PCA-based Multivariate Statistical Network Monitoring for Anomaly Detection

José Camacho, Alejandro Pérez-Villegas, Pedro García-Teodoro, Gabriel Maciá-Fernández

Department of Signal Theory, Telematics and Communications
School of Computer Science and Telecommunications - CITIC
University of Granada (Spain)

Abstract

The multivariate approach based on Principal Component Analysis (PCA) for anomaly detection received a lot of attention from the networking community one decade ago mainly thanks to the work of Lakhina and co-workers. However, this work was criticized by several authors that claimed a number of limitations of the approach. Neither the original proposal nor the critic publications were completely aware of the established methodology for PCA anomaly detection, which by that time had been developed for more than three decades in the area of industrial monitoring and chemometrics as part of the Multivariate Statistical Process Control (MSPC) theory. In this paper, the main steps of the MSPC approach based on PCA are introduced; related networking literature is reviewed, highlighting some differences with MSPC and drawbacks in their approaches; and specificities and challenges in the application of MSPC to networking are analyzed. All of this is demonstrated through illustrative experimentation that supports our discussion and reasoning.

Keywords:
Multivariate Statistical Process Control, Network Monitoring, Network Security, Principal Component Analysis, Anomaly Detection

1. Introduction

The outstanding capability of multivariate analysis to detect anomalies has been recognized in several domains, including industrial monitoring [1, 2, 3, 4] and networking [5, 6, 7, 8, 9]. The use of multivariate analysis for anomaly detection is typically referred to as Multivariate Statistical Process Control (MSPC) [10]. A main tool in MSPC is Principal Component Analysis (PCA).

The pioneering work by Lakhina et *al.* [5] introduced the use of PCA for network anomaly detection. Their approach received a lot of attention from the networking community one decade ago, and thus a variety of other proposals has been developed based on it. However, the approach was also criticized by a number of papers. Ringberg et *al.* [9] claimed that it is sensitive to calibration settings. In particular, that:

(i) The false positive rate is very sensitive to small differences in the number of principal components in the normal subspace.
(ii) The effectiveness of PCA is sensitive to the level of aggregation of the traffic measurements.
(iii) A large anomaly may inadvertently pollute the normal subspace, and go undetected.
(iv) Correct diagnosis is an inherently challenging problem.

Here we argue that these supposed problems are the result of flaws in adopting PCA to the anomaly detection field. It should be noted that such flaws are found not only in the original approach but also in its detractors. Although Lakhina et *al.* noted that similar approaches to theirs were already developed in the chemical engineering area, the bulk of the (by that time) well-established theory of MSPC based on PCA was ignored in their papers.

In this work, the theory of PCA-based MSPC is reviewed, differences with Lakhina et *al.* and posterior approaches being highlighted where appropriate and illustrated with examples. We refer to as Multivariate Statistical Network Monitoring (MSNM) the approach that follows the MSPC theory for anomaly detection in communication networks. The last term in MSNM, "monitoring", has been preferred to control, which is seldom used in the networking community. Furthermore, the term "control" has a different meaning in fields other than statistics, such as automatic feedback control [11, 12].

The rest of the paper is organized as follows. Section 2 reviews the principal works on PCA-based network anomaly detection. Section 3 presents fundamentals on statistical process control, in particular on the use of PCA-based MSPC. After that, Section 4 discusses the necessary pre-processing for networking data to be analyzed with PCA, while the proper processing for dynamic modeling is subsequently described in Section 5. The discussion and argumentation carried out until this point are demonstrated by means of some illustrative examples in Section 6. Finally, Section 7 summarizes the main contributions of the work and future challenges.

*Corresponding author: J. Camacho (email: josecamacho@ugr.es)

2. Related Work

Supervising computer and network systems is a key topic in the literature from several decades ago. The main purpose of existent solutions in the filed is the early detection of potential failures and malfunctions. From this, some recovery actions could be taken in order to restore the normal desired operation for the monitored environment.

The terms "failure" and "malfunction" must be interpreted as a global concept that can be caused by a number of different reasons, either accidental or not. One of the most studied causes is the one deliberately carried out by malicious users through attacks aimed at exploiting some system vulnerability. Whichever the origin of the failure or malfunction, the goal of the monitoring and detection systems is similar: to prevent the environment from decreasing its performance or even from crashing. For that, the usual procedure when determining the occurrence of some kind of "anomaly" (*i.e.*, a certain deviation from the normal expected behavior of the monitored environment) is to trigger an alarm as a previous step to solve the problem.

Among several other existent anomaly detection paradigms [13, 14], statistical solutions have been widely adopted [15]. In particular, multivariate approaches such as PCA were adopted several years ago [16, 17, 18, 19], their unsupervised nature being the main benefit argued in comparison with other solutions. As previously stated, maybe the most referred work is that of Lakina et *al.* [5]. Based on it, several further proposals have been developed in the literature.

Authors in [20] introduce a network anomaly detection for large distributed systems. It is based on a stochastic matrix perturbation analysis that characterizes the trade-off between the accuracy of anomaly detection and the amount of data communicated over the network. On the other hand, [21] discusses the problem of contaminated training data and propose to use PCA on the basis of robust estimators to overcome the necessity of a supervised preprocessing step for anomaly detection in the context of intrusion detection systems. Also [22] and [23] highlight the advantage of PCA in avoiding the need of labeled training datasets in comparison with other detection schemes.

Kim et *al.* [24] present a higher-order singular value decomposition (HOSVD) and higher-order orthogonal iteration (HOOI) algorithms on network traffic anomaly detection by rearranging the data in tensor formats. Simulation results show that the higher-order methods improve the detection performance while also reduce the complexity for large-scale networks. The work in [25] tries to solve scalability problems of PCA. For that, a sketch-based streaming PCA algorithm for the network-wide traffic anomaly detection in a distributed fashion is proposed.

Authors in [26] introduce a PCA-based methodology to detect anomalies related to potential losses of data in WSNs. Based on this, a subsequent data recovery procedure is also contributed. This relies on the exploitation of the spatial correlation inherent in WSNs. Different routing strategies to collect all the information around the network are analyzed to evaluate the suitability of the approach.

Reference [27] uses distributed principal component analysis (DPCA) and fixed-width clustering (FWC) in order to establish a global normal profile and to detect anomalies. The process of establishing the global normal profile is distributed among all sensor nodes. Authors also use weighted coefficients and a forgetting curve to periodically update the established normal profile. A similar work in obtaining user profiles in communication environments is that in [28].

In [29], the "classical" PCA approach is complemented with the Kullback-Leibler divergence to improve detection results. Similarly, [30] combines PCA with distance-based anomaly detection (DB-AD) to reduce dimensionality. Authors in [31] combine PCA with wavelet algorithms for network traffic anomaly detection. References [32] and [33] study PCA variants to solve the calibration sensitivity. Like the latter, [34] uses a entropy-based PCA.

As previously indicated, almost all of the existent works on PCA-based anomaly detection in networking are developed taking as a base the work by Lakhina [5]. This way, all of them present similar disadvantages. The main points in which the approach of Lakhina et *al.* [5] does not properly follow the MSPC theory are:

- Lakhina et *al.* use PCA to divide data in two subspaces for normal and anomalous behavior. Anomaly detection is performed only in the latter. In MSPC, PCA is used to split data in a structured subspace and a noisy subspace. Detection is performed in both subspaces using different statistics [11, 35, 36, 37].

- Lakhina et *al.* use data for the calibration of the anomaly detector that may incorporate anomalies. In MSPC, a two phases approach is performed so as to avoid this problem [2, 38].

- Lakhina et *al.* select the number of Principal Components to capture a specific amount of variance. Subsequently, in [39] they suggest another common approach which is equivalent to the so-called Scree plot, based on finding a knee in a plot where the amount of variance captured by the PCA model is represented in terms of the number of PCs retained. These approaches are well known to be impractical in most MSPC set-ups [12, 40].

- Lakhina et *al.* use a supervised fault diagnosis system based on the approach of [41], for which an a priori set of common faults is necessary. Subsequently, they extended in [39] their approach for unsupervised diagnosis, yet assuming a predefined structure in the fault. In MSPC, contribution plots are commonly used for faults diagnosis without the need of any set of previously defined common faults [11, 42, 43].

In what follows, we will present the fundamentals of MSPC and will demonstrate through experimentation that some of the limitations highlighted in current works on PCA-based anomaly detection are due to the incorrect use of the MSPC theory.

3. Statistical Process Control

Statistical Process Control (SPC) is a methodology pioneered by Walter Andrew Shewhart and supported by Williams Edwards Deming in the past century, with a tremendous impact in the U.S. and Japan manufacturing industry [44]. A main goal of SPC is to distinguish common causes from special causes of variation in a system. Common causes of variation reflect natural variability within a system, while special or assignable causes of variation reflect anomalous events. A system is said to be under statistical control when it is only affected by common causes of variation [11].

The SPC theory establishes two steps or phases that need to be fulfilled to set up an anomaly detector [10]. The first step (*phase I*) is devoted to detect all special causes of variability in the system and correct them. This is an iterative process in which the analyst staff detects each issue, diagnoses the probable root causes and reports them to whomever is responsible for solving them. The second step (*phase II*) is performed on a system that is under normal operation conditions (NOC) or statistical control. This essentially means that the system should be free of all the anomalies detected in phase I, and that all events that happen on a normal basis are due to the expected functioning of the system. The main idea beneath the definition of these two phases is that an anomaly detector should be developed only for a system under statistical control.

Traditional SPC is based on univariate statistics and univariate control charts, in which just one variable is monitored and/or visualized at a time (see Figure 1 as an example). This commonly happens in networking, where network analysis tools are typically limited to univariate time series signals [45]. As the number of variables to be monitored increases, the approach to visualize one variable at a time reduces its performance and it is more convenient to use multivariate statistics and charts.

Multivariate SPC (MSPC) is an extension of SPC to control several variables at the same time. Traditional MSPC, however, do not take into account correlation among variables, which may lead to accuracy and computational problems due to ill-conditioning [46]. When the number of variables is very large and/or the variables are highly inter-related, the use of latent variable methods such as PCA within the MSPC monitor is recommended.

3.1. Principal Component Analysis

PCA is applied to two-way data sets, where M variables or features are measured/computed for N observations or objects. The aim of PCA is to find the subspace of maximum variance in the M-dimensional variable space. The original variables are linearly transformed into the Principal Components (PCs). These are the eigenvectors of $\mathbf{XX} := \mathbf{X}^T \cdot \mathbf{X}$, typically for mean centered $\mathbf{X}$ and sometimes also after auto-scaling – i.e. normalizing to unit variability.

PCA follows the expression:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^t + \mathbf{E}_A, \tag{1}$$

where $\mathbf{T}_A$ is the $N \times A$ score matrix, $\mathbf{P}_A$ is the $M \times A$ loading matrix and $\mathbf{E}_A$ is the $N \times M$ matrix of residuals.

The score of a new observation in the PCA subspace is computed as follows:

$$\mathbf{t}_n = \mathbf{x}_n \cdot \mathbf{P}_A \tag{2}$$

where $\mathbf{x}_n$ is a $1 \times M$ vector representing a new observation and $\mathbf{t}_n$ a $1 \times A$ vector with the corresponding scores, while:

$$\mathbf{e}_n = \mathbf{x}_n - \mathbf{t}_n \cdot \mathbf{P}_A^t \tag{3}$$

corresponds to the residuals. Both scores and residuals are monitored in a MSPC system.

PCA can handle very large dimensional data sets. For instance, PCA is a common analysis tool in genomic data [47], which can have up to a million of variables. This capability is of utmost importance for anomaly detection because a high number of variables from multiple and variate data sources can be taken into account at the same time [48]. Also, common and anomalous patterns can be interpreted from the joined contribution of the variables involved. The more variables in the model, the more meaningful information about these patterns can be extracted. In comparison, other network analysis tools are typically limited to univariate or low dimensional time series signals [45].

3.2. Multivariate SPC based on PCA

In PCA-based MSPC, it is customary to monitor a pair of charts: the Q-statistic or SPE, which compresses the residuals [49]; and the D-statistic or Hotelling's T2 statistic [50], computed from the scores. Although it is widely recognized that the SPE provides of superior detection capability than the D-statistic [37], both statistics are complementary [2]. With the statistics computed from the calibration data, control limits at a certain confidence level can be established in the charts [38, 40, 51, 49]. Afterwards, new data are monitored using these limits. Thus, anomalies are detected when the limits are significantly or consistently exceeded. Also, making the most of the nature of latent variable models, the contribution of the variables to an anomaly signaled can be investigated with the contribution plots [43, 46, 52, 53].

Both the D-statistic and the Q-statistic for observation n can be computed from the following equations:

$$D_n = \sum_{a=1}^{A} \left(\frac{t_{an} - \mu_{\mathbf{t}_a}}{\sigma_{\mathbf{t}_a}}\right)^2 \tag{4}$$

$$Q_n = \sum_{m=1}^{M} (e_{nm})^2 \tag{5}$$

where t_{an} represents the score of the observation in the a-th component, $\mu_{\mathbf{t}_a}$ and $\sigma_{\mathbf{t}_a}$ stand for the mean and the standard deviation of the scores of that component in the calibration data, respectively, and e_{nm} represents the residual value corresponding to the m-th variable.

Following the SPC approach, the analysis is performed in two phases, as previously discussed. In phase I, available data

are inspected for special causes of variation, which are iteratively solved. Once collected data are free of special causes of variation, these are used to model the NOC for the calibration of the MSPC system. Subsequently, new data are monitored with that system. Thus, in phase II we should distinguish between calibration data, which should be free of anomalies, and monitored or test data. Notice that this distinction, highly relevant, is not made in the work by Lakhina et *al.* [5]. If this is not done, detection and diagnosis may be affected by anomalies in the PCA model, as reported in point *iii)* in Section 1 from [9]: *a large anomaly may inadvertently pollute the normal subspace*.

The scores are linear combinations of the original variables and so, according to the Central Limit Theorem, they are supposed to be approximately Normal distributed [40]. As a consequence, the D-statistic in phase I times a constant follows a beta distribution [38]:

$$D \sim \frac{(N-1)^2}{N} B_{A/2,(N-A-1)/2} \tag{6}$$

Therefore, the corresponding Upper Control Limit (UCL) for the D-statistic at significance level α is given by:

$$UCL(D)_\alpha = \frac{(N-1)^2}{N} B_{(A/2,(N-A-1)/2),\alpha} \tag{7}$$

For new incoming data in phase II, the D-statistic times a constant follows an F distribution [38]:

$$D \sim \frac{A \cdot (N^2-1)}{N \cdot (N-A)} F_{A,(N-A)} \tag{8}$$

And the corresponding UCL at significance level α is given by:

$$UCL(D)_\alpha = \frac{A \cdot (N^2-1)}{N \cdot (N-A)} F_{(A,(N-A)),\alpha} \tag{9}$$

Regarding the UCL for Q-statistic, several procedures can be used. Again, the residuals can be assumed to follow a multinormal distribution. Jackson and Mudholkar showed in [49] that an approximate critical value at significance level α is given by:

$$UCL(Q)_\alpha = \theta_1 \cdot \left[\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{10}$$

where $\theta_n = \sum_{a=A+1}^{rank(\mathbf{X})} (\lambda_a)^n$, with $rank(\mathbf{X})$ the rank of the matrix of data $\mathbf{X}$ and λ_a the eigenvalues of matrix $\frac{1}{N-1} \cdot \mathbf{E}_A^T \cdot \mathbf{E}_A$, $\mathbf{E}_A$ being the matrix of residuals; $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$; and z_α is the $100 \cdot (1-\alpha)\%$ standardized normal percentile.

Alternatively, one can use an approximation based on the weighted chi-squared distribution proposed by Box [51]. Control limits for Q-statistic that distinguish phase I and phase II can also be found [10].

To achieve an adequate performance of the monitoring charts in phase II, it is highly recommended to readjust the control limits using the calibration data in a leave-one-out basis [54, 55]. Limits are raised or lowered so that the Overall Type I (OTI) risk equals the imposed significance level α. Following the definition in [40], the OTI is the percentage of faults in the NOC calibration observations:

$$OTI = 100 \cdot \frac{nf}{N}\% \tag{11}$$

where nf is the total number of faults (*i.e.*, single observations where the statistic computed crosses the control limit) in the NOC calibration data.

3.3. Diagnosis

Once a fault is detected, a diagnosis system capable to find its root-causes is desired. The most generalized approach for diagnosis in MSPC is the contribution plots [43, 46, 52]. Contribution plots show the contribution of the variables to an anomalous value of the monitoring statistics. Generally speaking, these are bar plots where the contribution of the set of variables to a single statistic (D-st or Q-st) can be inspected. Also, control limits can be defined to signal significant contributions, which detect the variables related to a given fault that should be considered for diagnosis. According to Alcala and Qin [53], there are three main classes of contribution plots: General Descompositive Contributions (GDC), Reconstruction Based Contributions (RBC) and Diagonal Contributions (DG). The most extended approach is referred to as the Complete Decomposition Contribution (CDC) by [53], and belongs to the GDC class. The CDC index for variable i is as follows [53]:

$$CDC_i = \mathbf{x}_n \cdot \mathbf{M}^{1/2} \cdot \boldsymbol{\xi}_i^t \cdot \boldsymbol{\xi}_i \cdot \mathbf{M}^{1/2} \cdot \mathbf{x}_n^t \tag{12}$$

where $\boldsymbol{\xi}$ is a $1 \times M$ vector with zeros exception made on a one in the i-th position, and $\mathbf{M}$ is defined in accordance with the statistic diagnosed.

For the Q-statistic, $\mathbf{M}$ follows:

$$\mathbf{M} = \mathbf{P}_{-A} \cdot \mathbf{P}_{-A}^t \tag{13}$$

where $\mathbf{P}_{-A}$ are the residual components after A PCs have been extracted, defined as the eigenvectors of $\mathbf{E}_A^t \cdot \mathbf{E}_A$.

For the D-statistic, $\mathbf{M}$ follows

$$\mathbf{M} = \mathbf{P}_A \cdot \boldsymbol{\Lambda}^{-1} \cdot \mathbf{P}_A^t \tag{14}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with the first A eigenvalues of $\mathbf{X}^t \cdot \mathbf{X}$.

The original proposal by Lakhina et *al.* [5] base the diagnosis in the approach of Dunia and Qin [41], which lies into RBC methods. This original approach requires of a set of predefined faults against which new faults in incoming data can be diagnosed. A main limitation of this is that new faults, not previously seen in the past data, cannot be adequately diagnosed. RBC methods have been mainly developed by Prof. Qin and co-workers, being successfully generalized for unsupervised diagnosis (take for instance [56])

The RBC index for variable i is as follows [53]:

$$RBC_i = \frac{(\boldsymbol{\xi}_i \cdot \mathbf{M} \cdot \mathbf{x}_n^t)^2}{\boldsymbol{\xi}_i \cdot \mathbf{M} \cdot \boldsymbol{\xi}_i^t} \tag{15}$$

An alternative to contribution plots are *o*MEDA plots [57]. The *o*MEDA algorithm was designed to identify the variables which differ between two groups of observations in a latent subspace. From that point of view, it can be seen as an extension of contribution plots to compare groups of observations. If one of these groups is the centre of coordinates, the result is similar to a contribution plot. The *o*MEDA can also be used to identify the variables that most vary in a given direction contained in a subspace. That way, we can also diagnosis score trends in a projection subspace.

The *o*MEDA technique is employed as follows. Firstly, a dummy variable is designed to cover the observations of interest. Take the following example: a number of subsets of observations $\{\mathbf{C}_1, ..., \mathbf{C}_N\}$ form different clusters of anomalies. One may be interested in identifying, for instance, the variables related to the deviation of $\mathbf{C}_1$ from the NOC data $\mathbf{L}$ without considering the rest of clusters. For that, a dummy variable $\mathbf{d}$ is created so that observations in $\mathbf{C}_1$ are set to 1, observations in $\mathbf{L}$ are set to -1, while the remaining observations are left to 0. Similarly, $\mathbf{C}_1$ can be compared to the centre of coordinates (the data average) by setting $\mathbf{d}$ so that observations in $\mathbf{C}_1$ are set to 1 and the remaining to 0. Doing this with a cluster of one single observation is similar to issuing a contribution plot of that observation. Finally, values other than 1 and -1 can be included in the dummy variable if desired, which is useful for diagnosing trends in the scores. *o*MEDA is then performed using this dummy variable as follows:

$$d^2_{A,(i)} = \frac{1}{N} \cdot (2 \cdot \Sigma^d_{(i)} - \Sigma^d_{A,(i)}) \cdot |\Sigma^d_{A,(i)}| \tag{16}$$

with $\Sigma^d_{(i)}$ and $\Sigma^d_{A,(i)}$ being the weighted sum of elements for variable i in $\mathbf{X}$ and its projection $\mathbf{X}_A$ according to the weights in $\mathbf{d}$, respectively.

Two main differences of *o*MEDA with contribution plots are that the former has sign information while contribution are quadratic indexes, and that *o*MEDA specifies a contribution in a sub-space, not in a given specific statistic.

3.4. Model Calibration: Number of PCs and Normalization

A main matter of study is how to select the number of PCs A in a PCA model [35]. Recent studies claim that this decision is dependent on the application for which PCA is used [58]. For this reason, we have intentionally located this section after Section 3.2, since the application needs to be properly understood before deciding a procedure to select A. The aim of PCA in MSPC is to select the optimum division in model and residual subspaces so that the statistical distributions of $\mathbf{T}_A$ and $\mathbf{E}_A$ defined from the calibration data are representative of the distributions in incoming data, provided that the system under analysis remains in control.

Some guidelines to select A for monitoring by assessing the stability of $\mathbf{P}_A$ were suggested in [58], but it remains as an open and challenging issue. Generally speaking, the model subspace should be calibrated so that the amount of variance in the structural part of the system (eigenvectors and eigenvalues) holds in future data. The eigenvectors are imposed in the construction of the scores, while eigenvalues are imposed in the definition of the D-statistic. If the MSPC system is properly calibrated, the amount of variance satisfying those constrains should be similar in both calibration and test data. Note that the more PCs added to the model, the more constraints are imposed on that structure. Contrarily, on the residual part no constrains are imposed exception made on the amount of variance left in the residual subspace, measured in the Q-statistic. It should be noted that there is no perfect optimal A, since a certain degree of noise is always incorporated in a PCA model no matter the value of A. Here, noise is understood as non-structural variability captured in the PCA model, that is, specific variability for the calibration observations, not repeated in future data.

According to the previous discussion, it should be noted that the underestimation of A is less harmful for the MSPC system than its overestimation. If A is underestimated, the anomaly detector is not calibrated with the optimum number of constrains. The risk of underestimating A is that there may be a subset of faults that would be detected in the D-statistic with A PCs and would not be detected neither in the D-statistic nor in the Q-statistic with $A - n$ PCs. However, if A is overestimated, the MSPC system includes constraints that are not generalizable to new incoming data. This introduces a bias between the values of the D-statistic and Q-statistic in calibration and test data. In particular, the Q-statistic tends to be higher in test data than in calibration data, leading to many false alarms [12]. This effect is expected to grow with the number of parameters in the PCA model, which is a function of A and the number of variables.

In any case, the approach to select A used in the networking community [5, 8], so as to capture a high percentage of the variability of the data, is discouraged, especially when the number of variables is very large [4]. A proper selection of the number of PCs as discussed above and the readjustment of the control limits according to the OTI risk are adequate means for avoiding the problem claimed at point *i)* in Section 1 from [9]: *the supposed sensitivity of the anomaly detector with the parameter A*.

Another relevant choice for model calibration is that of the data normalization [59]. Most common normalization operations in MSPC are mean-centering and auto-scaling in the observations mode. After mean-centering, PCA is focused on variability. Lakhina et *al.* [5] used this normalization. The auto-scaling operation is a mean-centering followed by a scaling procedure to set the variance of the variables to 1. This is recommended when variables with non-comparable units are present.

As the number of variables in MSPC grows, different normalizations may drive to a very different set of detected anomalies. The auto-scaling operation homogenizes the relevance of the set of variables in the MSPC system, but at the same time may amplify the noise. On the other hand, mean-centering focuses the MSPC system on variables with high variability. This approach may not be the most interesting in network anomaly detection, since security related variables tend to present low variability. When it is possible to define a degree of relevance of the set of variables by an expert, it is recommended to perform an auto-scaling operation followed by a weighted operation where the

relevance of the variables is considered. This allows to focus the PCA model and monitoring statistics on the most relevant variables.

4. Data pre-processing

Data pre-processing here is understood as the computations needed to transform the data collected from a monitored system into suitable input data for the supervision modules. Since PCA is based on (co)variance, which is a quantitative measure, information needs to be transformed into quantitative values. Notice that, in the field of MSPC, the term pre-processing commonly refers to data normalizing operations previously discussed.

As already defined, PCA is suited to analyze two-way data sets, which contain a number of observations (rows) of a number of features or variables (columns) Therefore, to apply a multivariate statistical monitoring procedure to an industrial process or a communication network, some appropriate variables need to be measured on that process or network. Each complete measurement of the vector of variables is what we call an *observation*. The definition of the observations and variables is a main difference between MSPC and MSNM, and the reason why the latter is probably more complex than the former. In MSPC, monitored variables, like temperatures, pressures, concentrations, etc. are directly measured from the process. Thus, none or very little pre-processing is needed. Observations are typically ordered in time, for regular or variable sampling rates, and again with little or none pre-processing, typically interval-wise averaging or sampling.

In MSNM we have the opposite situation. In a network, most of the information comes in the form of logs or network traffic, information that cannot be directly used in a MSNM set-up. Rather, logs and network traffic need to be translated into quantitative variables, and there is a bunch of possibilities to do so. This is typically referred to as data parsing or feature engineering [45]. Furthermore, the definition of the observations in MSNM is not straightforward. Although observations are typically ordered in time, it may be interesting to define the observations in terms of relevant entities in a network, such as source or destination IPs or service ports. This makes MSNM more challenging but also more flexible and powerful than traditional MSPC.

A main challenge in MSNM is the logs parsing, since almost each vendor defines its own log format. Although there have been some standardization efforts on the format, *e.g.,* the Common Event Expression (CEE - http://cee.mitre.org/), there is a general lack of adoption. This makes the analysts to devote a significant percentage of time to log parsing. The key for the selection of variables in MSNM systems is to identify general means of translating log information into quantitative variables.

Lakhina et *al.* [5] proposed the definition of counters as quantitative variables. The counters were restricted to counts of packets and bytes arranged by origin-destination border gateways of backbone networks. Camacho et *al.* [48] generalized this definition to consider several sources of data. They proposed the feature-as-a-counter approach, so that variables are basically counters for the number of associated events. Each variable is defined as the number of times, n_i^w, a given event i takes place in the logs during a given time window w. This is a general definition suitable for most of the types of information of interest in anomaly detection: traffic volume (*e.g.,* number of incoming or outgoing packets/flows within a given period), application-specific traffic (*e.g.,* number of requests to a given port or group of ports), location-specific traffic (*e.g.,* number of packets from/to a specific subnet or group of addresses), specific events in the logs (*e.g.,* number of logs with a specific event), events severity (*e.g.,* number of logs with a specific event code), etc. The window size w may be defined so that scarce measurement matrices are avoided. That is, it should be big enough so that a given event takes place more than once in most intervals. Furthermore, by properly selecting w in the parsing, the combination of information from different and variate sources of data is simplified [48]: data sources can be combined by appending the corresponding data matrices in the mode of the variables. If the sampling rate of a data source s is faster than the final sampling rate, cumulative or average values can be used to compute the corresponding variables values for the observations. If the sampling rate of a data source is slower than the final sampling rate, proportional or repeated values can be used instead. Also, some form of interpolation may be used if necessary.

Another class of variables are those representing a sample distribution. These are commonly more suitable than counters to summarize the information in traffic data, and less sensitive to packet sampling according to [39]. For instance, we may be interested not only in the cumulative throughput in an access link during a given time window w, but also in the distribution of packets or flows sizes in that interval. Histograms are useful artifacts to transform a distribution into one or more quantitative variables. The histogram is composed of a number of quantitative variables (counters): $Z = \{n_i, i = 1, ..., L\}$. These variables can be directly entered into the system. However, this approach may lead to a huge number of variables. Although PCA can handle very large numbers of variables, this approach may lead to very scarce data matrices and problems in the interpretation of the results. Alternatively, the histograms can be summarized, for which measures of central tendency (*e.g.*, averages) and dispersion (*e.g.*, standard deviations) may be adequate. Also, Lakhina et *al.* proposed [39] the use of entropy to summarize the information in very large histograms, such as those obtained in terms of source IPs. The entropy is defined as:

$$H(Z) = -\sum_{i=1}^{L}\left(\frac{n_i}{S}\right) log_2\left(\frac{n_i}{S}\right) \tag{17}$$

with:

$$S = \sum_{i=1}^{L} n_i \tag{18}$$

Callegari et *al.* [60] also proposed the use of the Kullback-Leibler (K-L) divergence to capture dynamical information.

The K-L divergence in time is expressed as:

$$D_{KL}^{t} = \sum_{i=1}^{L} n_i^{(t-1)} log \frac{n_i^{(t-1)}}{n_i^t} \tag{19}$$

where n_i^t refers to the i-th counter in the t-th sampling interval. In the next section, alternative procedures to capture dynamical information that simplify the diagnosis of anomalies are presented.

The comparison among counters and quantitative variables based on histograms is out of the scope of the present paper, which is more concerned with the correct development of a PCA-based monitoring system. The interested reader is pointed to references [39], [48] and [60]. However, it should be noted that the proper selection of variables for anomaly detection is an interesting research topic, which may, by itself, deserve publication like it happens in other areas like e.g. pattern recognition and medical research.

5. Handling Dynamics

Brauckhoof et *al.* [8] suggest that the supposed limitations reported in [9] for PCA anomaly detection come from the fact that PCA does not consider the temporal correlation in the data. However, this claim is misled. PCA can model any type of linear correlation provided that the data arrangement is properly chosen. Dynamics can be incorporated in a PCA model adding Lagged Measurement Vectors (LMVs) [61, 62]. Figure 2 shows how data are arranged to incorporate LMVs in a PCA model and the resulting covariance matrix when this is done. This solution has been referred to as Dynamical PCA [61] or a derivation from the Karhunen-Loeve transform for functional analysis [8]. However, it should be noted that traditional PCA is applied over the modified data matrix, so that no change in the multivariate model is done but only a specific data arrangement is performed.

From the previous figure it is easy to understand the effect of the addition of LMVs. Dynamics of order d are built in the model by matrices $X_{t-d}^T \cdot X_t$. The more LMVs added, the more dynamic information included in the model. As stated in [54], depending on the nature of the process, part of the dynamic information may be negligible and it could be advantageous not to incorporate it to the model for the sake of parsimony (the use of minimum number of parameters). Furthermore, research on MSPC [4] has shown that the incorporation of dynamical information into the PCA model does not necessarily improve the monitoring performance: LMVs may introduce auto-correlation that may distort the shape of the monitoring statistics.

6. Experimental Section

In this section, two case studies are used to illustrate the MSNM approach. In the first one, data from the VAST 2012 2nd mini challenge, online available at [63], are used to illustrate the main steps in the design of a MSNM system, including the choice of normalization, number of PCs, control charts, diagnosis methods, the definition of the variables and the dynamical order in the model.

A key issue that cannot be illustrated with a real data set is the convenience of the statistical approach based on two phases, since commonly the network configuration needs to be modified for solving special sources of variation. This way, in the second case study the steps in phases I and II are shown for a controlled scenario, where special sources of variation are identified and eliminated. Moreover, an experiment with detailed ground-truth is performed to compare MSNM with alternative approaches.

To illustrate the development of the MSNM system, the MEDA Toolbox [64] for MATLAB will be employed.

6.1. Case Study I: VAST 2012 2nd mini challenge

The VAST 2012 2nd mini challenge [63] presents a corporate network where security incidents occur during two days. In particular, some staff members report unwanted messages and a non-legitimate anti-virus program appearing on their monitors. Also, their systems seem to be running more slowly than usual. In summary, a forensics operation is required to discover the most relevant security events and their root causes. Around 4,000 workstations and approximately 1,000 servers operate 24 hours a day. The data provided with the VAST 2012 mini challenge 2 consist of Cisco ASA firewall logs including a total of 23,711,341 data records, and IDS logs including 35,948 data records. The data set details are available at [63], from which the high complexity of the problem should be concluded.

Data from the firewall and IDS logs in the VAST 2012 mini challenge 2 have been parsed into M-dimensional vectors representing time intervals of one minute, as the resolution of the IDS entries prevent us from using shorter intervals. A total of 2,350 observations, each one with the information for one minute, are obtained.

For every sampling period of one minute, we have defined a set of 112 variables that represent the information from the two data sources: 69 variables for the firewall log and 43 for the IDS log (see Table 1). By using the same sampling period, both data sources are seamlessly combined by appending the data matrices. Every variable is labeled as `source_typelabel`, where `source` indicates if the variable is coming from the firewall (fw) or from the IDS (ids) logs, `type` indicates the type of the variable (*e.g.*, `ip` stands for a range of IP addresses and `p` for port), and `label` gives some specific information. For example, the variable `ids_pdns` collects the number of IDS logs related to incidents where the DNS port is present.

Data are split into a calibration set composed of the first 1,000 observations and a test set with the remaining. Although the network cannot be considered to be under NOC, for the sake of illustration we will approximate that situation by detecting and discarding outliers in the calibration data. This is done using the D-statistic and the Q-statistic, and for the several normalization and numbers of PCs considered afterwards. In all the cases, a very similar set of outliers was detected. The remaining observations where used as a common calibration set for the MSNM systems described below.

In an MSNM system, the first step is to select the number of PCs and perform the normalization. Figure 3 shows the curve of residual variance in terms of the number of PCs for the calibration data. It also includes a form of low-weight cross-validation named column-wise cross-validation (*ckf*) [65]. The *ckf* aids in the selection of PCs, and the minimum value in the curve should be chosen according to this criterion. However, it is not clear that this is an adequate selection criterion for MSNM [58]. We simply use it here as an example of PCs selection tool commonly used in the MEDA Toolbox [64]. Figure 3 shows both the result after mean-centering and after auto-scaling.

Recall that Lakhina et *al.* [5] used a mean-centering normalization. This would lead to Figure 3(a). According to their criterion for PCs selection, one PC would be enough. This is also supported by the *ckf* curve. Alternatively, if auto-scaling is used, the number of PCs should be selected from Figure 3(b). In this case, Lakhina et *al.* would select a large number of PCs (16 PCs app.), and following the *ckf* criterion we may select 9 PCs. This contradicts experimental results [12] that show that the number of PCs should be low when the number of variables is large, also in agreement with our previous discussion in Section 3. This is because any PC incorporates a certain degree of noise that will not be present in future data. If the model has too many PCs, the incorporated noise makes the MSPC system show too many false alarms in the Q-statistic, while the D-statistic may also be affected. In Figure 4(a) we illustrate this effect. The figure shows the ratio between the D-statistic of test and calibration observations and the same for the Q-statistic in terms of the number of PCs. The ideal ratio, equal to 1 so that the control limits computed from calibration data are valid for test data, is also shown as a reference. Ratios are generally high, around a value of 4, showing that in this specific division in calibration and test, test observations show higher variability than calibration ones on average. Also, for a high enough number of PCs the Q-statistic ratio tends to diverge. This causes a high number of false alarms in that statistic. In Figure 4(b) we show a re-sampling of the data, where calibration and test observations were randomly chosen. In this example, the ratios do start in 1 showing a similar variability in calibration and test, but again for a large enough number of PCs they tend to get unstable. These figures confirm our claim regarding that low numbers of PCs should be preferred to high numbers, so that the ratios remain in the stable part and a large number of false alarms is avoided. This shows that the PCs selection procedures considered so far in networking [5] are not adequate. However, a good procedure is still a matter of research.

In Figure 5, the D-statistic and Q-statistic charts[1] for the number of PCs that would be selected by Lakhina et *al.* [5] both after mean-centering and auto-scaling are shown. Notice that Lakhina et *al.* would only have used the Q-statistic in their monitoring system. However, there are anomalies that are only detected in the D-statistic. It is the case of observations #505 and #637 in Figure 5(a) and #480 and #271 in Figure 5(b). This problem grows with the number of PCs incorporated into the model, since more variability is captured by the model and so by the D-statistic. This shows that the monitoring scheme based only in the Q-statistic considered in [5] is not adequate.

In Table 2, the four time intervals with the highest value of both the D-statistic and the Q-statistic for different PCA models are compared. Following the previous discussion on the selection of the number of PCs according to several criteria, the considered models are the model with 1 PC for mean-centered data and the models with 1 PC, 9 PCs and 16 PCs for auto-scaled data. Recall that both statistics are complementary, which means that the main anomalies detected by each MSNM system are the union of those detected by each chart. The results show a certain degree of variability among the MSNM systems. Still, there are several common detections. For instance, the detections in the D-statistic in the first two models and in the Q-statistic in the last three models are exactly equal. However, several anomalies vary from one system to another. We will try to elucidate the reason for that variability in the following.

In Figure 6, *o*MEDA plots and CDC and RBC contribution plots are shown in the first, second and third column, respectively. Diagnosis plots were computed for both the D-statistic and the Q-statistic, for several of the models and for a number of anomalies selected from Table 2: namely, #418, #480, #526 and #505.

The profile of the three types of diagnosis plots for the same anomaly coincides in most situations, except for anomalies #526 and #505. For the latter, both *o*MEDA and CDC plots yield the same relevant variables, but RBC fails to do so. In fact, the diagnosis by RBC seems to show an artefact. Although this example will be further studied in the following, it is out of the scope of this paper to investigate statistically significant differences among the diagnosis methods. However, it should be noted that all methods perform adequately in most anomalies considered here. For instance, the diagnosis plots in Figures 6(a) and 6(b), issued for the anomaly #418 with different models, are coherent. It is also apparent from Figures 6(c) and 6(d) that anomalies #505 and #526 are related to the same phenomenon.

In Tables 3, 4 and 5, the main variables according to the diagnosis plots for faults #418, #505, #526 and #480 are presented. In the first and last tables, diagnosis results are equivalent regardless of the method employed: O (*o*MEDA), C (CDC) or R (RBC). Fault #418 is related to IDS alarms of high priority (ids_prio1) reporting potential corporate privacy violation (`ids_privacy`) in the DNS service (`ids_pdns` and `ids_ldns`). Fault #480 is related to FW logs of Syslog Severity Level 4 (`fw_syswarn`) related to %ASA-4-1 messages (`fw_asa4` and `fw_asa41`), and the FTP service (`fw_pftp`). A fast inspection to interval #480 in the corresponding log showed a large number (more than 700) of %ASA-4-106023 messages reporting the ACL blockage of FTP connections to an outside server. These 700 logs were found among a much larger (12,000) %ASA-6 messages. However, this huge number of %ASA-6 messages was maintained during the whole data capture, and it was un-

[1]Control limits are not shown in the figure to avoid confusion: this data should be interpretatively analyzed with phase I control limits, and solving for the problems found, probably performing modifications on the network configuration. This procedure is illustrated in the second example.

derstood by the MSNM system as part of the common variation in the network. This example illustrates the great capability of MSNM for discovering non-common events among a much larger amount of common events. This also shows the clear necessity of a phase I procedure in order to identify special causes of variation that should be avoided prior to the calibration of the final anomaly detector. Thus, this excess of %ASA-6 messages could be understood as such special cause. Anomalies #505 and #526 share the same highlighted variables for *o*MEDA and CDC, and therefore the same diagnosis: the faults correspond to a larger than usual amount of sysinfo logs (`fw_sysinfo`) with ASA-6 messages (`fw_asa6`) reflecting outgoing (`fw_outbound`) HTTP connections (`fw_phttp` and `fw_tcp`) from the WorkStations of the network (`fw_ipws` and `fw_pnstd`).

To further corroborate the diagnosis, the profiles of some of the variables highlighted in the diagnosis plots are shown in Figure 7. Some anomalies are annotated. For instance, time interval #418 is a clear anomaly in variable `ids_prio1`, validating the detection in the MSNM based on the PCA model with 1 PC from auto-scaled data (Table 2) and the diagnosis in Figure 6(a) and Table 3. Also, time interval #480 presents an anomalous value for variable `fw_pftp`, validating the detection in the MSNM based on the PCA model with 9 PCs from auto-scaled data (Table 2) and the diagnosis in Figure 6(e) and Table 5. Finally, time intervals #505 and #526 present anomalous values for variable `fw_pnstp`, validating the detection in the MSNM based on the PCA model with 1 PC from mean-centered data (Table 2) and the diagnosis by *o*MEDA and CDC in Figures 6(c) and 6(d) and Table 4. It is worth noting that in Figure 7(b) time intervals #505 and #526 do not show an anomalous value, which disagrees with the diagnosis by RBC in Figures 6(c) and 6(d). This confirms that these diagnosis plots show an artefact.

Results in Figure 7 show that the anomalies detected using different MSNM systems are all correct and accurate. This contradicts the criticism of [9] specified in the introduction, in particular that corresponding to point *i)*: *the false positive rate is very sensitive to small differences in the number of principal components in the normal subspace*. This is the case if the approach of Lakhina et *al.* is followed, but not for MSNM. Rather, it should be noted that all the detections are in fact correct, though different MSNM systems focus on different types of anomalies. Thus, in the case of the mean-centered MSNM system, the detection is focused on variables with higher variance (see for instance the variance in Table 4) while in the auto-scaled MSNM system, variables with low variability are promoted to highlight anomalies in them (see for instance the variance in Table 3). Therefore, the MSNM system should be configured taking into account, if possible, the main targets of detection. Anomalies detected are always correct, in the sense that they are always motivated by some source of anomalous behavior. However, that source might not be of interest considering the goal of the MSNM system. The proper alignment of actual detections with detection targets is the main challenge in the development of a MSNM system.

Finally, Figure 8 shows the monitoring charts for a dynamical model with 1 LMV and 1 PC after mean-centering. The main faults signaled (Table 6) are essentially the same as those for the statical model with 1 PC after mean-centering (Table 2), but faults are signaled twice as a consequence of the autocorrelation induced in the statistics. This confirms the findings in [4]. From that reference and the work performed here, and in disagreement with [8], we claim that the use of LMVs or dynamical models is not a solution to any problem that may or may not exist in PCA-based anomaly detection, provided that this is adequately performed. Certainly, this solution was proposed for the approach of Lakhina et *al.*, not for MSNM.

6.2. Case Study II: Controlled Scenario

The experimental setup for the controlled scenario consists of a network of 100 virtual machines running Linux Mint 17, with the topology shown in Fig. 9. We also include a Linux machine running Apache 2.4.7 web server, and a Netflow inspector collecting information of flows between the server and the rest of the network. The used implementation for the Netflow inspector is the kernel module ipt_NETFLOW 2.6.x-3.x for iptables [66]. The flows are collected with the following configuration: `active timeout=1800s`; `inactive timeout=15s`; `netflow protocol=v5`.

In order to include background normal traffic in the trace to be analyzed, the clients generate HTTP requests against the server following an exponential distribution for the inter-request time, with a mean value of 40 seconds between consecutive requests. There are 100 URLs randomly selected by the clients, all of them corresponding to resources of different sizes. To illustrate the use of MSNM during phase I, an anomaly is included in the trace: a malware located in the server performs a stealth nmap scanning every 5 minutes to the range 192.168.56.100/28, affecting 3 active machines in the network.

Netflow information is transformed to observations with 148 variables using the feature-as-a-counter approach for a sampling period of one minute. Table 7 shows a brief summary of the number of variables defined for this dataset. As an example to clarify the choice for the variables, we see that there are four variables defined to represent IP addresses, *e.g.*, number of source/destination public/private IP addresses, *i.e.*, `src_ip_private`, `src_ip_public`, `dst_ip_private`, `dst_ip_public`. Most variables are related to specific ports/services.

In Fig. 10, the control charts for phase I are shown. The D-statistic chart shows a pattern every 5 minutes, approximately. This pattern does not lead to the detection of anomalies if theoretical control limits are used in the chart. However, theoretical limits are based on assumptions that may not hold in all circumstances, and clear patterns like the one presented in the chart should be checked. If the observations with high values are compared to the average observation using *o*MEDA, the plot in Fig. 11(a) is obtained. This plot shows that high values of the D-statistic are the result of observations where most variables obtain a larger value than usual. Since the variables highlighted are related to specific ports, the straightforward diagnosis is that a port scanning is taking place every 5 minutes. Considering that the chart in Fig. 10(a) identifies the minutes in which the scanning took place, the security staff can

inspect the Netflow data corresponding to that period and find the source of scanning. With this information, the malware associated can be disabled on the infected machine, solving the problem, *i.e.*, the special cause of variation. In the Q-statistic chart of Fig. 10(b), a single outlier is found at minute 30. The diagnosis based on *o*MEDA in Fig. 11(b) highlights protocols UDP, DNS and usage of public IPs. Inspecting the Netflow data at that interval, it was found that this anomaly was related to an excess of DNS queries to local server at IP address 127.0.1.1, and HTTP traffic directed to the public addresses 204.45.82.194 and 91.189.92.150. We have checked that these addresses correspond with public Linux Mint repositories, thus concluding that the analyzed event is a result of automatic software update. This was not concluded to be an anomaly. Instead, it was considered a common source of variability and no additional action was taken in the network configuration. To avoid future false alarms in a MSNM, DNS and HTTP flows with the specified pattern where filtered out in the Netflow sensor.

In a second experiment, after disabling the scanning malware from the server and filtering out DNS and HTTP flows related to software update, a trace of traffic under NOC was obtained. The corresponding monitoring charts are shown in Fig. 12. The highest statistics in the calibration data were diagnosed as common causes of variability, and no additional action was performed. Thus, we can consider that the network is under statistical control and we can proceed with phase II. The charts show both the theoretical control limits at phase II and the leave-one-out adjusted control limits, the latter being preferred for a reduction of false alarms in the monitoring system.

In a third experiment, additional traffic to test the monitoring system was generated with a new anomaly, in which an intrusion was performed to the server by using a vulnerable application and injecting a `reverse_tcp` payload that automatically connected to the hackers machine at TCP port 4444 (default port in the Metasploit framework). This connection was used to exfiltrate data by periodically sending server log information about the HTTP requests received (file `access.log` of the Apache server). The monitoring charts are shown in Fig. 13. Anomalies are only detected in the Q-statistic chart. In the first anomaly (instant t=1), we checked with *o*MEDA that the relevant variables point out to a higher amount of traffic than expected. After inspection, this anomaly was regarded as a false alarm, as we confirmed that the increase in the traffic was due to the statistical nature of the HTTP requests from legitimate clients to the server. On the contrary, the second anomaly was related to the Metasploit port as well as a relevant increase in the number of packets. The inspection of the corresponding Netflow data led us to identify the source of the attack.

A final, more complex, experiment was carried out to compare the MSNM system with the approaches of Lakhina et *al.* [5, 39] and Brauckhoof et *al.* [8]. For that, the ground-truth specified in Table 8 was defined. The experiment was carried out with the calibration data defined for phase II. It started in a state of NOC, with the 100 HTTP clients operating. During the experiment, several DoS attacks were performed, which differ in the rate of the attack and whether spoofing was defined with prior knowledge of the network segment or not. Moreover, two data exfiltrations with Metasploit, similar to that in the previous experiment, were implemented. Finally, there was a period where the number of HTTP clients were reduced, which could be a consequence of a malfunction in the network or the service.

There are some comments in due regarding the use of Netflow as the sensor in this experiment. First, when the high rate DoS attack took place, the sensor got overloaded. For this reason, the DoS was only detectable in Netflow data during the first two minutes of the attack. A higher cache might have prevented this problem, but then a higher DoS could also cause this overflow. Second, during the first sampling time and the last three sampling times, traffic was lower than in calibration data.

Taking the ground-truth and these comments into consideration, we defined the time bins in which detections should be flagged, and metrics like recall, specificity and accuracy, the latter the most general performance measure, can be computed. This was done for the approaches of Lakhina et *al.* [5, 39], Brauckhoof et *al.* [8] and the proposed MSNM system. Results are shown in Table 9. Those approaches in bold letters in the first column are the ones in which we have scrupulously followed the recommendations of the authors regarding the preprocessing and the selection of the number of PCs. However, for a wider comparison between Lakhina et *al.* and the MSNM system under the same conditions, we have compared them for different preprocessing methods (mean-centering and autoscaling) and numbers of PCs. Results for Lakhina et *al.* show the sensitivity problem claimed by Ringberg et *al.* [9]. Depending on the choice (PCs or preprocessing) the performance of the method can be degraded to a large extent. Including dynamics in the monitoring system, as proposed by Brauckhoof et *al.*, does not lead to a real improvement. However, if both the D-statistic and Q-statistic are used, like we propose in the MSNM system, the performance is improved in comparison to Lakhina et *al.* and the sensitivity problem vanishes. In fact, MSNM detection results improve up to 20 points on the Lakhina et *al.* and Brauckhoof et *al.* proposals.

7. Conclusion

The multivariate approach based on Principal Component Analysis (PCA) for anomaly detection in networking has been developed during the last decade. This approach bears a number of differences to the more developed PCA-based Multivariate Statistical Process Control (MSPC) approach in the industrial processing and chemometric literature. In this paper, we show with examples that these differences are not justified and we coin the name Multivariate Statistical Network Monitoring (MSNM) for the application of MSPC in networking, that we support. Using MSNM, the limitations reported in the literature for the use of PCA in networking are effectively avoided.

By inheriting the MSPC approach in networking, a large amount of solutions from the industrial/chemometric community can be directly translated to the network monitoring problem. Thus, there is a vast literature on missing data estimation [67], [68], [69], data fusion [70], [71], hypothesis testing [72], [73], data equalization [74], [75], [76], data preprocessing [59],

three-way/n-way modelling [2, 40] and other data analysis procedures using multivariate models like PCA. All these methods conform a powerful tool set that provides the analyst with high capabilities for network monitoring and supervision.

Though the MSPC theory is already well-developed, there are a number of challenges that need further study. Among others, we highlight how to select the number of PCs to reduce the incorporation of noise in the model and how to best incorporate dynamics into the model in terms of detection and diagnosis ability. Furthermore, in particular for MSNM, a lot of work on how to define normalization, pre-processing and data arrangement, depending on the data sources and according to specific goals like network security, needs to be performed.

Acknowledgement

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through project TIN2014-60346-R. We would like to thank Marta Fuentes García for useful comments on the contribution plots.

References

[1] Y. Chen, T. Kourti, J. F. MacGregor, Analysis and monitoring of batch processes using projection methods: An evaluation of alternative approaches, in: Chemometrics and Analytical Chemistry, Seattle, WA, 2002.

[2] P. Nomikos, J. F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE Journal 40 (8) (1994) 1361–1375. doi:10.1002/aic.690400809.

[3] X. Hu, R. Subbu, P. Bonissone, H. Qiu, N. Iyer, Multivariate anomaly detection in real-world industrial systems, in: IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), 2008, pp. 2766–2771.

[4] J. Camacho, J. Picó, A. Ferrer, On-line monitoring of batch processes based on PCA: Does the modelling structure matter?, Analytica Chimica Acta 642 (2009) 59–68.

[5] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, ACM SIGCOMM Computer Communication Review 34 (4) (2004) 219. doi:10.1145/1030194.1015492.

[6] S. P. V. Chatzigiannakis, G. Androulidakis, Improving network anomaly detection effectiveness via an integrated multi-metric-multi-link (M3L) PCA-based approach, SECURITY AND COMMUNICATION NETWORKS 2 (2009) 289–304. doi:10.1002/sec.

[7] G. Münz, Dissertation Traffic Anomaly Detection and Cause Identification Using Flow-Level Measurements, 2010.

[8] D. Brauckhoff, K. Salamatian, M. May, Applying PCA for traffic anomaly detection: Problems and solutions, Proceedings - IEEE INFOCOM (2009) 2866–2870doi:10.1109/INFCOM.2009.5062248.

[9] H. Ringberg, A. Soule, J. Rexford, C. Diot, Sensitivity of PCA for traffic anomaly detection, ACM SIGMETRICS Performance Evaluation Review 35 (1) (2007) 109. doi:10.1145/1269899.1254895.

[10] A. Ferrer, Latent Structures-Based Multivariate Statistical Process Control: A Paradigm Shift, Quality Engineering 26 (1) (2014) 72–91. doi:10.1080/08982112.2013.846093.

[11] J. MacGregor, T. Kourti, Statistical process control of multivariate processes, Control Engineering Practice 3 (3) (1995) 403–414. doi:10.1016/0967-0661(95)00014-L.

[12] J. Camacho, New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes., PhD Thesis, Universidad Politcnica de Valencia, Valencia, 2007.

[13] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers & Security 28 (2009) 18–28.

[14] M. Bhuyan, D. Bhattacharyya, J. Kalita, Network anomaly detection: Methods, systems and tools, IEEE Communications Surveys & Tutorials 16 (1) (2014) 303–336.

[15] H. Om, T. Hazra, Statistical techniques in anomaly intrusion detection system, International Journal of Advances in Engineering & Technology 5 (1) (2012) 387–398.

[16] A. Kanaoka, E. Okamoto, Multivariate statistical analysis of network traffic for intrusion detection, in: 14th. International Workshop on Database and Expert Systems Applications (DEXA'03), 2003, pp. 1–5.

[17] M. Shyu, S. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, in: IEEE Foundations and New Directions of Data Mining Workshop (ICDM'03), 2003, pp. 171–179.

[18] G. Qu, S. Hariri, M. Yousif, Multivariate statistical analysis for network attacks detection, in: 3rd. ACS/IEEE International Conference on Computer Systems and Applications, 2005, pp. 1–6.

[19] D. Bodenham, N. Adams, Continuous monitoring of a computer network using multivariate adaptive estimation, in: 13th. International Conference on Data Mining Workshops, 2013, pp. 311–318.

[20] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, N. Taft, In-Network PCA and anomaly detection, in: Neural Information Processing Systems (NIPS), 2006, pp. 617–624.

[21] R. Kwitt, U. Hofmann, Unsupervised anomaly detection in network traffic by means of robust PCA, in: International Multi-Conference on Computing in the Global Information Technology (ICCGI'07), 2007, pp. 1–4.

[22] S. Hakami, Z. Zaidi, B. Landfeldt, T. Moors, Detection and identification of anomalies in wireless mesh networks using Principal Component Analysis (PCA), in: International Symposium on Parallel Architectures, Algorithms, and Networks, 2008, pp. 266–271.

[23] B. Rubinstein, B. Melson, L. Huang, A. Joseph, S. lau, N. Taft, D. Tygar, Compromising PCA-based anomaly detectors for network-wide traffic, Tech. rep. (2008).

[24] H. Kim, S. Lee, X. Ma, C. Wang, Higher-order PCA for anomaly detection in large-scale networks, in: 3rd. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2009, pp. 85–88.

[25] Y. Liu, L. Zhang, Y. Guan, Sketch-based streaming PCA algorithm for network-wide traffic anomaly aetection, in: International Conference on Distributed Computing Systems, 2010, pp. 807–816.

[26] R. Magan-Carrion, J. Camacho, P. Garcia-Teodoro, Multivariate statistical approach for anomaly detection and lost data recovery in wireless sensor networks, International Journal of Distributed Sensor Networks In press (2015) 1–40.

[27] M. Livani, M. Abadi, Distributed PCA-based anomaly detection in wireless sensor networks, in: International Conference for Internet Technology and Secured Transactions (ICITST), 2010, pp. 1–8.

[28] M. Dusi, C. Vitale, S. Niccolini, C. Callegari, Distributed PCA-based anomaly detection in telephone networks through legitimate-user profiling, in: IEEE International Conference on Communications (ICC), 2012, pp. 1107–1112.

[29] C. Callegari, L. Gazzarini, S. Giordano, M. Pagano, T. Pepe, A novel PCA-based network anomaly detection, in: IEEE International Conference on Communications (ICC'11), 2011, pp. 1–5.

[30] M. Xie, S. Han, B. Tian, Highly efficient distance-based anomaly detection through univariate with PCA in wireless sensor networks, in: IEEE International Conference on Communications (ICC'11), 2011, pp. 564–571.

[31] S. Novakov, C. Lung, I. Lambadaris, N. Seddigh, Studies in applying PCA and wavelet algorithms for network traffic anomaly detection, in: IEEE 14th. International Conference on High Performance Switching and Routing, 2013, pp. 185–190.

[32] A. Delimargas, E. Skevakis, H. Halabian, I. Lambadaris, N. Seddigh, B. Nandy, R. Makkar, Evaluating a modified PCA approach on network anomaly detection, in: 5th. International Conference on Next Generation Networks and Services (NGNS), 2014, pp. 124–131.

[33] D. Liu, C. Lung, N. Seddigh, B. Nandyu, Entropy-based robust PCA for communication network anomaly detection, in: IEEE/CIC International Conference on Communications in China (ICCC), 2014, pp. 171–175.

[34] Y. Kanda, R. Fontugne, K. Fukuda, T. Sugawara, ADMIRE: Anomaly detection method using entropy-based PCA with three-step sketches, Computer Communications 36 (2013) 575–588.

[35] J. E. Jackson, A user's guide to principal components, Wiley series in probability and mathematical statistics. Probability and mathematical statistics, Wiley-Interscience, 2003.
[36] J. V. Kresta, J. F. Macgregor, T. E. Marlin, Multivariate statistical monitoring of process operating performance, The Canadian Journal of Chemical Engineering 69 (1) (1991) 35–47. doi:10.1002/cjce.5450690105.
[37] B. M. Wise, N. L. Ricker, D. F. Veltkamp, B. R. Kowalski, Theoretical basis for the use of principal component models for monitoring multivariate processes, Process Control and Quality 1 (1) (1990) 41–51.
[38] N. D. Tracy, J. C. Young, R. L. Mason, Multivariate Control Charts for Individual Observations, Journal of Quality Technology 24 (2) (1992) 88–95.
[39] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, ACM SIGCOMM Computer Communication Review 35 (4) (2005) 217. doi:10.1145/1090191.1080118.
[40] P. Nomikos, J. MacGregor, Multivariate Statistical Process Control Charts for Montoting Batch Processes (1995). doi:doi:10.1016/0967-0661(95)00014-L.
[41] R. Dunia, S. Joe Qin, Subspace approach to multidimensional fault identification and reconstruction, AIChE Journal 44 (8) (1998) 1813–1831.
[42] T. Kourti, P. Nomikos, J. F. MacGregor, Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS, Journal of Process Control 5 (4) (1995) 277–284. doi:10.1016/0959-1524(95)00019-M.
[43] J. A. Westerhuis, S. P. Gurden, A. K. Smilde, Generalized contribution plots in multivariate statistical process monitoring, Chemometrics and Intelligent Laboratory Systems 51 (2000) 95–114.
[44] T. J. Boardman, The Statistician Who Changed the World: W. Edwards Deming, 1900-1993, The American Statistician 48 (3) (1994) 179–187.
[45] R. Marty, Applied Security Visualization, Pearson Education, USA, 2008.
[46] T. Kourti, J. F. MacGregor, Multivariate SPC methods for process and product monitoring, Journal of Quality Technology 28 (4).
[47] H. Milting, A. Kassner, C. Oezpeker, M. Morhuis, B. Bohms, J. Boergermann, J. Gummert, Genomics of Myocardial Recovery in Patients with Mechanical Circulatory Support, The Journal of Heart and Lung Transplantation 32 (4, Supplement) (2013) 229. doi:http://dx.doi.org/10.1016/j.healun.2013.01.582.
[48] J. Camacho, G. Macia-Fernandez, J. Diaz-Verdejo, P. Garcia-Teodoro, Tackling the big data 4 vs for anomaly detection, Proceedings - IEEE INFOCOM (1) (2014) 500–505. doi:10.1109/INFCOMW.2014.6849282.
[49] J. E. Jackson, G. S. Mudholkar, Control procedures for residuals associated with Principal Component Analysis., Technometrics 21 (1979) 331–349.
[50] H. Hotelling, Multivariate Quality Control. Techniques of Statistical Analysis, MacGraw-Hill, New York, 1947.
[51] G. E. P. Box, Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems: Effect of Inequality of Variance in One-Way Classification, The Annals of Mathematical Statistics 25 (1954) 290–302.
[52] J. F. MacGregor, C. Jaeckle, C. Kiparissides, M. Koutoudi, Process monitoring and diagnosis by multiblock PLS methods, AIChE Journal 40 (5) (1994) 826–838.
[53] C. F. Alcala, S. Joe Qin, Analysis and generalization of fault diagnosis methods for process monitoring, Journal of Process Control 21 (3) (2011) 322–330.
[54] J. Camacho, J. Picó, Online monitoring of batch processes using multi-phase principal component analysis, Journal of Process Control 16 (10) (2006) 1021–1035. doi:10.1016/j.jprocont.2006.07.005.
[55] H. Ramaker, E. N. M. van Sprang, J. A. Westerhuis, S. P. Gurden, A. K. Smilde, F. H. van der Meulen, Performance assessment and improvement of control charts for statistical batch process monitoring., Statistica Neerlandica 60 (3) (2006) 339–360.
[56] C. F. Alcala, S. J. Qin, Reconstruction-based contribution for process monitoring, Automatica 45 (7) (2009) 1593–1600.
[57] J. Camacho, Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models, Journal of Chemometrics 25 (11) (2011) 592–600.
[58] J. Camacho, A. Ferrer, Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects, Chemometrics and Intelligent Laboratory Systems 131 (2014) 37–50. doi:10.1016/j.chemolab.2013.12.003.
[59] R. Bro, A. K. Smilde, Centering and Scaling in Component Analysis., Journal of Chemometrics. 17 (2003) 16–33.
[60] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, T. Pepe, A novel PCA-based network anomaly detection, in: IEEE International Conference on Communications, 2011, pp. 1–5. doi:10.1109/icc.2011.5962595.
[61] W. Ku, R. H. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems 30 (1) (1995) 179–196. doi:10.1016/0169-7439(95)00076-3.
[62] J. Camacho, J. Picó, A. Ferrer, Bilinear modelling of batch processes. Part I: Theoretical discussion, Journal of Chemometrics 22 (5) (2008) 299–308. doi:10.1002/cem.1113.
[63] Vast challenge 2012, `http://www.vacommunity.org/VAST+Challenge+2012`, accessed: July 2015.
[64] J. Camacho, A. Pérez-Villegas, R. A. Rodríguez-Gómez, E. Jiménez-Manas, Multivariate exploratory data analysis (meda) toolbox for matlab, Chemometrics and Intelligent Laboratory Systems 143 (0) (2015) 49 – 57. doi:http://dx.doi.org/10.1016/j.chemolab.2015.02.016.
[65] E. Saccenti, J. Camacho, On the use of the observation-wise k-fold operation in pca cross-validation, Accepted in Journal of Chemometrics.
[66] Netflow iptables module. opensource project, `http://sourceforge.net/projects/ipt-netflow/files/ipt-netflow/`, accessed: June 2015.
[67] F. Arteaga, A. Ferrer, Dealing with missing data in MSPC: several methods, different interpretations, some examples, Journal of Chemometrics 16 (2002) 408–418.
[68] P. R. C. Nelson, P. A. Taylor, J. F. MacGregor, Missing data methods in PCA and PLS: score calculations with incomplete observations, Chemometrics and Intelligent Laboratory Systems 35 (1996) 45–65.
[69] F. Arteaga, A. Ferrer, Framework for regression-based missing data imputation methods in on-line MSPC, Journal of Chemometrics 19 (2005) 439–447.
[70] I. Van˜Mechelen, A. K. Smilde, A generic linked-mode decomposition model for data fusion, Chemometrics and Intelligent Laboratory Systems 104 (2010) 83–94.
[71] A. K. Smilde, J. A. Westerhuis, S. de Jong, A framework for sequential multiblock component methods, Journal of Chemometrics 17 (2003) 323–337.
[72] F. Lindgren, B. Hansen, W. Karcher, M. \protectSj\˜ostr\ om, L. Eriksson, Model Validation by Permutation Tests: Applications to Variable Selection, Journal of Chemometrics 10 (1996) 521–532.
[73] N. M. Faber, R. Rajkó, How to avoid over-fitting in multivariate calibration–The conventional validation approach and an alternative, Analytica Chimica Acta 595 (2007) 98–106.
[74] J. M. González-Martínez, A. Ferrer, J. A. Westerhuis, Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping, Chemometrics and Intelligent Laboratory Systems 105 (2011) 195–206.
[75] A. Kassidas, J. F. MacGregor, P. A. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal 44 (1998) 864–875.
[76] N. Nielsen, J. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometrics data analysis using correlation optimised warping, Journal of Chromatography 805 (1998) 17–35.

Figures

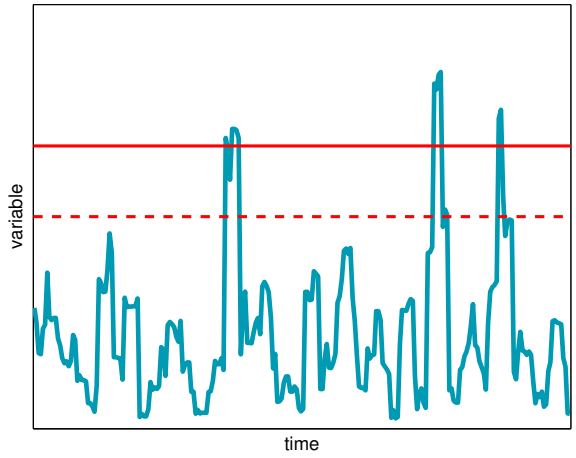
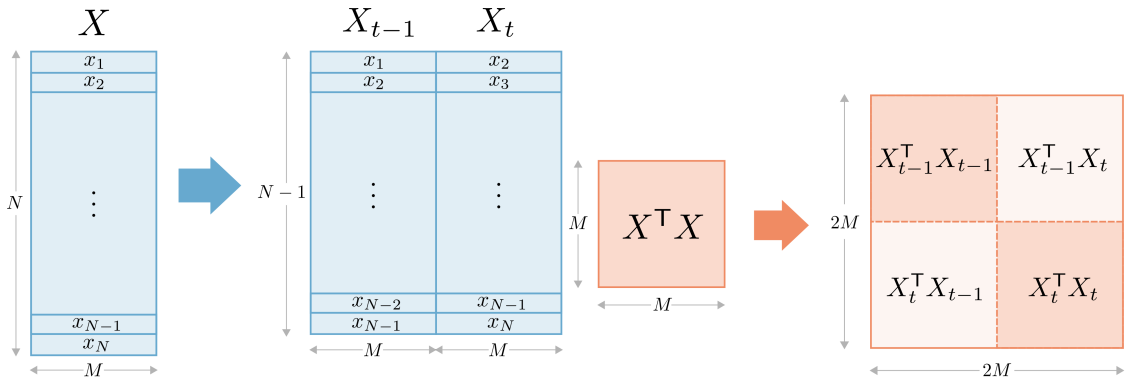
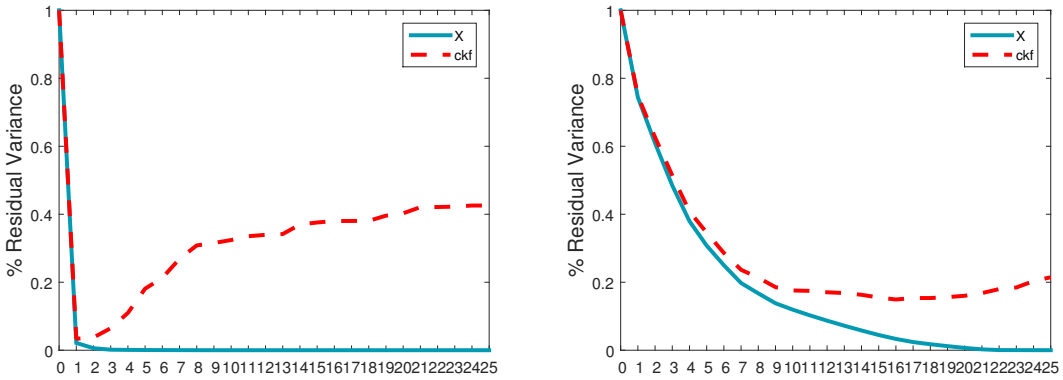


Figure 1: Sample control chart of a networking variable. Control limits are presented for a 95% (dashed line) and for a 99% (solid line) confidence levels.



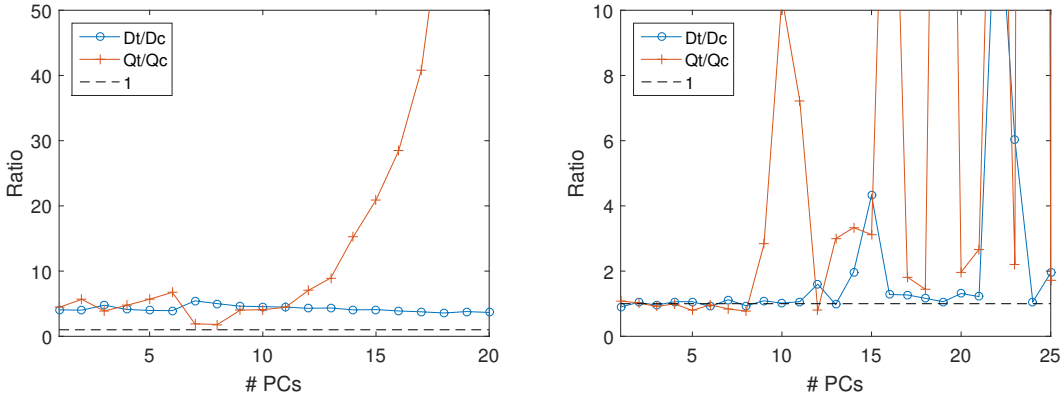
(a) PCA model incorporating one LMV (order 1). (b) Covariance matrices of the model.

Figure 2: Scheme of a PCA model incorporating dynamics.



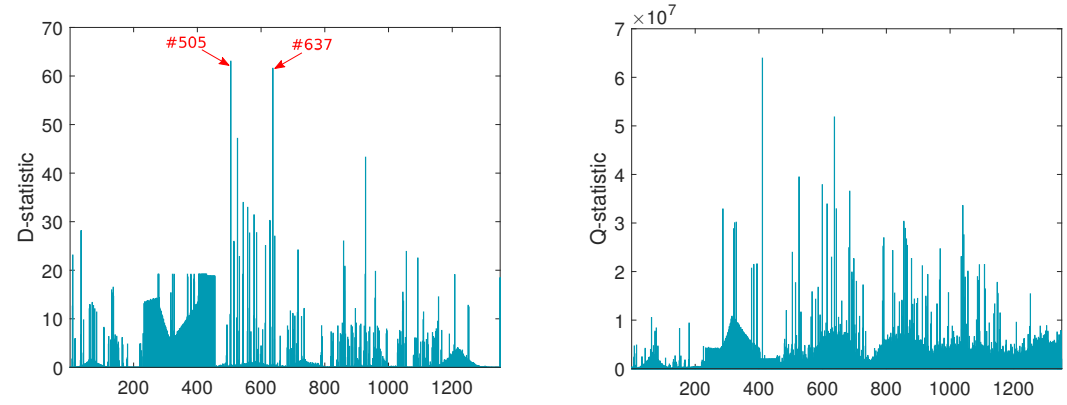
(a) Mean-centring (b) Auto-scaling

Figure 3: Percentage of residual variance (blue line) and column-wise k-fold (ckf) curve (red dashed line).

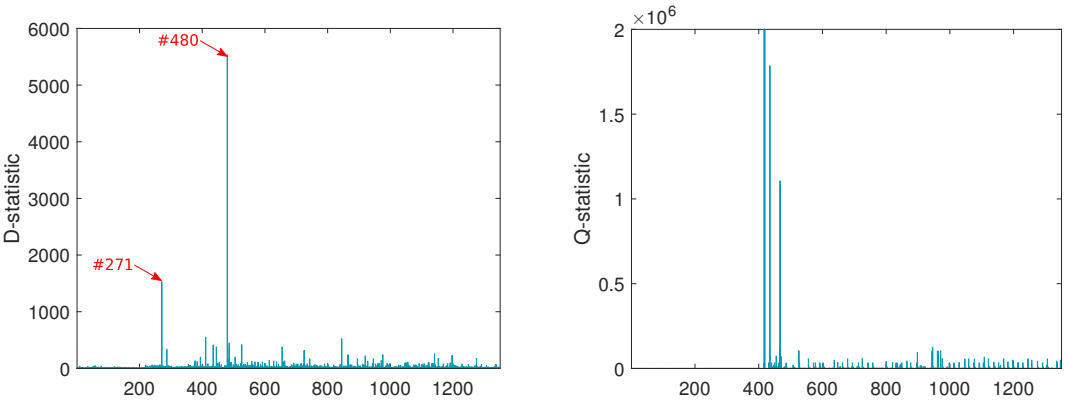


(a) Original data (b) Resampling

Figure 4: Ratio of statistics between calibration data and test data.

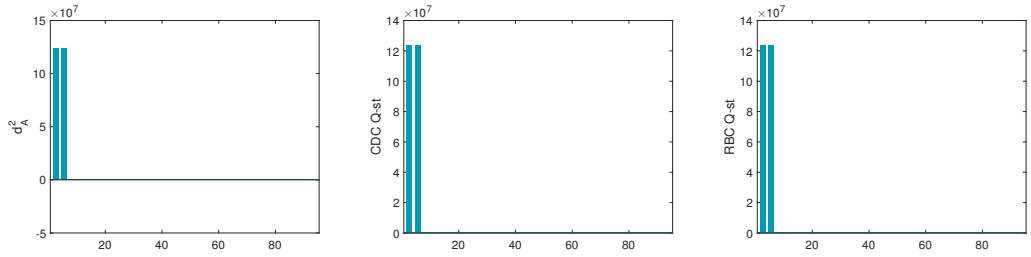


(a) 1 PC (mean-centring)

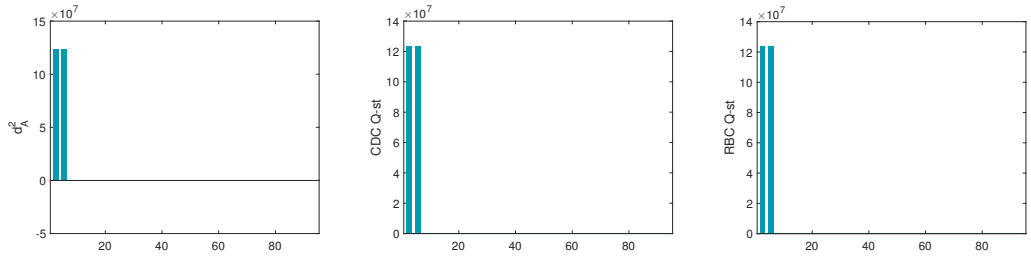


(b) 16 PCs (auto-scaling)

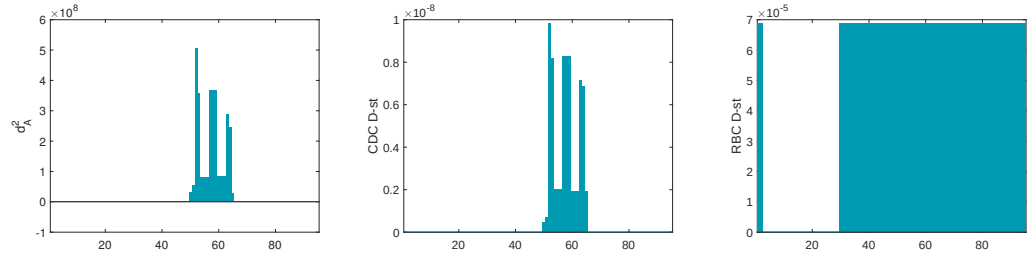
Figure 5: Multivariate statistical charts.



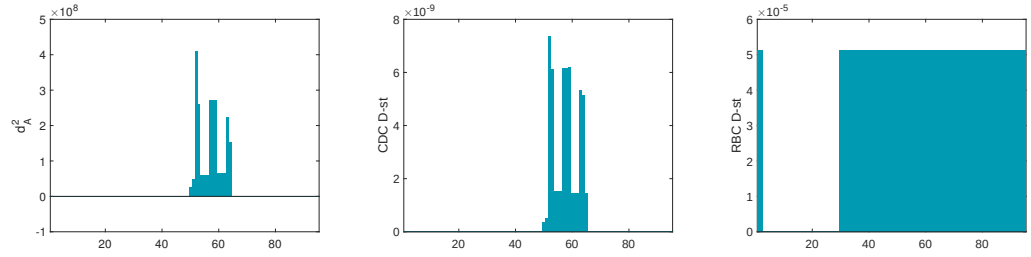
(a) #418, Residuals in 1 PC (auto-scaling)



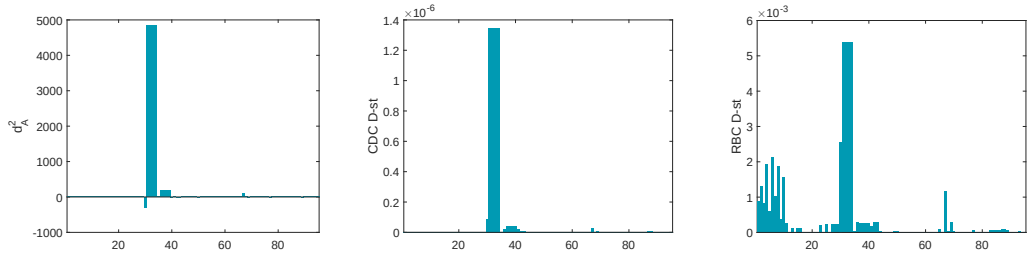
(b) #418, Residuals in 16 PCs (auto-scaling)



(c) #505, Model in 1 PC (mean-centring)

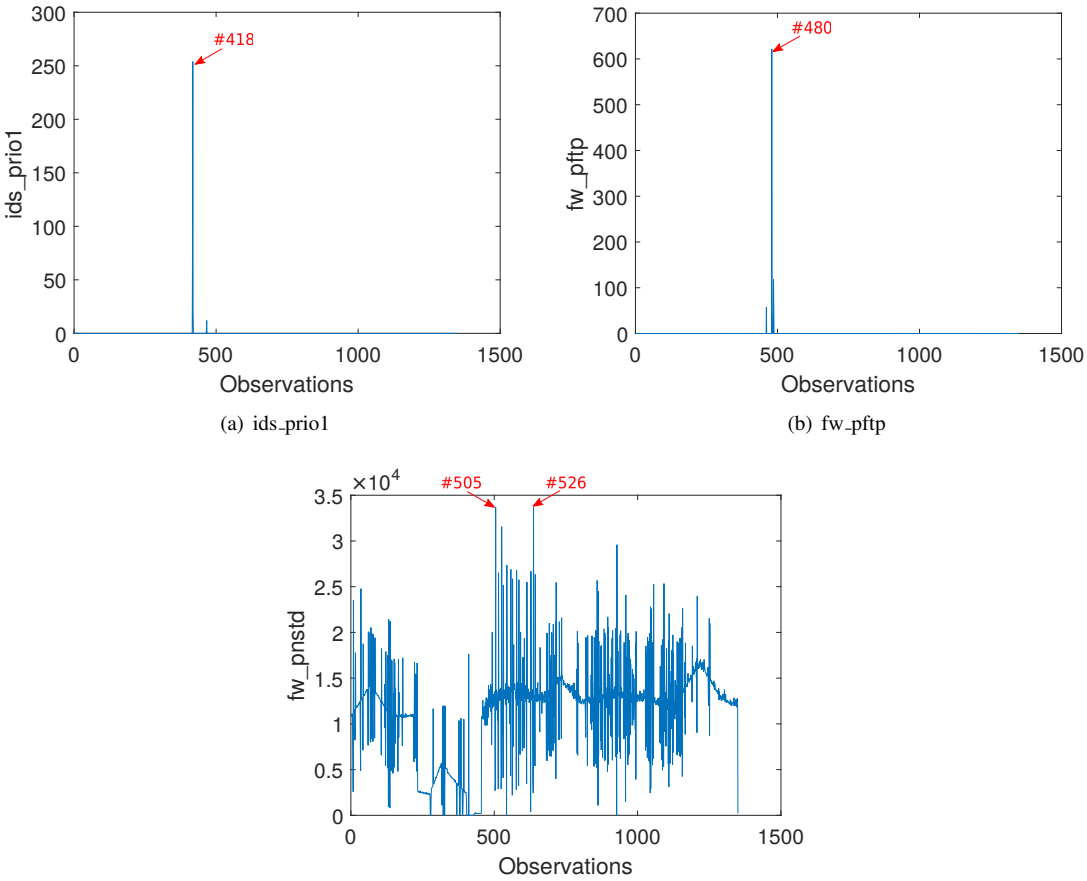


(d) #526, Model in 1 PC (mean-centring)



(e) #480, Model in 9 PCs (auto-scaling)

Figure 6: Diagnosis plots.



(a) ids_prio1

(b) fw_pftp

(c) fw_pnstd

Figure 7: Variables profile in time.

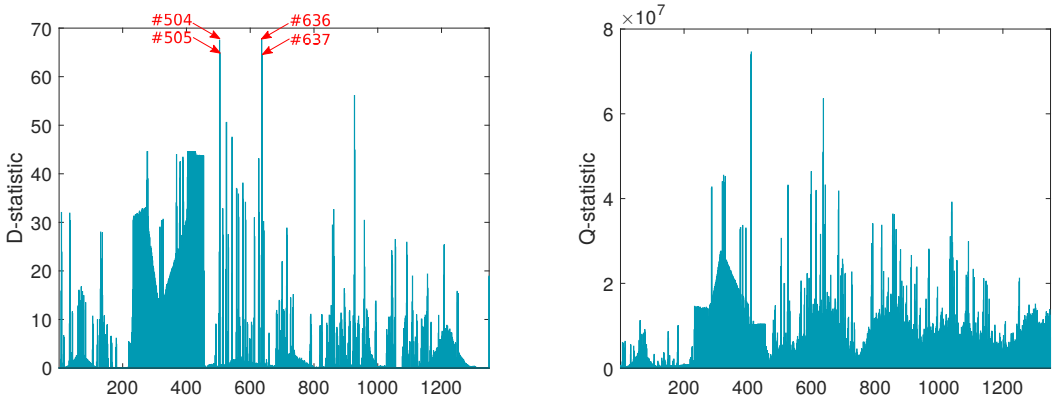


Figure 8: Multivariate statistical charts for the MSNM based on a PCA model with one LMV and 1 PC (mean-centring).

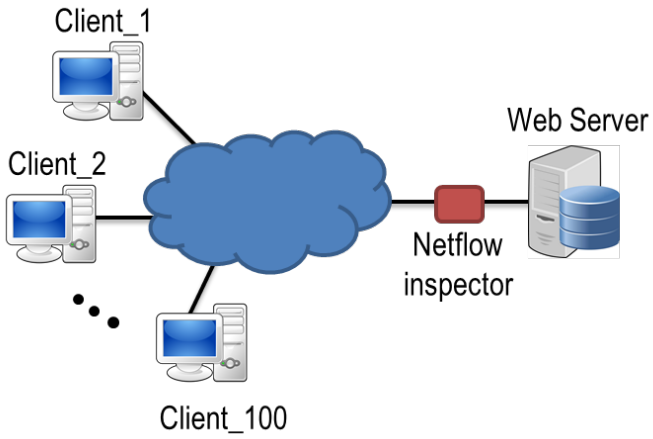
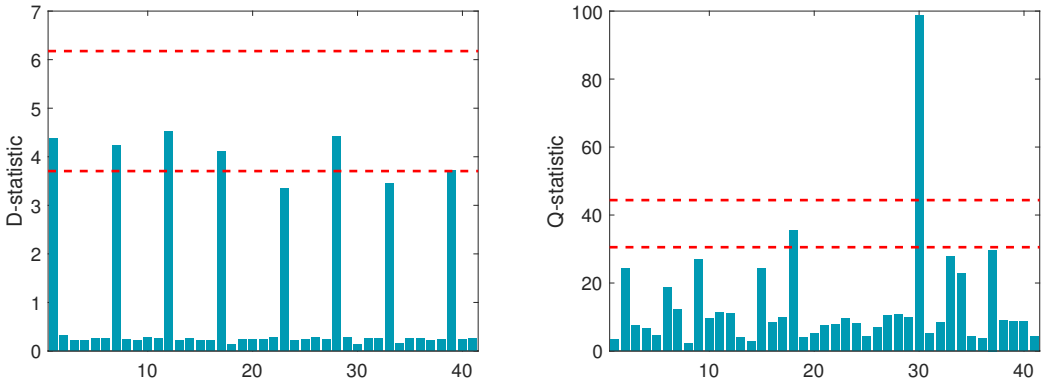
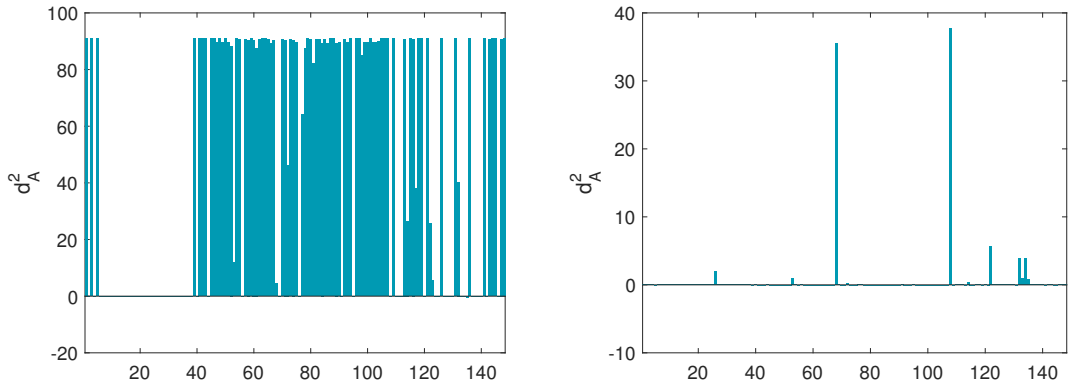


Figure 9: Topology of the controlled scenario.



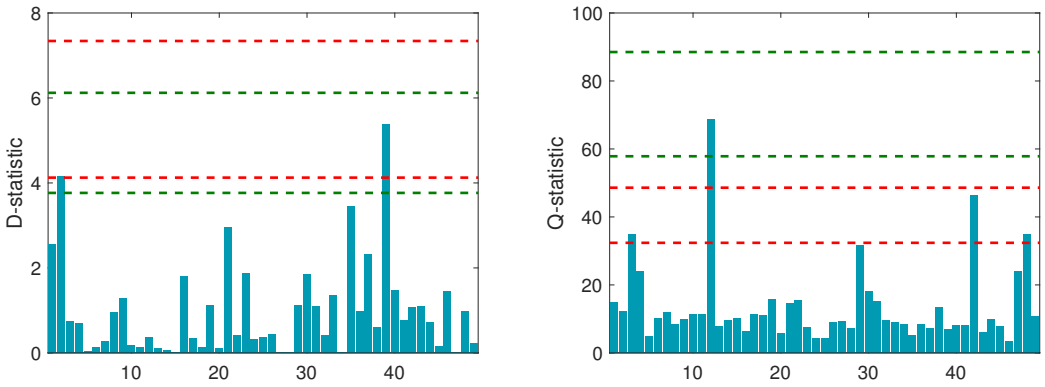
(a) D-statistic (b) Q-statistic

Figure 10: Control charts in phase I.



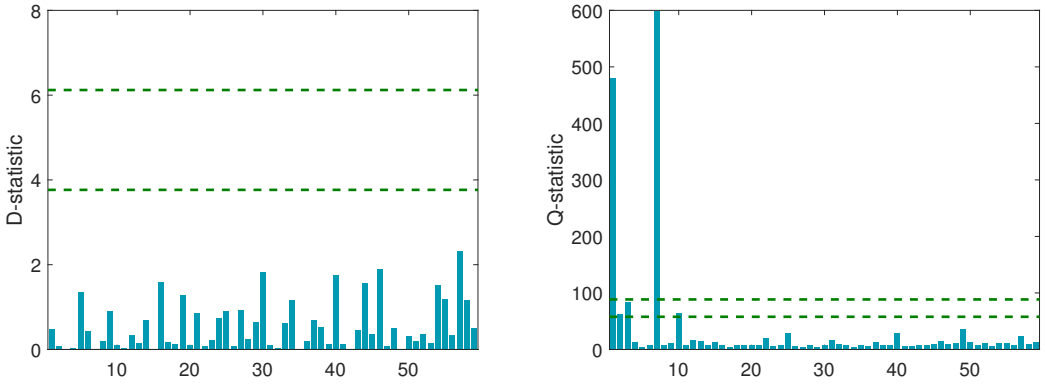
(a) *o*MEDA for the anomaly in Fig. 10(a) (b) *o*MEDA for the anomaly in Fig. 10(b)

Figure 11: *o*MEDA plots in phase I.



(a) D-statistic (b) Q-statistic

Figure 12: Control charts in phase II for calibration data.



(a) D-statistic (b) Q-statistic

Figure 13: Control charts in phase II for test data.

Tables

Table 1: Number of variables defined for every feature class in VAST 2012.

	Feature class	**#Variables**
Firewall log	Syslog priority	5
	Operation	6
	Message code	25
	Protocol	3
	IP address	9
	Port number	17
	Direction	2
	Conn. built/teardown	2
	Subtotal	**69**
IDS log	IP address	9
	Port number	17
	Alert class	5
	Priority	3
	Text label	9
	Subtotal	**43**

Table 2: Temporal intervals with highest D- and Q- statistics.

1 PC (mean-centring)		1 PC (auto-scaling)		9 PCs (auto-scaling)		16 PCs (auto-scaling)	
D-st	**Q**-st	**D**-st	**Q**-st	**D**-st	**Q**-st	**D**-st	**Q**-st
505	411	505	418	480	418	480	418
637	637	637	417	486	417	271	417
526	526	526	419	526	419	411	419
928	599	928	435	445	435	845	435

Table 3: Variables with the highest contribution in anomaly #418, residuals in 1 PC for auto-scaling.

Variable	Description	Variance	Source	Method
'ids_prio1'	IDS Priority 1 Alarms	0	IDS logs	O C R
'ids_pdns'	DNS port	0	IDS logs	O C R
'ids_privacy'	Privacy warning	0	IDS logs	O C R
'ids_ldns'	DNS	0	IDS logs	O C R

Table 4: Variables with the highest contribution in anomalies #505 and #526, model in 1 PC for mean-centering.

Variable	Description	Variance	Source	Method
'fw_phttp'	HTTP port	$4.8 \cdot 10^6$	FW logs	O C
'fw_sysinfo'	SYSINFO (Syslog priority 6)	$4.9 \cdot 10^6$	FW logs	O C
'fw_asa6'	ASA-6 messages	$5.7 \cdot 10^6$	FW logs	O C
'fw_tcp'	TCP connections	$5.7 \cdot 10^6$	FW logs	O C
'fw_ipws'	connections from/to WorkStations	$5.7 \cdot 10^6$	FW logs	O C
'fw_outbound'	outbounds connections	$5.6 \cdot 10^6$	FW logs	O C
'fw_pnstd'	ports>1024	$6.9 \cdot 10^6$	FW logs	O C

Table 5: Variables with the highest contribution in anomaly #480, model in 9 PCs for auto-scaling.

Variable	Description	Variance	Source	Method
'fw_pftp'	FTP port	79.5	FW logs	O C R
'fw_syswarn'	SYSWARN (Syslog priority 4)	79.5	FW logs	O C R
'fw_asa4'	ASA-4 messages	79.5	FW logs	O C R
'fw_asa41'	ASA-41 messages	79.5	FW logs	O C R

Table 6: Temporal intervals with highest statistics.

1 LMV and 1 PC (mean-centring)

D-st	**Q**-st
636	411
504	410
505	637
637	636

Table 7: Number of variables defined for every feature class in the controlled scenario.

	Feature class	#Variables
Netflow	IP address	4
	Port number	102
	Protocol	5
	TCP flags	6
	Type of service	3
	Number of packets	10
	Number of bytes	10
	Interface	8
	Total	**148**

Table 8: Ground-truth specification for the last experiment.

Relative time	Time-stamp	**Description**
00:00	[10:14:48]	Initialize 100 HTTP clients
00:20	[10:34:53]	High rate DoS for 5 min – spoofing within network segment
00:30	[10:44:49]	Data exfiltration 1
00:40	[10:54:53]	Low rate DoS for 5 min – spoofing within network segment
00:50	[11:04:53]	Low rate DoS for 5 min – spoofing to any IP address
01:00	[11:14:45]	Data exfiltration 2
01:05	[11:19:48]	Reduce to 50 HTTP clients
01:25	[11:39:48]	Increase to 100 HTTP clients
01:50	[12:04:48]	Stop HTTP clients
01:52	[12:06:48]	Stop experiment

Table 9: Results.

Approach	Preprocessing	#PCs	#LMVs	TP	TN	FP	FN	Recall	Specificity	**Accuracy**
Lakhina et *al.* [5, 39]	MC	2	0	11	73	0	29	0.28	1.00	**0.74**
Lakhina et *al.*	AS	1	0	22	50	23	18	0.55	0.68	**0.64**
Lakhina et *al.*	AS	4	0	37	71	2	3	0.93	0.97	**0.96**
Brauckhoof et *al.* [8]	MC	3	1	14	72	1	26	0.35	0.99	**0.76**
MSNM	MC	2	0	34	73	0	6	0.85	1.00	**0.95**
MSNM	AS	1	0	37	68	5	3	0.93	0.93	**0.93**
MSNM	AS	4	0	37	71	2	3	0.93	0.97	**0.96**