat{\boldsymbol{q}}_{\max}$. Hence, the proposed framework was applied to the time frames where $q = 3$, $q = 4$ and $q = 5$ to respectively evaluate its effectiveness for port scan, synflood and fraggle attack detection.

5.5.1. Time Analysis

Three approaches were evaluated for eigen similarity analysis: incremental, individual and incremental individualized approaches. For the incremental individualized approach, each minute was incrementally appended into the selected $\boldsymbol{X}^{(q)}$ for obtaining $\boldsymbol{v}_{(n)}$ to similarity analysis of the n-th minute, until detect the first n-th minute under attack. Subsequently, $\boldsymbol{X}_n$ became the new reference of traffic without network attack and each subsequent minute must have its similarity individually evaluated. For the incremental approach, each n-th minute must be incrementally appended into $\boldsymbol{X}^{(q)}$, for obtaining the next eigenvectors $\boldsymbol{v}_{(n)}$ for individual time similarity analyis. For the individual approach, each n-th minute must be individually appended into $\boldsymbol{X}^{(q)}$, without

incremental append, but doing individual appended into $\boldsymbol{X}^{(q)}$ for obtaining the next eigenvectors $\boldsymbol{v}_{(n)}$ for individual similarity analyis.

Table 3 presents the results of the evaluation of three approaches for similarity analysis of eigenvectors for port scan detection.

Table 3: Eigen Similarity Analysis for Port Scan Detection

Time Frame q	**Time** n	**Similarity Analysis**			**Attack?**
		Incremental Individualized	**Incremental**	**Individual**	
3	1	0.9946	0.9946	0.9946	no
3	2	0.9934	0.9934	0.9999	no
3	3	0.9912	0.9912	0.9999	no
3	4	0.9888	0.9888	0.9999	no
3	5	0.9856	0.9856	0.9998	no
3	6	0.9840	0.9840	0.9999	no
3	7	0.9824	0.9824	1.0000	no
3	8	0.9794	0.9794	0.9999	no
3	9	0.9673	0.9673	0.9926	no
3	10	0.9674	0.9674	0.9997	no
3	11	0.9733	0.9733	0.9993	no
3	12	0.9702	0.9702	0.9993	no
3	13	0.9677	0.9677	0.9999	no
3	14	0.9646	0.9646	0.9998	no
3	15	0.0216	0.0216	0.0276	yes
3	16	0.9621	0.0209	1.0000	no
3	17	0.9611	0.0199	0.9998	no
3	18	0.9612	0.0191	0.9999	no
3	19	0.9613	0.0186	0.9998	no
3	20	0.9638	0.0190	1.0000	no

Table 3 shows the evaluation of the time frame $q = 3$, when the port scan attack was simulated, considering the incremental individualized, incremental and individual approaches for eigen similarity analysis. According to the presented results, it is possible to observe the high similarity between network traffic without attack, which was larger than 0.9610 for all evaluated cases, and emphasize the expressive low similarity when evaluated the traffic with the simulated port scan attack ($n = 15$), which was lower than 0.0276 for all evaluated approaches.

Comparing the approaches for similarity analysis, it is possible to observe that all approaches highlight the low similarity when evaluated the traffic under attack. However, the incremental approach figured out low similarity for times without attack, where $n = 16, 17, 18, 19, 20$, what indicates that the incremental approach can produce false posive results. This behavior occurs because the

incremental approaches appends all selected traffic into the reference traffic for comparison against the original reference traffic, what makes more evident the first lack of similarity but reduces the changing detection capability after an attack detection.

Table 4 presents the results of the evaluation of the similarity analysis of eigenvectors for synflood detection. It shows the evaluation of the time frame $q = 4$, when the synflood attack is simulated, considering the incremental individualized, incremental and individual approaches for eigen similarity analysis. According to the results, it is possible to observe the high similarity between network traffic without attack, which is larger than 0.9907 for all evaluated cases, and emphasize the expressive low similarity when evaluated the traffic with synflood attack (between $n = 11$ and $n = 20$), which is lower than 0.1244 for all evaluated approaches.

Table 4: Eigen Similarity Analysis for Synflood Detection

Time Frame q	**Time** n	**Similarity Analysis**			**Attack?**
		Incremental Individualized	**Incremental**	**Individual**	
4	1	1.0000	1.0000	1.0000	no
4	2	0.9999	0.9999	1.0000	no
4	3	0.9997	0.9997	0.9999	no
4	4	0.9998	0.9998	1.0000	no
4	5	0.9965	0.9965	0.9908	no
4	6	0.9975	0.9975	1.0000	no
4	7	0.9977	0.9977	1.0000	no
4	8	0.9980	0.9980	1.0000	no
4	9	0.9987	0.9987	0.9999	no
4	10	0.9991	0.9991	1.0000	no
4	11	0.0085	0.0085	0.0284	yes
4	12	0.0162	0.0120	0.0343	yes
4	13	0.0248	0.0158	0.0427	yes
4	14	0.1243	0.0185	0.1041	yes
4	15	0.0082	0.0162	0.0103	yes
4	16	0.0404	0.0070	0.0580	yes
4	17	0.0397	0.0007	0.0573	yes
4	18	0.0408	0.0042	0.0584	yes
4	19	0.0408	0.0079	0.0584	yes
4	20	0.0477	0.0092	0.0757	yes

The incremental approach produces better results if compared with other evaluated approaches, with lower values and maximum of 0.0185 for times under attack, but this approach presents change detection limitation after the first

outlier of similarity, as shown in Table 3 for port scan detection.

Comparing the incremental individualized and the individual approaches for eigen similarity analysis, it is possible to observe that the incremental individualized approach obtain lowest values for almost all cases, except for the time $n = 14$, where incremental individualized approach identified a larger similarity than the individual approach. The incremental individualized appends information about each evaluated traffic, therefore it incorporates traffic behaviors that can reduce the outlier capability detection, as ocurred for the time $n = 14$.

Table 5 presents the results of the eigen similarity analysis evaluation for fraggle detection.

Table 5: Eigen Similarity Analysis for Fraggle Detection

Time Frame q	**Time** n	**Similarity Analysis**			**Attack?**
		Incremental Individualized	**Incremental**	**Individual**	
5	1	1.0000	1.0000	1.0000	no
5	2	0.9999	0.9999	1.0000	no
5	3	1.0000	1.0000	1.0000	no
5	4	0.9999	0.9999	1.0000	no
5	5	0.9993	0.9993	0.9997	no
5	6	0.9993	0.9993	0.9997	no
5	7	0.9994	0.9994	1.0000	no
5	8	0.9995	0.9995	1.0000	no
5	9	0.9995	0.9995	1.0000	no
5	10	0.9995	0.9995	1.0000	no
5	11	0.0031	0.0031	0.0021	yes
5	12	0.0019	0.0025	0.0009	yes
5	13	0.0030	0.0026	0.0020	yes
5	14	0.0030	0.0027	0.0020	yes
5	15	0.0030	0.0028	0.0020	yes
5	16	0.0012	0.0025	0.0002	yes
5	17	0.0030	0.0026	0.0020	yes
5	18	0.0030	0.0026	0.0020	yes
5	19	0.0030	0.0027	0.0020	yes
5	20	0.0069	0.0023	0.0083	yes

For fraggle attack detection, the lack of similarity between legitimate and malicious traffic was more evident than for the evaluation of synflood and port scan detecion. This behavior can be explained by the number of packets generated through the fraggle attack simulation, that was significative larger than the number of packets generated during the synflood simulation. Considering the three approaches, the largest value for times under attack was 0.0083, while

the shortest value for times without attacks was 0.9993.

Therefore, considering the evaluation for port scan, synflood and fraggle detection, the incremental approach can produce false positive results, while the individual and incremental individualized approaches produce quite similar results, even though the individual approach be more simple and require less memory and processing time.

These results highlight the capabilitiy of change detection based on similarity between legitimate and malicious traffic from DoS or port scan attacks, endorsing the effectiveness and safety for adoption of threshold for attack detection through eigen similarity analysis.

5.5.2. Port Analysis

Given $\hat{N}$, which is the set of estimated n-th minutes under attack, it is possible to apply cosine similarity analysis to identify variation of the most significant eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port, during a n-th minute. Therefore, the incremental individualized and individual approaches of eigen similarity analysis were evaluated, for detection of ports under DoS and port scan attacks, according to results presented in following tables. For this evaluation, the v last most significant eigenvectors without attack was used as reference for similarity analysis against each target port m-th.

Table 6 presents the results of the evaluation of eigen similarity analysis for detection of ports under port scan attack, showing only the time frame $q = 3$ and minute $n = 15$, due to the simulated port scan attack occurred only at this time, although the remain time frame has been completely evaluated and presented high similarity to the reference of traffic without network attack.

The incremental individualized approach presented more sensibility to anomaly detection than the individual approach, the former produced the identification of a low similarity of 0.0298 for almost all ports under attack, unless the port 21, although the simulation has attacked this port. The individual approach was not able to identify low similarity for ports under attack, resunting in values of

Table 6: Eigen Similarity Analysis for Detection of Ports Under Port Scan Attack (q=3 and n=15)

Port p	**Approaches**		**Attack?**
	Incremental Individualized	**Individual**	
80	0.9999	0.9999	no
443	0.9999	0.9999	no
53	0.9999	0.9999	no
21	0.9999	0.9997	yes
22	0.0298	0.9997	yes
23	0.0298	0.9997	yes
25	0.0298	0.9997	yes
110	0.0298	0.9997	yes
143	0.0298	0.9997	yes
161	0.0298	0.9997	yes
69	0.0298	0.9997	yes
123	0.0298	0.9997	yes
445	0.0298	0.9997	yes
600	0.9999	0.9999	no
19	0.9999	0.9999	no
67	0.9999	0.9999	no
68	0.9999	0.9999	no

0.9997 for ports with anomalous traffic and 0.9999 for ports without network attack.

For the evaluation of the proposed approaches for identification of ports under synflood and fraggle attack, all minutes of each time frame, in which one attack location was estimated, were analyzed. Even though, due to space limitations, only the results of the firt minute where a low similarity was idenfified will be shown such as where $n = 11$. Nevertheless, the results obtained for the evaluation of traffic without attack presented high similarity to the reference traffic, with similarities close to 0.9999, and the evaluation of the other minutes under attack presented results quite similar to the resuts shown in the Tables 7 and 8.

Table 7 presents the results of the evaluation of eigen similarity analysis for detection of ports under synflood attack, showing only the time frame $q = 4$ and minute $n = 11$.

According to results presented in Table 7, both approches identifies low similarity for the traffic of port 600, which is the target port of the simulated synflood attack, but the incremental individualized approach identifies the low-

Table 7: Eigen Similarity Analysis for Detection of Ports Under Synflood Attack (q=4 and n=11)

Port p	**Approaches**		**Attack?**
	Incremental Individualized	**Individual**	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	nos
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	0.0077	0.0427	yes
19	1.0000	1.0000	no
67	1.0000	1.0000	no
68	1.0000	1.0000	no

est similarity and presents better sensibility to identification of synflood attack through eigen similarity analysis assisted by threshold definition.

Table 8 presents the results of the evaluation of eigen similarity analysis for detection of ports under fraggle attack, showing only the time frame $q = 5$ and minute $n = 11$.

The results for the avaluation of ports under fraggle attack, shown in Table 8, were similar to the results obtained for synflood analysis, with the identification of low similarity for traffic of the port under attack. Nevertheless, for fraggle analysis, the individual approach identified the lowest similarity, that is 0.0004 while the incremental individualized approach obtained a similarity of 0.0031.

The incremental individualized approach was able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches presented false positives or low sensibility to eigen similarity analyis for network attack detection. This approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

Table 8: Eigen Similarity Analysis for Detection of Ports Under Fraggle Attack (q=5 and t=11)

Port p	**Approaches**		**Attack?**
	Incremental Individualized	**Individual**	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	no
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	1.0000	1.0000	no
19	0.0031	0.0004	yes
67	1.0000	1.0000	no
68	1.0000	1.0000	no

According to the shown significant lack of similarity between legitimate and malicious traffic, it is possible to adopt safe thresholds for DoS and port scan detection through eigen similarity analysis.

5.6. Complexity Analysis

This subsection discusses the computational complexity of the proposed framework, focusing on the main steps, which are the eigenvalues decomposition (EVD), largest eigenvalues analysis, application of MOS scheme and eigen similarity analysis, according to Figure 7 and equations presented in Section 4.

The EVD, calculated according to (6), requires the previous calculation of covariance matrix, according to Equations 2, 3, 4 and 5. The covariance matrix calculation is $O(M^2N)$ and the EVD is $O(N^3)$, where M denotes the number of network ports and N denotes the period time. Therefore, the computational complexity for all steps for EVD can be represented as $O(M^2N+N^3)$ and yields an $O(N^3)$ upper bound on the worst-case running time for EVD.

EDC and EFT are the MOS schemes that presented accuracy on the evaluation for the network attack detection. The computational complexity evaluation for MOS focuses on EDC scheme, since EDC requires less processing time than

EFT but presents the same accuracy for the evaluated scenario. EDC scheme is $O(QlogQ + Q + QlogQ)$ and its worst-case running time can be represented as $O(QlogQ)$, where Q denotes the number of time frames.

The largest eigenvalue analysis is $O(\hat{d}Q)$, where $\hat{d}$ denotes the number of time frame under attack, according to Algorithm 1. Subsequently, the eigen similarity analysis relies on EVD and cosine similarity analysis, which is $O(N^2)$, for $\hat{d}$ time frames, therefore the eigen similarity analysis is which is $O(\hat{d}(M^2N + N^3 + N^2))$ and yields an $O(N^3)$ upper bound on the worst-case running time for eigen similarity analysis.

Therefore, the proposed framework is $O(N^3 + QlogQ + \hat{d}Q + N^3)$ and its worst-case running time is $O(N^3)$. The computational complexity of EVD is predominant in the framework, but the approach splits the data into time frames with period time N, which makes possible to limitate the growth of N even for evaluations of cases with total time larger than N, reducing the impact caused by the computational complexity of EVD.

6. Conclusion and Future Works

This paper models the network traffic as a signal processing formulation for applying to the framework for detection and identification of network attacks, which is based on eigenvalue analysis, model order selection (MOS) and eigen similarity analysis.

The proposed framework is evaluated and the experimental results show that synflood, fraggle and port scan attacks can be detected accurately and with great detail in an automatic and blind fashion, applying signal processing concepts for traffic modeling and through approaches based on MOS and eigen similarity analysis. The main contributions of this work were: the extension of an approach based on MOS combined with eigen analysis to blindly detect time frames under network attack; the proposal and evaluation of an eigen similarity based framework to identify details of network attacks, presenting accuracy of timely detection and identification of TCP/UDP ports under attack.

This paper evaluated the effectiveness of MOS schemes for fraggle attack detection, extending our previous work [7] and showing that the analysis of the largest eigenvalues by time frames can be applied to detect the number of port scanning, and DoS attacks, but still requiring more information for detailed attack detection. Therefore, we proposed a novel approach for detailed network attack detection, based on eigen similarity analysis.

The incremental individualized approach of eigen similarity analysis, is able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches present false positives or low sensibility to eigen similarity analyis for network attack detection. Therefore, the incremental individualized approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

According to the significant similarity difference between legitimate and malicious traffic, it is possible to adopt safe thresholds for DoS and port scan detection through eigen similarity analysis.

Future research is directed to the fully automated blind attack detection, extending the evaluation to performance and accuracy analyis of the approach to well-known datasets of network attack detection, and analysing the performance according to parameter configuration in order to better understand the impact of N, M and Q and identify their best configuration for obtaining gains into processing time and acuracy.

Appendix A. Model Order Selection (MOS)

The model order selection is a key point in many digital signal processing applications, including radar, sonar, communications, channel modeling, medical imaging, among others. MOS allows analysis of reduced data set, through separating noise components of the main components, for example. Moreover, the model order is crucial for many parameter estimation techniques [16], since the amount of parameters to be estimated depends on the model order.

The model selection procedure chooses the "best" model of a finite set of models, according to some criterias [17]. Therefore, given some data set, it is chosen a model which was evaluated as the best model to describe the specified data set.

The state of the art regarding estimation techniques of model order based on eigenvalues includes: Akaike's Information Theoretic Criterion - AIC [18, 19]; Minimum Description Length - MDL [20, 19]; Efficient Detection Criterion - EDC [21]; Stein's Unbiased Risk Estimator - SURE [22]; RADOI [23] and Exponential Fitting Test - EFT [24, 25, 5].

In AIC, MDL and EDC techniques, the information criterion is a function of the geometric mean *g(k)* and the arithmetic mean *a(k)* relating to smaller k eigenvalues, where k is a candidate value for the model order d [16].

Basically, the difference between the AIC, MDL and EDC schemes is the penalty function *p(k, N, α)*, so these techniques can be written in general as [16]:

$$\hat{d} = \arg\min_{k} \; J(k), \tag{A.1}$$

where

$$J(k) = -N(\alpha - k) \log \left(g(k)/a(k)\right) + p(k, N, \alpha), \tag{A.2}$$

where $\hat{d}$ is an estimate d of the model order, N is the number of samples, $\alpha = M$ and means the number of variables of the problem, and $0 \leq k \leq \min[M, N]$. Penalty functions for AIC, MDL and EDC are given by the Table A.9.

Table A.9: Penalty functions for the schemes AIC, MDL and EDC

Scheme	**Penalty function** $p(k, N, \alpha)$
AIC	$k(2\alpha - k)$
MDL	$0.5k(2\alpha - k)\log(N)$
EDC	$0.5k(2\alpha - k)\sqrt{N\,\ln(\ln N)}$

The Exponential Fitting Test (EFT) can effectively be used in cases where the number of samples N is small. This technique is based on observations of data contaminated only with white noise, where the profile of eigenvalues can be approximated by an exponential decaying [24].

Given λ_i be the i-th eigenvalue, the exponential model can be expressed by:

$$E\{\lambda_i\} = E\{\lambda_1\} \cdot q(\alpha, \beta)^{i-1}, \tag{A.3}$$

where $E\{\cdot\}$ is the expectation operator, and it is considered that the eigenvalues are ordered in the that λ_1 represents the largest eigenvalue. The term *q(α, β)* is defined as:

$$q(\alpha, \beta) = \exp\left\{-\sqrt{\frac{30}{\alpha^2+2} - \sqrt{\frac{900}{(\alpha^2+2)^2} - \frac{720\alpha}{\beta(\alpha^4+\alpha^2-2)}}}\right\}, \tag{A.4}$$

where $0 < q(\alpha, \beta) < 1$. According to [25], if $M \leq N$, then $\beta = N$.

References

[1] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: ACM SIGCOMM Computer Communication Review, Vol. 35, ACM, 2005, pp. 217–228.

[2] W. Lu, A. A. Ghorbani, Network anomaly detection based on wavelet analysis, EURASIP J. Adv. Signal Process 2009 (2009) 4:1–4:16. doi:10.1155/2009/837601.
URL http://dx.doi.org/10.1155/2009/837601

[3] C.-T. Huang, R. K. C. Chang, P. Huang, Editorial: Signal processing applications in network intrusion detection systems, EURASIP J. Adv. Signal Process 2009 (2009) 9:1–9:2. doi:10.1155/2009/527689.
URL http://dx.doi.org/10.1155/2009/527689

[4] L. Zonglin, H. Guangmin, Y. Xingmiao, Y. Dan, Detecting distributed network traffic anomaly with network-wide correlation analysis, EURASIP J. Adv. Signal Process 2009 (2009) 2:1–2:11. doi:10.1155/2009/752818. URL http://dx.doi.org/10.1155/2009/752818

[5] B. M. David, J. P. C. L. da Costa, A. C. Nascimento, D. Amaral, M. Holtz, R. T. de Sousa Jr, Blind automatic malicious activity detection in honeypot data, in: The International Conference on Forensic Computer Science (ICoFCS), 2011.

[6] J. P. C. L. da Costa, E. P. de Freitas, B. M. David, A. M. R. Serrano, D. Amaral, R. T. de Sousa Jr, Improved blind automatic malicious activity detection in honeypot data, in: The International Conference on Forensic Computer Science (ICoFCS), 2012.

[7] D. F. Tenório, J. P. C. L. da Costa, R. T. de Sousa Jr, Greatest eigenvalue time vector approach for blind detection of malicious traffic, in: The International Conference on Forensic Computer Science (ICoFCS), 2013.

[8] D. Mudzingwa, R. Agrawal, A study of methodologies used in intrusion detection and prevention systems (idps), in: Southeastcon, 2012 Proceedings of IEEE, IEEE, 2012, pp. 1–6.

[9] W. He, G. Hu, X. Yao, G. Kan, H. Wang, H. Xiang, Applying multiple time series data mining to large-scale network traffic analysis, in: 2008 IEEE Conference on Cybernetics and Intelligent Systems, 2008, pp. 394–399.

[10] A. Ghourabi, T. Abbes, A. Bouhoula, Data analyzer based on data mining for honeypot router, in: Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on, IEEE, 2010, pp. 1–6.

[11] F. Raynal, Y. Berthier, P. Biondi, D. Kaminsky, Honeypot forensics, in: Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, IEEE, 2004, pp. 22–29.

[12] S. Almotairi, A. Clark, G. Mohay, J. Zimmermann, A technique for detecting new attacks in low-interaction honeypot traffic, in: Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on, IEEE, 2009, pp. 7–13.

[13] W. Z. A. Zakaria, M. L. M. Kiah, A review on artificial intelligence techniques for developing intelligent honeypot, in: Proceeding of: 8th International Conference on Computing Technology and Information Management, At Seoul, Korea, 2012.

[14] Y.-J. Lee, Y.-R. Yeh, Y.-C. F. Wang, Anomaly detection via online oversampling principal component analysis, Knowledge and Data Engineering, IEEE Transactions on 25 (7) (2013) 1460–1470. doi:10.1109/TKDE.2012.99.

[15] S. Jin, D. S. Yeung, A covariance analysis model for ddos attack detection, in: Communications, 2004 IEEE International Conference on, Vol. 4, IEEE, 2004, pp. 1882–1886.

[16] J. P. C. L. da Costa, A. Thakre, F. Roemer, M. Haardt, Comparison of model order selection techniques for high-resolution parameter estimation algorithms, in: Proc. 54th International Scientific Colloquium (IWK'09), Ilmenau, Germany, 2009.

[17] J. Rajan, P. Rayner, Model order selection for the singular value decomposition and the discrete karhunen–loeve transform using a bayesian approach, IEE Proceedings-Vision, Image and Signal Processing 144 (2) (1997) 116–123.

[18] H. Akaike, A new look at the statistical model identification, Automatic Control, IEEE Transactions on 19 (6) (1974) 716–723.

[19] M. Wax, T. Kailath, Detection of signals by information theoretic criteria, Acoustics, Speech and Signal Processing, IEEE Transactions on 33 (2) (1985) 387–392.

[20] A. Barron, J. Rissanen, B. Yu, The minimum description length principle in coding and modeling, Information Theory, IEEE Transactions on 44 (6) (1998) 2743–2760.

[21] L. Zhao, P. Krishnaiah, Z. Bai, On detection of the number of signals in presence of white noise, Journal of Multivariate Analysis 20 (1) (1986) 1–25.

[22] M. O. Ulfarsson, V. Solo, Rank selection in noist pca with sure and random matrix theory, in: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, IEEE, 2008, pp. 3317–3320.

[23] E. Radoi, A. Quinquis, A new method for estimating the number of harmonic components in noise with application in high resolution radar, EURASIP Journal on Applied Signal Processing 2004 (2004) 1177–1188.

[24] J. Grouffaud, P. Larzabal, H. Clergeot, Some properties of ordered eigenvalues of a wishart matrix: application in detection test and model order selection, in: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on, Vol. 5, IEEE, 1996, pp. 2463–2466.

[25] A. Quinlan, J.-P. Barbot, P. Larzabal, M. Haardt, Model order selection for short data: An exponential fitting test (eft), EURASIP Journal on Advances in Signal Processing 2007.