Semi-supervised Multivariate Statistical Network Monitoring for Learning Security Threats

José Camacho, Gabriel Maciá-Fernández, Noemí Marta Fuentes-García, and Edoardo Saccenti

***Abstract*—This paper presents a semi-supervised approach for intrusion detection. The method extends the unsupervised Multivariate Statistical Network Monitoring approach based on Principal Component Analysis by introducing a supervised optimization technique to learn the optimum scaling in the input data. It inherits the advantages of the unsupervised strategy, capable of uncovering new threats, with that of supervised strategies, able of learning the pattern of a targeted threat. The supervised learning is based on an extension of the gradient descent method based on Partial Least Squares (PLS). Moreover, we enhance this method by using sparse PLS variants. The practical application of the system is demonstrated on a recently published real case study, showing relevant improvements in detection performance and in the interpretation of the attacks.**

***Index Terms*—Multivariate Statistical Network Monitoring, Anomaly Detection, Intrusion Detection, Semi-supervised learning, Partial Least Squares regression, Principal Components Analysis.**

I. Introduction

Cibersecurity incidents are considered one of the most relevant threats for businesses in almost any market. According to the VERIZON annual Data Breach Investigation Report (DBIR) [1], tens of thousands of attacks targeted private and public companies during 2017. To effectively fight against this real menace, the security industry has identified that an essential line of defense should be based on the joint effort of all stakeholders combining their technical skills and information [2]. In this regard, the use of anomaly-based Intrusion Detection Systems (IDS) [3] is fundamental to unveil previously unknown attack strategies and thwart potential attacks to organizations.

Main technical challenges in the field of IDS design are the need for handling massive and disparate sources of information, the extraction of useful knowledge for the forensic analysis of incident data and the optimization of the detection system to targeted threats. Dealing with very different sources of information and a vast amount of data has fostered the use of machine learning and data mining techniques [4]. However, it becomes essential to utilize tools and approaches that provide interpretability of results, that is, information about the features and the real cause of an attack in order to efficiently respond to it. Regretfully, the bulk of machine learning and data mining approaches does not satisfy this requirement. Regarding the optimization to targeted threats, proper system update is a relevant feature, in particular to make the most of information sharing technologies that distribute warnings among corporations when new threats arise. The adaptation to targeted threats of machine learning models is more technically challenging than that of traditional rule-based systems (e.g. antivirus, rule-based IDS like snort, correlation engines, etc.), more extended in the industry.

Multivariate Analysis has been recognized as an outstanding approach for anomaly detection in several domains, including industrial monitoring [5] and networking [6]. In the field of industrial processing, a well developed strategy is Multivariate Statistical Process Control (MSPC). A main tool within MSPC is Principal Component Analysis (PCA), which was proposed by Lakhina et *al.* for intrusion detection in [7]. Some of the benefits of PCA are its unsupervised nature, that does not require any a-priori knowledge on the data, and its capability to provide diagnosis information for a given anomaly, a main advantage over other machine learning methodologies. This diagnosis support allows shortening the lag from detection to response in a security incident, and therefore has practical implications.

In previous work [8], we introduced a methodology named Multivariate Statistical Network Monitoring (MSNM), an extension of PCA-MSPC applied to the intrusion detection problem. MSNM overcomes some reported limitations of the original PCA approach of Lakhina et *al.* [9]. It is based on an over-parameterization of the feature space, that is, on defining a large number of data features within the detection system. This combines with the multivariate approach based on PCA, that can handle high-dimensional data with millions of variables. However, an open problem and main limitation in MSNM is how to select the relative relevance of the features in the system. Experiments in [8] showed high sensitivity of the detection to the relative relevance (scaling) of the features in PCA. We can make the most of this sensitivity to enhance the detection ability to a set of targeted threats. In particular, defining an optimum scaling of the features to identify the pattern of a recently identified threat is useful to update the monitoring system. This would enhance MSNM with an adaptation mechanism equivalent to what it is done in a traditional rule-based detection system when adding rules with the fingerprint of recently identified threats.

This paper presents a new procedure to optimize the scaling of the features in MSNM for the detection of targeted threats. This makes MSNM a semi-supervised learning approach, where the original unsupervised PCA model is enhanced to be optimum for the detection of specific anomalies. For this purpose, we employ and improve an optimization algorithm

J. Camacho, G. Maciá-Fernández and N. M. Fuentes-García are with the Department of Signal Theory, Telematics and Communications, School of Computer Science and Telecommunications - CITIC University of Granada, Granada, Spain.

E. Saccenti is with the Laboratory of Systems and Synthetic Biology, Wageningen University & Research, Wageningen, the Netherlands.

Manuscript received June 18, 2018; revised XX, 2018.

[10][11] originally introduced in the context of process optimization. This algorithm is based on Partial Least Squares (PLS) [12][13], a multivariate regression technique. We refer to this algorithm as run-to-run PLS (R2R-PLS). The R2R-PLS has been shown to outperform state-of-the-art optimization techniques, like genetic algorithms, in large search spaces [10].

The contributions of this paper are the following:

- We cast the problem of adapting an anomaly detection system to targeted threats into an optimization problem. In particular, we apply an optimization scheme in the context of anomaly detection with MSNM, overcoming one of its principal limitations: the sensitivity of detection to the features scaling. While this is specially interesting in the context of intrusion detection, this contribution also applies to different application domains [5].
- For this purpose, we extend the original R2R-PLS in [10], [11] to a general optimization problem, simplifying its implementation and improving its computational efficiency.
- We further extend the R2R-PLS optimization to sparse PLS variants, very popular in biological sciences [14], leading to an improvement of the optimization performance and in the understanding of the connection between the features scale and the MSNM detection.
- We demonstrate previous contributions in a recently published real case study [15].

The rest of the paper is organized as follows. Section 2 discusses related work to this paper. Section 3 introduces the MSNM technique. Section 4 reviews PLS and two sparse variants. Section 5 presents our particular implementation of the R2R-PLS algorithm. Its application in the adaptation of MSNM to new threats is presented in Section 6. Section 7 demonstrates the application of the approach to the real case study. Section 8 draws some concluding remarks.

II. Related Work

Machine learning (ML) techniques have been widely applied to cybersecurity problems. ML refers to the combination of statistics and artificial intelligence to learn a model from data [4][16][17]. This is a global term widely used to refer to the task in which one automatically calibrates (trains) a model or algorithm to obtain a descriptive output for a given input. If the value for the output is previously known and used in the training, then the learning is called *supervised* and it usually applies to classification and regression problems. However, if only the input data are known and the objective is to extract patterns or common behaviour from the data, the learning is known as *unsupervised* [16][18]. Mixed approaches are considered to be *semi-supervised* learning [18]. Support vector machines, neural networks or decision trees are common examples of supervised learning, thought extensions exist in the unsupervised setting. Factorization methods like PCA and clustering algorithms such as K-means are often applied for unsupervised analyses and often combined with supervised methods.

In cybersecurity, unsupervised ML methods are applied to the anomaly detection problem, *a.k.a.* intrusion detection problem, while supervised methods can be used to detect and classify previously observed attacks. In this context, the use of PCA was proposed more than a decade ago [19]. Due to its unsupervised nature, PCA does not require –and is not limited by– an a-priori specification of potential anomalies in the system. This means that PCA is still useful to detect new types of anomalies, something mandatory in real world anomaly detection. The most referred work for PCA anomaly detection is that of Lakhina et *al.* [7], form which alternative proposals have been developed [20][21][22][23]. One of them, the MSNM methodology [8], is the base of the approach of this paper. This methodology allows to combine traffic data with other security data sources, demonstrating high detection capabilities and with the advantage of providing diagnosis support [24].

The learning process in ML yields a model from a training data set. This learning process is also referred to as model calibration, and it is generally performed by optimizing the model parameters in consecutive steps until convergence [4][17]. The calibration of the parameters is performed following optimization strategies, which pursue to find, at least, local optimums for their values and, hopefully, global solutions [25]. These approaches can be classified as follows [16]: Stochastic approximation methods, expectation-maximization methods and greedy optimization. Within stochastic optimization, gradient descent methods apply the derivative of the optimization function to obtain the direction of search. In recent papers [10][11] a variation of this type of optimization based on Partial Least Squares (PLS) was presented. PLS is a particular form of regression model suitable to handle high dimensional data sets. For this reason, the PLS-based optimization is specially suited to solve optimization problems where the search space is of high dimensionality. Therefore, it is a practical choice in ML to optimize a large number of model parameters.

While PLS is well suited to model high dimensional data, a recent trend of research has explored the ability of a type of methods that perform PLS regression combined with variable selection. These are generally referred to as sparse PLS (SPLS) methods. Several variants of SPLS been proposed [14][26][27][28][29] with the goal of performing variable selection during model calibration, discarding non-informative variables. Results show that SPLS variants are more stable and often yield improved performance in very high dimensional set-ups.

This paper presents a semi-supervised approach for intrusion detection. The method extends the unsupervised MSNM approach by introducing a supervised optimization technique to learn the optimum scaling of the features. It inherits the advantages of the unsupervised strategy, capable of uncovering new threats, with that of supervised strategies, capable of learning the pattern of a given threat. Considering that the number of features of MSNM corresponds to the dimension of the search space in the optimization problem, and that this number can be very large, we apply the PLS-based optimization for the supervised learning. Furthermore, we extend this optimization technique using sparse variants of PLS, improving the learning ability and model interpretability.

III. Multivariate Statistical Network Monitoring

The MSNM follows 4 main steps: *1) Parsing*, *2) Fusion*, *3) Detection*, and *4) Diagnosis*. The first three steps are equivalent to what it is commonly done in other machine learning methodologies. However, step 4 is a main advantage in MSNM. This step is possible thanks to the white-box, exploratory characteristics of PCA as the core of the approach. PCA is a linear model and as such it is easy to interpret in terms of the connection between anomalies and features, something much more complicated in the non-linear machine learning variants.

A. Parsing

The information captured from a network is usually presented in the form of system logs or network traces, and cannot be directly used to feed a typical tool for anomaly detection. Therefore, some sort of parsing and feature engineering needs to be done in order to generate quantitative features that can be used for data modeling.

Lakhina et *al.* [7] proposed the definition of counters obtained from *Netflow* records as quantitative features for anomaly detection using PCA. In [30], we generalized this definition to consider several sources of data, proposing the feature-as-a-counter approach. Each feature contains the number of times a given event (*e.g.* number of packets sent from public IPs or number of flows associated to destination port 80) takes place during a given time window. The parsing transforms the raw data in a stream of features, where each time interval of *e.g.* 1 minute is represented by a feature vector of counts.

B. Fusion

The feature-as-a-counter definition simplifies the fusion of different data sources in a single set of features. For each different source of data, a set of features (counters) is defined. The sampling rate for each source may be different, due to the specific dynamics of the source. Thus, to combine the features from different sources these need to be stretched/compressed to a common sampling rate, yielding a unique matrix of data of high dimensionality. In practice, when possible, all sources are parsed at the same time rate, so that the fusion operation is done by simply appending the features associated to the different sources.

The combination of the feature-as-a-counter and the fusion procedure is specially suited for the subsequent multivariate analysis. It yields high dimensional feature vectors that need to be analyzed with dimension reduction techniques, like PCA. The diagnosis procedure benefits from the definition of a large number of features for a better description of the anomaly taking place. Furthermore, counters and their correlation are easy to interpret.

C. Detection

The core of MSNM is PCA. PCA is applied to data sets where M variables or features are measured on N observations with the aim of finding the subspace of maximum variance in the M-dimensional feature space. The original features are linearly transformed into the Principal Components (PCs), which are the eigenvectors of $\mathbf{XX} := \mathbf{X}^T \cdot \mathbf{X}$, typically for mean centred (MC) $\mathbf{X}$ and sometimes also after auto-scaling (AS, normalization to unit variance).

The PCA model follows the expression:

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

where A is the number of PCs, $\mathbf{T}_A$ 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.

For each observation, corresponding to a feature vector collected in a given time interval, the corresponding score vector 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 the 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.

For the detection of anomalies in MSNM, a pair of charts are monitored: the Q-statistic (Q-st), which compresses the residuals; and the D-statistic (D-st) or Hotelling's T2 statistic, computed from the scores. The D-st and the Q-st for an observation can be computed from the following equations:

$$D_n = \mathbf{t}_n \cdot (\Sigma_T)^{-1} \cdot \mathbf{t}_n^t \tag{4}$$

$$Q_n = \mathbf{e}_n \cdot \mathbf{e}_n^t \tag{5}$$

where Σ_T represents the covariance matrix of the scores in the calibration data.

With the statistics computed from the calibration data, upper control limits (UCL), *i.e.* detection thresholds, can be established in the charts at a certain confidence level [31] to decide if future events are anomalous. A straightforward approach to define the UCLs is by using percentiles over the statistics computed from the calibration data $\mathbf{X}$. Once the system is calibrated and control limits computed, it can be applied to incoming data/traffic. An anomaly is identified when either the D-st or the Q-st exceeds the pre-defined UCLs.

D. Diagnosis

Once an anomaly is detected, a diagnosis step is performed to identify the features associated with it. This information is very useful to identify and, eventually, troubleshoot the possible root causes of the anomaly. The contribution of the features to a given anomaly can be investigated with the contribution plots or similar tools, like oMEDA [32]. Thus, anomalies are detected in the D-st and/or Q-st statistics, and then the diagnosis is performed with e.g. oMEDA. The output of oMEDA is a $1 \times M$ vector where each element contains the contribution of the corresponding feature to the anomaly under study. Those contributions with large magnitude, either positive or negative, are considered to be relevant.

While the diagnosis capabilities of MSNM are a main advantage over other ML techniques, these capabilities are not the focus of this paper. The interested reader is referred to [24].

IV. Partial Least Squares Regression and Sparse variants

This section introduces the regression techniques that we will employ within the optimization of the scaling parameters of MSNM. The regression models are used to estimate the gradient in the optimization. Since the scaling parameters, and so the gradient, are high dimensional, we need regression techniques that are suitable for the analysis of high dimensional data sets. This is the case of PLS and sparse variants.

A. Partial Least Squares

PLS is a particular form of regression that is suitable in presence of collinearity in the predictors, common in high dimensional problems. Collinearity makes the classical multivariate least-squares (LS) regression break down due to singularity of the covariance matrix. PLS extends LS by inheriting the philosophy of PCA.

Briefly, given a $N \times O$ response matrix $\mathbf{Y}$ and a $N \times M$ set $\mathbf{X}$ of predictors variables, the PLS algorithm defines a subspace of $\mathbf{X}$ which maximizes its covariance with $\mathbf{Y}$. The PLS model can written as:

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{P}^T + \mathbf{E} \tag{6}$$

$$\mathbf{Y} = \mathbf{T} \cdot \mathbf{Q}^T + \mathbf{F} \tag{7}$$

where $\mathbf{T}$ is the $N \times A$ score matrix, A the number of latent variables (LVs) to be used to fit the model, $\mathbf{P}$ and $\mathbf{Q}$ are the $M \times A$ and $O \times A$ loading matrices for the predictors and the response, respectively, and $\mathbf{E}$ and $\mathbf{F}$ are the $N \times M$ and $N \times O$ residual matrices of $\mathbf{X}$ and $\mathbf{Y}$. Setting

$$\hat{\mathbf{B}}_{PLS} = \mathbf{W} \cdot (\mathbf{P}^T \cdot \mathbf{W})^{-1} \cdot \mathbf{Q}^T \tag{8}$$

with $\mathbf{W}$ the $M \times A$ matrix of weights, the two models given by Equations (6) and (7) can be re-arranged in a single regression model given by

$$\mathbf{Y} = \mathbf{X} \cdot \hat{\mathbf{B}}_{PLS} + \mathbf{F} \tag{9}$$

The PLS model can be obtained, among others, using the NIPALS algorithm, and the optimal number of LVs A can be estimated by cross-validation.

B. Sparse PLS

Although PLS can effectively handle noisy predictor variables, the inclusion of variables which are non-relevant for the prediction of the response usually decreases the prediction performance of the model. Several variants of the PLS algorithm have been proposed [14][26][27][28][29] with the goal of performing variable selection during model calibration. This family of algorithms is termed Sparse PLS (SPLS), and are often reported to show improved performance over PLS in high dimensional data. The most extended approach to define sparse models is based on the LASSO regularization [33], often applied using a soft-thresholding operation [34].

One popular SPLS algorithm is the variant proposed by Lê Cao *et al.* [35] where the (sparse) PLS problem is solved using singular value decomposition [36]. Briefly, given the response ($\mathbf{Y}$) and predictor ($\mathbf{X}$) matrices, the R-rank matrix

$$\mathbf{C} = \mathbf{X}^{\mathrm{T}}\mathbf{Y} \tag{10}$$

can be decomposed as

$$\mathbf{C} = \mathbf{G}\mathbf{D}\mathbf{U}^{\mathrm{T}} \tag{11}$$

where the matrices $\mathbf{G}$ ($N \times R$) and $\mathbf{U}$ ($L \times R$) are orthonormal and $\mathbf{D}$ is $R \times R$ diagonal containing the singular values of $\mathbf{C}$. Using this formulation, the loading vectors $\mathbf{p}^r$ and $\mathbf{q}^r$ for $\mathbf{X}$ and $\mathbf{Y}$ are the singular vectors $\mathbf{g}^r$ and $\mathbf{u}^r$ of $\mathbf{G}$ and $\mathbf{U}$, respectively.

A sparse PLS solution can be obtained by penalizing (*i.e.* forcing to 0) the loadings [37], which are a measure of the relative importance of each variable to the PLS model, by solving the optimization problem:

$$\arg\min_{\mathbf{p},\mathbf{q}} \|\mathbf{C} - \mathbf{p}\mathbf{q}^{\mathrm{T}}\|_F^2 + \lambda_1\|\mathbf{p}\|_1 + \lambda_2\|\mathbf{q}\|_1 \tag{12}$$

whose solution is given by the soft-thresholding $g_\lambda(x) = \mathrm{sign}(x)\left(|x| - \lambda\right)_+$ applied to the standard PLS solution. The two parameters λ_1 and λ_2 control the sparsity for $\mathbf{X}$ and $\mathbf{Y}$, respectively. However, a more practical and equivalent alternative is to select the number of non zero components of the loadings $\mathcal{N}_x$ and $\mathcal{N}_y$ [35] in the soft-thresholding [34]. In the context of this paper we restrict ourselves to set $\mathcal{N}_x$, so that only loadings in the x-block are sparse, since the response is univariate. Optimum values of this parameter can be found by cross-validation.

C. Group-wise PLS

A different approach to obtain sparse PLS solutions is the recently proposed Group-wise PLS (GPLS) [38], where the solution is found by defining groups of correlated predictor variables rather than using regularization.

Briefly, the GPLS algorithm starts by defining a set of K (possibly overlapping) groups of correlated variables that are obtained from a $M \times M$ correlation map $\mathbf{M}$ computed from $\mathbf{C} = \mathbf{X}^{\mathrm{T}}\mathbf{Y}$. Subsequently, the weights and scores of K PLS models of 1 LV, each of them considering only the set of variables corresponding to one of the groups, are computed. From these, the one with the largest correlation with $\mathbf{Y}$ is retained while the others are discarded. This is repeated for a number of LVs.

The GPLS approach is particularly suited for data exploration, but when data is sparse in a group-wise fashion (i.e. when there are groups of correlated variables related to the response) the algorithm outperforms PLS and SPLS in terms of goodness of prediction.

The fitting of a GPLS model requires the definition of a threshold, γ, to identify the groups of variables in $\mathbf{M}$, controlling simultaneously the number and size of the groups of variables to be used. Optimal values for γ can be chosen by graphically inspecting the correlation map $\mathbf{M}$, by controlling

the trade-off between group size and dimension or by using cross-validation.

V. Run-to-run xPLS optimization

The optimization of the scaling parameters of MSNM is defined as a gradient descent algorithm, where the gradient is estimated with the PLS-based methods discussed in the previous section. This optimization approach has been shown to outperform state-of-the-art optimization techniques, in particular genetic algorithms, in high dimensional search spaces [10].

Let us define $\mathbf{u}$ as a set of inputs we can vary as desired, typically within some specific bounds or constraints, to a system we would like to optimize. Let us also define $\mathbf{y}$ as the set of outputs we would like to optimize (either maximize or minimize) by setting appropriate values to $\mathbf{u}$. The goal of the optimization algorithm is to find those values in the input that optimize the output, see Fig.1. Without loss of generality, in the following we will assume we desire to maximize the values in $\mathbf{y}$ by properly setting $\mathbf{u}$, which is contrary to common optimization literature but more appropriate for our specific case.

The run to run (R2R) optimization algorithm [10], extended for a general optimization problem and for PLS, SPLS and GPLS, can be summarized as follows:

0. Select user defined parameters K (number of individual solutions in each iteration) and r_c (level of exploration). Initialize input solution candidate $\mathbf{u}_i$ for $i = 0$.
1. Repeat for $k = \{1...K\}$
   1.1. Generate random variant solution $\tilde{\mathbf{u}}_i^k = \mathbf{u}_i + r_c \cdot \mathbf{r}_i^k$, for $\mathbf{r}_i^k$ drawn from a multinormal random distribution.
   1.2. Apply input $\tilde{\mathbf{u}}_i^k$ to the system in Fig.1 and measure output $\tilde{\mathbf{y}}_i^k$.
2. Fit a xPLS model with the K instances of the inputs $\tilde{\mathbf{u}}_i^k$ arranged in the rows of $\tilde{\mathbf{U}}_i$, and outputs $\tilde{\mathbf{y}}_i^k$ arranged in rows of $\tilde{\mathbf{Y}}_i$:

$$\tilde{\mathbf{Y}}_i = \tilde{\mathbf{U}}_i \cdot \hat{\mathbf{B}}_i + \mathbf{F} \tag{13}$$

3. Compute the next input solution candidate[a] as: $\mathbf{u}_{i+1} = \mathbf{u}_i + 3 \cdot \hat{\mathbf{B}}_i$
4. Check for convergence in the solution and otherwise loop back to Step 1.

In each iteration, the xPLS model captures the variability around $\mathbf{u}_i$ related to the response. This allows the estimation of the gradient in the optimization. A random signal is added to the input to ensure that there is enough trade-off between exploration and exploitation.

We have intentionally overlooked the problem of meta-parameter estimation within the optimization. As already discussed, we need to define the number of LVs in PLS and there is one additional meta-parameter in each of the two sparse variants. Unfortunately, the use of cross-validation or other automatic means for meta-parameter selection is computational intensive, but this problem can be overcome. Within a gradient based optimization, we can simplify meta-parameter selection for the sake of computational efficiency. This is done by fixing meta-parameters during the optimization. To minimize a detrimental effect on performance, we set PLS, SPLS or GPLS to be very parsimonious, with the intuition that if the model was parsimonious in excess, this would be overcome by performing more iterations in the optimization. First, we suggest to use one single LV in step 2., since any further contribution of additional LVs can be done in future iterations. Regarding parameters $\mathcal{N}_x$ in SPLS and γ in GPLS, we suggest to set them so that models are very sparse, since again any missing variability in one iteration can be taken into account in future iterations.

[a]For minimization, the second addend is subtracted to the current solution.

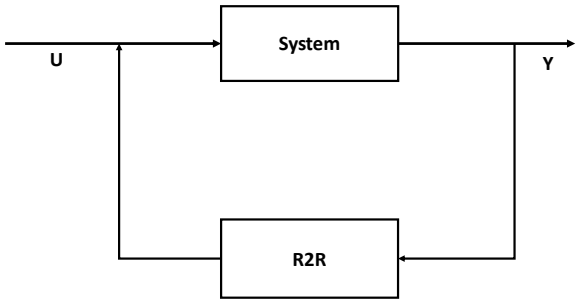

Fig. 1: Optimization scheme.

VI. Semi-supervised Multivariate Statistical Network Monitoring

This section particularizes the R2R optimization algorithm to our specific application: the optimization of the scaling parameters in MSNM. Here, the input $\mathbf{u}$ we would like to set is the scaling of the features in MSNM so as to maximize its detection performance, which is the output $\mathbf{y}$. In the following we discuss in detail inputs and outputs and the complete system.

A. Output of the optimizer

There are several possibilities to measure the detection performance of an anomaly detection system like MSNM. Here we will use the Area Under the ROC Curve (AUROC or AUC) computed from a labelled data set, where observations are labelled as normal or attacks. The receiver operating characteristic (ROC) curve shows the evolution of the true positive rate (TPR) versus the false positive rate (FPR) for different values of the classifying threshold, discussed below. The TPR is the percentage of true attacks that are identified by the MSNM system, while the FPR is the percentage of normal observations identified as attacks. The AUC is a scalar that quantifies the quality of the anomaly detector. An anomaly detector should present an AUC as close to 1 as possible, while an AUC around 0.5 corresponds to a random classifier.

The ROC curves for MSNM are obtained by varying a threshold in a specific combination of the Q-st and the D-st:

$$MSNM_n = \frac{A \cdot D_n}{M \cdot UCL_D} + \frac{(M - A) \cdot Q_n}{M \cdot UCL_Q} \tag{14}$$

where $MSNM_n$ is the output of the anomaly detector at a given observation n, D_n and Q_n the corresponding statistics and UCL_D and UCL_Q the corresponding 99% Upper Control Limits (UCL), computed as percentiles in the calibration data. Recall that A is the number of components in PCA and M the number of features in the data.

B. Input of the optimizer

To optimize the AUC, we modify the values of the scaling of the features. These features were computed in the parsing step of MSNM. The scale of the features changes their relative importance in the PCA model. Since PCA maximizes variance, the higher the relative scale (weight) of a variable, the more percentage of its information is captured in the scores and the less in the residuals of the model in Equation (1). However, understanding how this scaling impacts the detection ability of a MSNM system for a given attack is a real challenge, due to the non-linear nature of the detection statistics.

Given a vector $\mathbf{x}_m$ of size $N \times 1$, corresponding to a column of $\mathbf{X}$ with the values of one given feature (counter) in the N calibration observations, which we assume to be mean centred, its scaled version is given by

$$\tilde{\mathbf{x}_i} = \mathbf{x}_i w_i \tag{15}$$

Here we propose to define a set of values of $w_1, w_2, \ldots, w_M$ to be chosen such that when applied to the calibration data $\mathbf{X}$, detection by MSNM will be optimal in terms of AUC for one type or a set of types of attacks identified in a labelled data set.

C. Optimization procedure

The complete semisupervised system is depicted in Fig. 2. A detailed description of the optimization procedure follows:

1) Row vector $\mathbf{w}_0$, $1 \times M$, is initialized such as $\mathbf{w}_0 = (1, 1, \ldots, 1)_M$. The number of individual solutions in each iteration, $K \gg 1$, and the level of exploration r_c are chosen.
2) Row vector $\mathbf{w}_0^1$, $1 \times M$, is generated such as $\mathbf{w}_0^1 = \mathbf{w}_0 + r_c \cdot \mathbf{r}_0^1$ where $\mathbf{r}_0^1$ is a $1 \times M$ random vector whose entries are $\approx N(0, 1)$.
3) The weighting vector $\mathbf{w}_0^1$ is applied to the calibration data $\mathbf{X}$ and the detection performance of the resulting MSNM system AUC_0^1 is recorded.
4) Step 1 and 2 are repeated K times: resulting vectors $\mathbf{w}_0^1, \mathbf{w}_0^2, \ldots, \mathbf{w}_0^K$ are arranged in a $K \times M$ matrix $\mathbf{S}_0$ and the corresponding system performances in a $K \times 1$ vector $\mathbf{y}_0 = (AUC_0^1, AUC_0^1, \ldots AUC_0^K)^t$. The k-th row of $\mathbf{S}$ contains the k-th vector of weights $\mathbf{w}_0^k$ and $y_k = AUC_0^k$.
5) A xPLS model is fitted regressing $\mathbf{y}_0$ on $\mathbf{S}_0$ obtaining a set of regression coefficient $\hat{\mathbf{B}}_0$.
6) The current solution is updated: $\mathbf{w}_1 = \mathbf{w}_0 + 3 \cdot \hat{\mathbf{B}}_0$.
7) We check for convergence in $\mathbf{w}$, and otherwise loop to step 2.

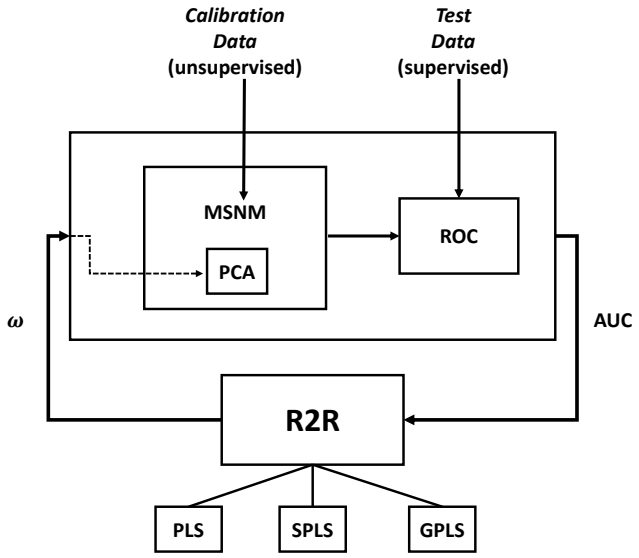

Fig. 2: Semi-supervised MSNM Optimization scheme.

TABLE I: Features of the calibration and the test sets in the UGR'16 dataset.

Feature	Calibration	Test
Capture start	10:47h 03/18/2016	13:38h 07/27/2016
Capture end	18:27h 06/26/2016	09:27h 08/29/2016
Attacks start	N/A	00:00h 07/28/2016
Attacks end	N/A	12:00h 08/09/2016
Number of files	17	6
Size (compressed)	181GB	55GB
# Connections	$\approx$ 13,000M	$\approx$ 3,900M

VII. Case Study: UGR'16 Data Set

A. Experimental Framework

We evaluate our approach to optimize variables scaling in a real scenario. The dataset considered is the publicly available UGR'16 dataset [15]. These data consist of Netflow network traces taken from a real Tier 3 ISP network composed of virtualized and hosted services of many companies and clients. Netflow sensors were located in the border routers of the network, capturing all the incoming and outgoing traffic from the ISP. All the details related to the dataset are summarized in Table I and can be consulted in [15]. Two blocks of data are provided, one for training models (*calibration set*), and the other for testing the results obtained from those models (*test set*).

The UGR'16 dataset is especially interesting for our experiments because the collected traffic includes controlled attack traffic against fake victims generated by 25 virtual machines that were deployed within the network. Thus, our aim is to test if our optimization algorithm is able to capture the relevant variables for every attack type and properly scale them to achieve good detection results. Although the variety of attacks in this dataset is limited, this is the only recent dataset (see

TABLE II: Variable values considered as features in our detection system.

Variable	#features → values
Source IP	2 → *public, private*
Destination IP	2 → *public, private*
Source port	50 → *specific services, Other*
Destination port	50 → *specific services, Other*
Protocol	5 → *TCP, UDP, ICMP, IGMP, Other*
Flags	6 → *A, S, F, R, P, U*
ToS	3 → *0, 192, Other*
# Packets in	5 → *very low, low, medium, high, very high*
# Packets out	5 → *very low, low, medium, high, very high*
# Bytes in	5 → *very low, low, medium, high, very high*
# Bytes out	5 → *very low, low, medium, high, very high*

survey of datasets in [15]) that includes real background traffic for a considerable amount of time (4 months), which is an essential requisite to properly evaluate the false positive rate in our detection results.

The attack traffic was performed during the test set collection, in particular during its first 12,000 observations, and it presents these different patterns:

- Low-rate DoS *(dos)*: TCP SYN attack during 3 minutes by using the tool *hping3*. There are three different variants, where one-to-one or many-to-one are combined with different schedulings.
- Port scanning *(scan11)*: Continuous one-to-one scanning from an attacker to a single victim's IP during 3 minutes by using the `nmap` tool.
- Port scanning *(scan44)*: Continuous scanning from 4 different attacker machines to four victim's IP in parallel during 3 minutes by using the `nmap` tool.
- Botnet traffic *(nerisbotnet)*: The test set includes botnet traffic traces obtained from the execution of the malware known as *Neris*, corresponding to the capture *CTU-Malware-Capture-Botnet-42* available in [39]. In this malware version, infected bots send SPAM, connect to an HTTP C&C server and use HTTP to perform some ClickFraud.

For the processing of the dataset, we consider time intervals of one minute. All the flows during an interval are aggregated and summarized into a M-dimensional vector, which corresponds to an observation. In particular, we define a set of M=138 network-related features, corresponding to 11 different Netflow variables, as shown in Table II.

Calibration data was cleaned of outliers following the Phase 1 approach in [8], so we expect it to be mainly composed of normal observations. A main real threat identified and cleaned from the calibration and test data is a SPAM campaign driven from some of the virtual machines located in the ISP. The cleaned calibration set is used to calibrate the MSNM system. The first 12,000 observations in the cleaned test data were split in two independent parts with 6,000 observations. The first set is used within the optimization to select the optimum scaling. The second set is used to validate the results. We also used the unclean second test data set, including the SPAM traffic, to assess the performance of the methods with previously unseen attacks.

B. Results

To assess the performance of the semi-supervised MSNM approach, we compare it with the unsupervised MSNM and the Support Vector Machine (SVM) technique, bluethe latter being a representative of a supervised approach. Two variants of the unsupervised MSNM for two preprocessing schemes, Mean-Centering (*MSNM-MC*) and Auto-Scaling (*MSNM-AS*), are considered. Two variants of the SVM are also considered, one based on the linear kernel (*SVM-L*) and one on the radial basis function kernel (*SVM-RBF*). To calibrate the anomaly detector, unsupervised techniques only make use of the (cleaned) calibration data, described in the previous section. Supervised techniques employ only the (cleaned) first test data set, which includes both normal traffic and all the types of artificial attacks. SVM metaparameters are selected according to recommendations in the Matlab documentation of function 'fitcsvm'. The semi-supervised MSNM is initially set to the *MSNM-MC*, and then optimized with the R2R algorithm using the (cleaned) first test data set. In all optimization runs, K is set to 100 and r_c is set to 0.01. Methods are compared in terms of the AUC computed for the (cleaned) second test set, and thus from independent data to that used in the calibration of the anomaly detectors. We derive confidence intervals in the performance using resampling techniques without replacement.

Fig. 3 shows the AUCs of the different detection approaches for the four artificial attack patterns. For the calibration, no distinction is made on the type of attack, that is, the anomaly detectors are calibrated using two types of labels: normal and attack. For the evaluation, each group of AUCs is computed comparing normal data with the specific type of attack, leaving out the observations corresponding to the other attack patterns. We expect supervised approaches to outperform the others in this experiment, since all the attacks under evaluation were used in their calibration. As expected, unsupervised methods generally yield the worst results, in particular the *MSNM-MC* approach. The *MSNM-AS* is generally outperformed by semi-supervised techniques. The improvement is more notable in the case of the *nerisbotnet* pattern. It is remarkable that this is generated by a botnet that is equipped with mechanisms to hide its traffic and, thus, the detection of this traffic following unsupervised methodologies is a real challenge. In general we can conclude that the R2R optimization is improving the performance of the MSNM, but this improvement is limited when diverse attacks patterns are optimized at the same time, since different patterns may counteract in the optimization. Supervised approaches based on SVM yield very good results. In particular, the *SVM-RBF* shows a similar performance to semi-supervised MSNM, and the linear kernel is among the best anomaly detectors for all the anomalies and clearly outperforms the other methods in '*nerisbotnet*'.

One typical advantage that is claimed for unsupervised (and most semi-supervised) methods in comparison to supervised approaches is that they are capable of identifying new threats. This is because they are based on a model of normality, and any new threat that does not follow that model can be detected. To check whether the semi-supervised MSNM retains this capability after the scaling optimization, we compared its

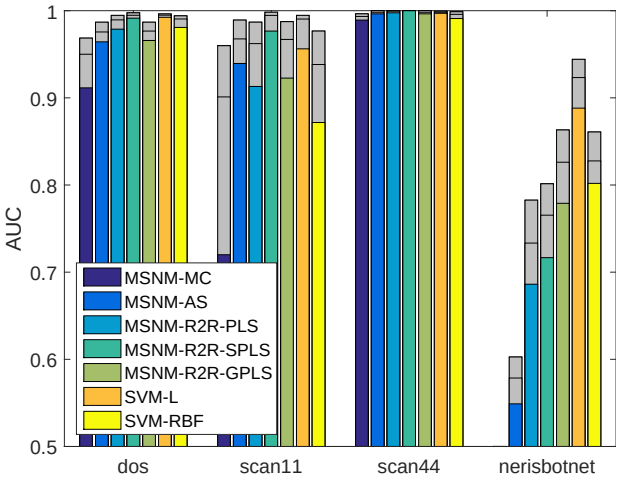

Fig. 3: AUC values per artificial attack type. Supervised and semi-supervised approaches calibrated without distinguishing attack types. Quartiles (25% and 75%) and median are shown in the bars.

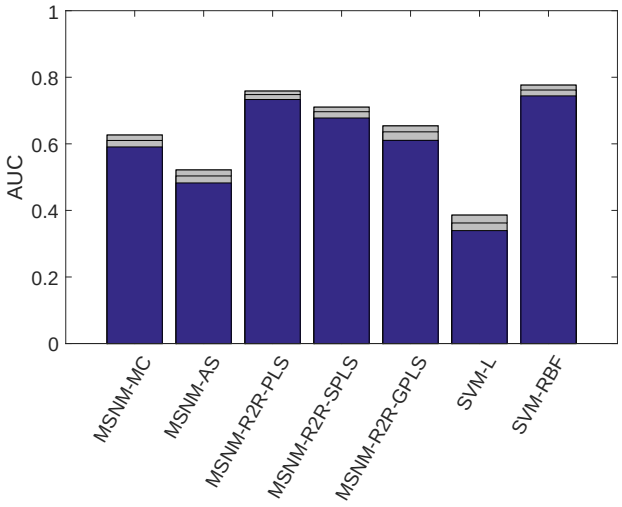

Fig. 4: AUC values for the real SPAM traffic. Quartiles (25% and 75%) and median are shown in the bars.

performance with the other methods in the detection of the real SPAM traffic in the (unclean) second test data set. The anomaly detectors are the same as those used before, and therefore they were calibrated using independent and cleaned data, in absence of any SPAM trace. Therefore, supervised techniques are not expected to outperform the others in this experiment. Results are shown in Fig. 4. We can see that the semi-supervised methods provide a good performance in comparison to the rest, and even outperform the unsupervised MSNM. This cannot be understood as a general result. In principle, all MSNM variants have the same *apriori* probability to detect new threats, since this depends very much on the relationship between the new threat and the normality model. As the new threat can be anything, this relationship will likely vary from case to case. However, this experiment shows that the optimization did not have a negative effect on the ability of MSNM to detect the unseen threat. Regarding supervised approaches, the *SVM-L* yielded a very poor result on the new attack type, as expected for a supervised technique, but the *SVM-RBF* showed the best result among the methods. blueThis result can be explained due to the special properties of *SVM-RBF*. This method is supervised since during calibration, the classifier is optimized to distinguish between the two classes of data. However, the model identifies a support that contains one of the classes, and the rest of the feature space is assigned to the other class. If the former is the class for normal observations, and the latter models the attacks, the configuration of *SVM-RBF* mimics that in MSNM, or in general of an anomaly detector, and is indeed very useful to detect new anomalies. Note, however, that if the labels for normal and anomalous classes are switched, the *SVM-RBF* provides the worst performance, and that, according to our previous discussion, this specific result is very dependent on the relationship between the new threat and the normality model.

Another experiment we can conduct is to calibrate the semi-supervised and supervised methods specifically per attack type. This approach can be used to calibrate ensembles of detectors, which may be a suitable approach when the profiles of different attacks vary to a large extent. Mimicking rule-based detectors, where each rule corresponds to a single type of attack, we can have an anomaly detector optimized for each attack type. Performance results are shown in Fig. 5. We can see that this approach is generally beneficial for semi-supervised approaches, while in supervised methods it even degrades the results. Among the semi-supervised methods, sparse PLS methods tend to outperform standard PLS, in particular for the '*nerisbotnet*' attack.

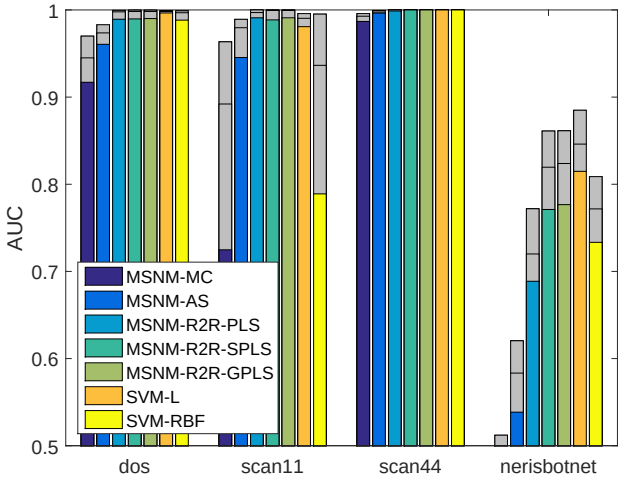

Fig. 5: AUC values per artificial attack type. Supervised and semi-supervised approaches calibrated per attack type. Quartiles (25% and 75%) and the median are shown in the bars.

blueFrom the previous experiments, we can conclude that the R2R optimization leads to a general improvement of the MSNM performance, conforming a semi-supervised approach that is competitive with state-of-the-art techniques and that retains the capability to detect new threats. Note that the un-

supervised MSNM was generally outperformed by supervised techniques, but this comes at a price: supervised techniques are not generally applicable to most real cases, where the labelling of observations is not available. This is actually the commonplace in the cybersecurity industry [40]. Semi-supervised techniques can still be used when none or a partial labelling is available. Furthermore, the MSNM approach has the additional advantage over state-of-the-art supervised techniques to be an interpretable model, and thus easier to use and understand by practitioners [8]. While above we only compared in terms of detection, MSNM also provides diagnosis support, that is, information about why an attack was identified as such. This is actually a principal ability to reduce the time of response to an attack or to quickly identify a false positive. Black-box models, like the non-linear SVM, cannot be interpreted. Therefore, they do not provide the information about why an attack was identified. Finally, it should be noted that none of the SVMs generally outperformed the semi-supervised methods in all the detection experiments performed.

It turns out that the result of the R2R optimization can also be interpreted. Fig. 6 compares the scaling profiles obtained from the optimization with PLS, SPLS and GPLS, and with and without distinguishing among the attack types. In general, sparse methodologies provide clearer profiles, with lower numbers of picks and easier to interpret. The picks reflect those features that are relevant for the detection of the type of attack. We can also see that the optimization using all attack types (GloOpt) is dominated by the *'nerisbotnet'* attack pattern, since in all cases the profile shows a large pick in the same feature than the profile specifically optimized for that attack. Again, this illustrates the limitation of using this semi-supervised approach for a set of disparate attacks, rather than on a per-attack basis.

To interpret the profiles in Fig. 6, we selected those peaks exceeding the average scaling value plus one standard deviation and listed them in Tables III, IV and V. Recall that these variables will have a higher influence on the MSNM detector, but the rest of variables will also impact the detection, to a lesser extent.

In Table III (PLS) we see that the *GloOpt* selects three features: the number of connections with source port 1080 (`sport_socks`), with source port between 49152 and 65535 (`sport_private`) and with destination port 6667 (`dport_irc`). The first feature is related to the SOCKS proxy, an Internet proxy service. That port has been associated in the past to several types of attacks, mainly trojans and SPAM. The second feature is a very general one, and might have been incorrectly selected due to the low signal-to-noise ratio in R2R-PLS. The last feature is related to the *nerisbotnet*. It is out of question that the IRC port is also related to normal activity. However, in the traffic of the network we are monitoring, the amount of IRC is low and the counter in `dport_irc` can be a valid means to detect malicious activity.

Looking at the selection in Table IV, for SPLS, we can see that the number of features selected is reduced and more interpretable, but we still find some potentially inconsistent results. For instance, in the list of the most relevant variables for *nerisbotnet*, we can see that dport_oracle is present, despite the fact that this botnet does not generate any traffic towards the oracle port. This potential inaccuracy is not affecting the detection results of this attack, as shown in Fig. 5, which are considerably improved in comparison to the other MSNM variants. Using GPLS (Table V) the features are subsequently reduced. While this is in general convenient, it does not necessarily imply a benefit in terms of performance. For instance, the global optimizer is mainly focus on the *nerisbotnet* attack. Differences between SPLS and GPLS may be associated to the degree of sparseness of the models, which is governed by the specific metaparameters used but also by the specificities of the training data. In SPLS we used $\mathcal{N}_x = 2$, the most parsimonious value that results in multivariate regression vectors. In GPLS we set $\gamma = 0.8$. The same metaparameters may result in opposite sparseness levels for a different data set. However, we can generally conclude that the use of sparse methods within the R2R algorithm led to improvements in terms of AUC and parsimony, and therefore of interpretability of results.

TABLE III: Variables with highest weights from PLS optimization.

GloOpt:	sport_socks, sport_private dport_irc
SpecOpt dos:	sport_telnet, sport_rapservice dport_http
SpecOpt scan11:	sport_syslog, sport_reserved dport_kpasswd tcpflags_ACK npackets_medium
SpecOpt scan44:	sport_cups dport_telnet, dport_kpasswd
SpecOpt nerisbotnet:	dport_irc tcpflags_PSH npackets_medium

TABLE IV: Variables with highest weights from SPLS optimization.

GloOpt:	sport_private dport_kpasswd, dport_irc
SpecOpt dos:	sport_telnet
SpecOpt scan11:	dport_kpasswd
SpecOpt scan44:	sport_ldaps dport_telnet, dport_kpasswd
SpecOpt nerisbotnet:	dport_oracle, dport_irc

TABLE V: Variables with highest weights from GPLS optimization.

GloOpt:	dport_irc
SpecOpt dos:	sport_telnet
SpecOpt scan11:	dport_kpasswd
SpecOpt scan44:	dport_telnet, dport_kpasswd
SpecOpt nerisbotnet:	dport_irc

VIII. Discussion and Conclusion

In this paper, we provide a solution to combine the advantages of unsupervised and supervised learning in the context of intrusion detection. For this, we use the Multivariate Statistical Network Monitoring approach, recently proposed, and we enhance an optimization algorithm based on Partial

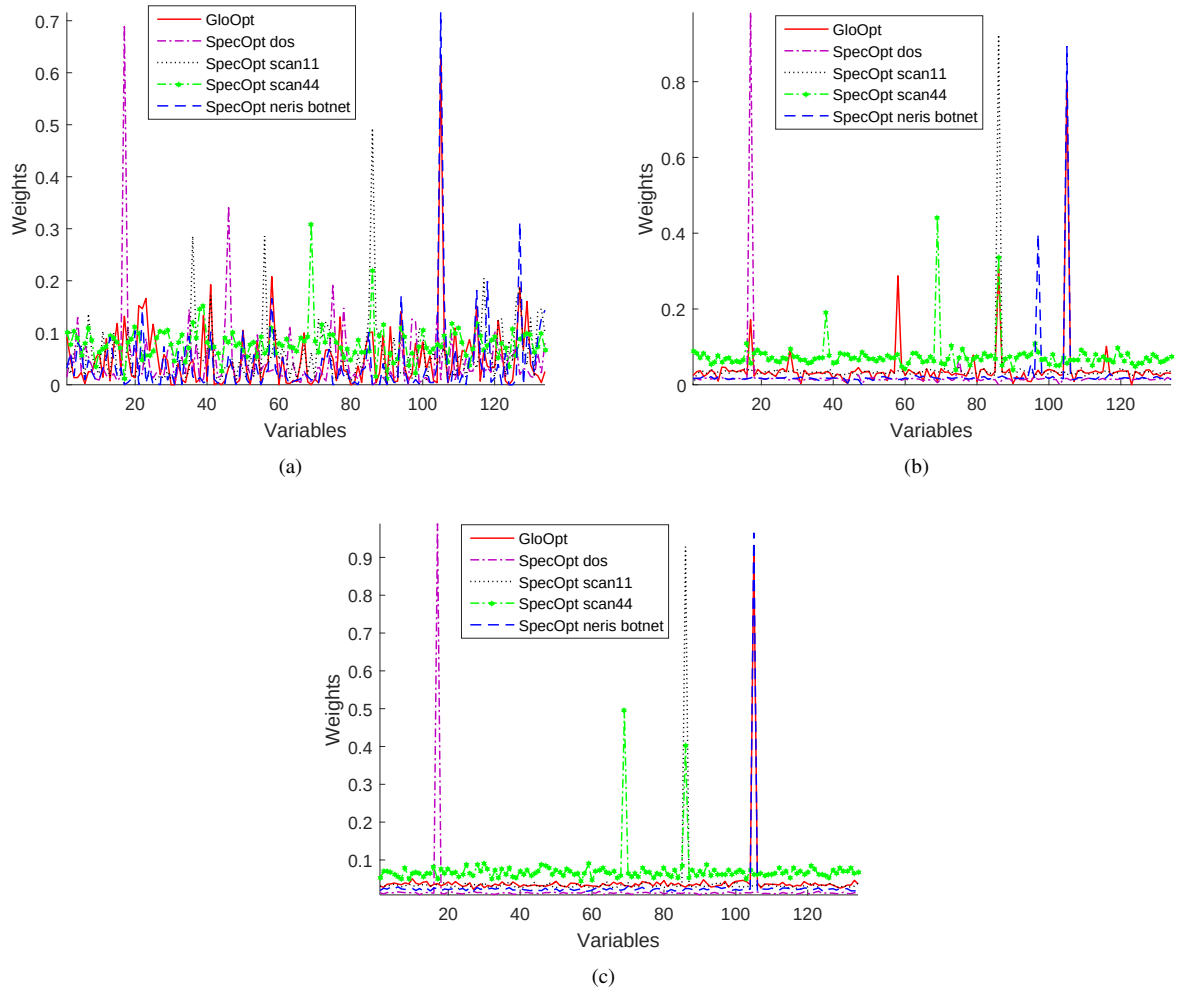

Fig. 6: Optimized weights by PLS (a), SPLS (b) and GPLS (c). GlobOpt makes reference to the optimized profile obtained with no distinction among attack types. SpecOpt makes reference to the optimized profile per attack type.

Least Squares, specially suited for multivariate optimization problems. The result is an anomaly detection system that can be optimized for the detection of a set of attack patterns. Our approach provides a machine learning technique with similar flexibility to update to new attack patterns as in a rule based system. Combined with unsupervised methods, we can still identify unseen (zero-day) patterns of malicious activity. This paper also introduces for the first time the application of sparse methodologies in intrusion detection, which were seen to be very effective within the proposed semi-supervised detection machine. Results with real traffic showed the practical applicability of the approach.

ACKNOWLEDGMENT

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through projects TIN2014-60346-R and TIN2017-83494-R. Anonymous reviewers are acknowledged for their useful comments.

REFERENCES

[1] VERIZONE, "Data breach investigation report," 2017.

[2] M. Solomon. The multiplier effect of collaboration for security operations. [Online]. Available: https://www.securityweek.com/multiplier-effect-collaboration-security-operations

[3] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1, pp. 18 – 28, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404808000692

[4] S. Dua and X. Du, *Data mining and machine learning in cybersecurity*. CRC press, 2016.

[5] A. Ferrer, "Latent structures-based multivariate statistical process control: A paradigm shift," *Quality Engineering*, vol. 26, no. 1, pp. 72–91, 2014.

[6] G. Fernandes, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença, "Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization," *Journal of Network and Computer Applications*, vol. 64, pp. 1–11, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516000618

[7] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 219–230, oct 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1030194.1015492

[8] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, June 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404816300116

[9] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for

traffic anomaly detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 109–120, jun 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1269899.1254895

[10] J. Camacho, J. Picó, and A. Ferrer, "Self-tuning run to run optimization of fed-batch processes using unfold-pls," *AIChE Journal*, vol. 53, no. 7, pp. 1789–1804, 2007.

[11] J. Camacho, D. Lauri, B. Lennox, M. Escabias, and M. Valderrama, "Evaluation of smoothing techniques in the run to run optimization of fed-batch processes with u-PLS," *Journal of Chemometrics*, vol. 29, no. 6, pp. 338–348, 2015. [Online]. Available: http://dx.doi.org/10.1002/cem.2711

[12] H. Martens and T. N. s, *Multivariate Calibration*. John Wiley & Sons, 1992.

[13] P. Geladi and B. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, pp. 1–17, 1986.

[14] C. Colombani, P. Croiseau, S. Fritz, F. Guillaume, a. Legarra, V. Ducrocq, and C. Robert-Granie, "A comparison of partial least squares (PLS) and sparse PLS regressions in genomic selection in French dairy cattle," *J Dairy Sci*, vol. 95, no. 4, pp. 2120–2131, 2012.

[15] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón Sánchez, "Ugr'16: a new dataset for the evaluation of cyclostationarity-based network IDSs," *Computer & Security*, November 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817302353

[16] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Wiley-IEEE Press, 2007.

[17] E. Alpaydin, *Introduction to Machine Learning*. Massachusetts Institute of Technology, 2010.

[18] S. Skansi, *Introduction to Deep Learning. From Logical Calculus to Artificial Intelligence*. Springer, 2018.

[19] A. Kanaoka and E. Okamoto, "Multivariate statistical analysis of network traffic for intrusion detection," *14th international workshop on Database and Expert System Applications*, pp. 1–5, 2003.

[20] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel PCA-based network anomaly detection," in *IEEE International Conference on Communications*, 2011.

[21] A. Delimargas, E. Skevakis, H. Halabian, and I. Lambadaris, "Evaluating a modified PCA approach on network anomaly detection," *Fifth International Conference on Next Generation Networks and Services (NGNS)*, pp. 124–131, 2014.

[22] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "Improving PCA-based anomaly detection by using multiple time scale analysis and Kullback-Leibler divergence," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1731–1751, oct 2014. [Online]. Available: http://doi.wiley.com/10.1002/dac.2432

[23] M. Aiello, M. Mongelli, E. Cambiaso, and G. Papaleo, "Profiling DNS tunneling attacks with PCA and mutual information," *Logic Journal of IGPL*, pp. 1–14, 2016. [Online]. Available: http://jigpal.oxfordjournals.org/lookup/doi/10.1093/jigpal/jzw056

[24] J. Camacho, P. García-Teodoro, and G. Maciá-Fernández, "Traffic Monitoring and Diagnosis with Multivariate Statistical Network Monitoring: A Case Study," *IEEE Security & Privacy International Workshop on Traffic Measurements for Cybersecurity (WTMC 2017)*, 2017.

[25] D. G. Luenberger and Ye, *Linear and Nonlinear Programming*. Springer. International Series in Operations Research & Management Science 228, 2008.

[26] H. Chun and S. Keleş, "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 1, pp. 3–25, 2010.

[27] E. Andries, "Sparse models by iteratively reweighted feature scaling: A framework for wavelength and sample selection," *Journal of Chemometrics*, vol. 27, no. 3-4, pp. 50–62, 2013.

[28] R. Calvini, A. Ulrici, and J. M. Amigo, "Practical comparison of sparse methods for classification of Arabica and Robusta coffee species using near infrared hyperspectral imaging," *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 503–511, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.chemolab.2015.07.010

[29] J. Camacho and E. Saccenti, "Groupwise partial least square regression," *Journal of Chemometrics*, vol. 32, no. 3, p. e2964. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.2964

[30] J. Camacho, G. Maciá-Fernández, J. Díaz-Verdejo, and P. García-Teodoro, "Tackling the big data 4 vs for anomaly detection," *Proceedings - IEEE INFOCOM*, no. 1, pp. 500–505, 2014.

[31] P. Nomikos and J. F. MacGregor, "Multivariate statistical process control charts for monitoring batch processes," *Technometrics*, vol. 3, no. 3, pp. 403–414, 1995.

[32] 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*, vol. 25, no. 11, pp. 592–600, 2011.

[33] R. Tibshirani, "Regression Selection and Shrinkage via the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[34] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[35] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, and P. Besse, "A sparse pls for variable selection when integrating omics data," *Statistical applications in genetics and molecular biology*, vol. 7, no. 1, 2008.

[36] A. Lorber, L. E. Wangen, and B. R. Kowalski, "A theoretical foundation for the pls algorithm," *Journal of Chemometrics*, vol. 1, no. 1, pp. 19–31, 1987.

[37] H. Shen and J. Z. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of multivariate analysis*, vol. 99, no. 6, pp. 1015–1034, 2008.

[38] J. Camacho and E. Saccenti, "Group-wise Partial Least Squares Regression," *Journal of Chemometrics (Wiley)*, vol. 32, no. 3, p. 1:11, 2018.

[39] CTU-13 dataset. [Online]. Available: https://stratosphereips.org/category/dataset.html

[40] K. Kavanagh and O. Rochford, "Critical capabilities for security information and event management," *Gartner*, 2015.

José Camacho is Associate Professor in the Department of Signal Theory, Telematics and Communication and researcher in the Information and Communication Technologies Research Centre, at the University of Granada, Spain. He holds a degree in Computer Science from the University of Granada (2003) and a Ph.D. from the Technical University of Valencia (2007). His Ph.D. was awarded with the second Rosina Ribalta Prize to the best Ph.D. projects in the field of Information and Communication Technologies (ICT) from the EPSON Foundation, and with the D.L. Massart Award in Chemometrics from the Belgian Chemometrics Society. His research interests include exploratory data analysis, anomaly detection and optimization with multivariate techniques applied to data of very different nature, including manufacturing processes, chemometrics and communication networks. He is especially interested in the use of exploratory data analysis to Big Data for network security.

Gabriel Maciá-Fernández is an Associate Professor at the Department of Signal Theory, Telematics and Communications of the University of Granada (Spain). He received a MS in Telecommunications Engineering from the University of Seville, Spain, and the Ph.D. in Telecommunications Engineering from the University of Granada. In the period 1999-2005, he worked as a specialist consultant at "Vodafone Espaa". His research interests are focused on computer and network security, with special focus on intrusion detection, reliable protocol design, network information leakage and denial of service.

Marta Fuentes is a Ph.D. student in the Department of Signal Theory, Telematics and Communications of the University of Granada (Spain). She has a degree in Computer Sciences by this same University since 2012. Since then, she has been working for several companies, what has driven her to find a link between research and enterprise. In 2015 she also studied a Master Degree in Software Development in the University of Granada. Her PhD is based in network monitoring for anomalies detection and diagnosis by using data analysis.

Edoardo Saccenti received a MSc degree in Physics and a PhD degree in Structural Biology from the University of Florence, Italy. His main research is multivariate statistics in particular: Principal components analysis and related methods with a focus on the problem of dimensionality assessment and its relationships with inferential statistics in the frame of Random Matrix Theory; power analysis and sample size determination in the context of PCA, PLS-DA and network inference; sparse component methodologies for data exploration and interpretation.