Robust Feature Selection and Robust PCA for Internet Traffic Anomaly Detection

Cláudia Pascoal*, M. Rosário de Oliveira*, Rui Valadas†, Peter Filzmoser‡, Paulo Salvador§ and António Pacheco*
*UTL/IST and CEMAT, Lisbon, Portugal, Email: {cpascoal,rsilva,apacheco}@math.ist.utl.pt
†UTL/IST and Instituto de Telecomunicações, Lisbon, Portugal, Email: rui.valadas@ist.ut.pt
‡Dept. of Statistics and Probability Theory, Vienna Univ. of Technology, Vienna, Austria, Email: p.filzmoser@tuwien.ac.at
§DETI/Univ. of Aveiro and Instituto de Telecomunicações, Aveiro, Portugal, Email: salvador@av.it.pt

***Abstract*—Robust statistics is a branch of statistics which includes statistical methods capable of dealing adequately with the presence of outliers. In this paper, we propose an anomaly detection method that combines a feature selection algorithm and an outlier detection method, which makes extensive use of robust statistics. Feature selection is based on a mutual information metric for which we have developed a robust estimator; it also includes a novel and automatic procedure for determining the number of relevant features. Outlier detection is based on robust Principal Component Analysis (PCA) which, opposite to classical PCA, is not sensitive to outliers and precludes the necessity of training using a reliably labeled dataset, a strong advantage from the operational point of view. To evaluate our method we designed a network scenario capable of producing a perfect ground-truth under real (but controlled) traffic conditions. Results show the significant improvements of our method over the corresponding classical ones. Moreover, despite being a largely overlooked issue in the context of anomaly detection, feature selection is found to be an important preprocessing step, allowing adaption to different network conditions and inducing significant performance gains.**

I. Introduction

Despite the tremendous research effort around methods for anomaly detection in Internet traffic [1]–[7], there are still several issues preventing their deployment in real systems [8]. The main difficulty is probably the cost of errors, which is much higher than in other machine learning applications. Indeed, false negatives can cause severe damage to users and organizations, while false positives, even if occurring at low rate, can render an anomaly detector useless. We address this problem by proposing an (highly performant) anomaly detector that has two key characteristics: first, it couples a *feature selection* algorithm with an *outlier detection* method; second, it uses *robust statistics* tools in both procedures.

Robust statistics is a branch of statistics that is capable of integrating the presence of outliers in the statistical analysis. Over the years, robust versions for Regression Analysis, Discriminant Analysis, Cluster Analysis, Principal Component Analysis (PCA), Time Series, and many other classical statistical tools have been developed. A recent overview is provided in [9]. The robustified versions give reliable results even if outliers are present in the data. We believe that this large body of knowledge can be helpful in addressing many statistically related computer networking problems, and illustrate its strength in the context of anomaly detection. Our anomaly detector uses robust statistics both in the feature selection algorithm and in the outlier detection method. Feature selection is based on a robust mutual information estimator and outlier detection on robust PCA.

Feature selection is an important preprocessing step when analyzing high-dimensional data, to reduce dimensionality, remove irrelevant data, and increase learning accuracy. It has been widely investigated in the context of several machine learning applications, e.g., in the identification of Internet applications [10]–[12]. Strangely, it has been a largely overlooked issue in the framework of anomaly detection. In this work, we show that significant performance gains can be attained by adding this filtering stage.

Lakhina et al. [3] popularized the use of PCA for (network-wide) anomaly detection. Subsequent work [13] highlighted the limitations of classical PCA. Since classical PCA is sensitive to the presence of outliers, the only way to achieve good performance is to train the detector using only licit traffic, which requires a *perfect* ground truth, i.e., a dataset where one knows for sure whether a traffic object is an anomaly or corresponds to licit traffic. Such dataset is almost impossible to obtain with real traffic, since attacks, the majority of anomalies, try to travel unnoticed in networks [8], [14]. To address the limitations of classical PCA, [5] proposed the use of the Karhunen-Loeve expansion. Another alternative, which has been only explored by few authors, is to use robust PCA [1], [6]. Recently, [6] used a robust PCA algorithm based on projection pursuit, called PCAGRID, that we have developed in [15]. In this work, we will also use PCAGRID, considering the thresholds suggested in [16].

The problems in obtaining a perfect ground-truth also affect the evaluation of novel anomaly detection methods, because performance metrics can become biased when datasets are imperfectly labeled. This led some authors to advocate the use of computer simulation to compose ground-truth datasets [14]. However, difficulties in modeling the user behavior and the networking mechanisms that most impact the traffic, and in reverse-engineering several types of network attacks, render it hard to mimic real traffic through simulation [8], [17]. We believe that the right answer to the ground-truth problem is to use datasets captured in a (highly protected) laboratory

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

network, where the attacks are emulated internally, and the licit traffic is produced by a selected set of users accessing the Internet, but constrained on the type of applications they can use and on the sites they can visit. This is probably the closest we can get from a real traffic scenario, while assuring that licit traffic is not mistaken by anomalies (and vice-versa).

One important strength of our anomaly detector is its adaptability to different network environments and traffic conditions, a feature deemed critical by Sommer and Paxson [8]. Indeed, contrary to classical PCA, our robust PCA detector does not require having a perfect ground-truth for training; it just trains from the (eventually contaminated) background traffic. It is, in fact, an *unsupervised* learning algorithm. Moreover, the addition of feature selection allows matching the specificities of the traffic the detector is operating on, which can vary widely both in terms of the most frequent attacks and the relative weight of licit applications. Feature selection does require training with a labeled dataset but, due to its robustification, it is much more insensitive to mislabeled anomalies and atypical observations than previous proposals.

The paper is structured as follows. In section II we give a short tutorial on robust statistics. The feature selection algorithm and the outlier detection methods are presented in sections III and IV, respectively. Section V introduces the dataset used in the performance evaluation and discusses the results. Finally, section VI gives the main conclusions of the paper.

II. A PRIMER ON ROBUST STATISTICS

In statistical terminology, anomalies are often called *outliers*, referring to observations with different data structure than the majority of the available data points. This difference in data structure has to be described with appropriate statistical estimators, and it is important to use estimators that are themselves not affected by the outliers. Such estimators are said to be robust, because even in case of a certain amount of contamination they will yield reliable results.

The simplest case for outlier detection is the univariate case. Figure 1 (top) shows 15 simulated data drawn from a univariate normal distribution with mean 50 and standard deviation 10. The overlain density estimation confirms the closeness of the sample to normal distribution, and it also reveals that no outliers (or very extreme observations) are present. On the other hand, Figure 1 (bottom) shows modified data, where the two highest original values have been moved to the positions 80 and 90, respectively. The two outliers clearly have an effect on the density estimation, but they also can affect statistical estimates required for an automated procedure for outlier detection. For constructing such an outlier detection rule, it is common to look at the interval "center$\pm 1.96\times$spread": since under normal distribution this interval contains approximately the inner 95%, data points outside this interval can be considered as outliers (atypical observations). Of course, center and spread need to be estimated from the available data. The center is usually estimated by the arithmetic mean, but a more robust alternative is the median. Similarly, a classical way to estimate the spread is the empirical standard deviation, but a more robust alternative is the MAD (median absolute deviation), defined for a sample $\boldsymbol{x} = (x_1, \ldots, x_n)^t$ as

$$\mathrm{MAD}(\boldsymbol{x}) = 1.4826\,\mathrm{med}_i|x_i - \mathrm{med}(\boldsymbol{x})|,$$

where $\mathrm{med}(\boldsymbol{x})$ denotes the median of the data. Applying the classical and the robust rule to the original and the modified data results in the intervals shown in Figure 1. Both rules give similar answers for the uncontaminated data (top). For the contaminated data (bottom) the robust rule does not change, but the classical rule is very sensitive to the outliers. Not only the arithmetic mean has increased, also the standard deviation has been inflated by the outliers, leading to a rule that identifies only one of the two outliers (this phenomenon is called *masking effect*). There exist several alternative robust estimators of location and spread in the literature, e.g., see [9]. The basic idea of these estimators is to automatically assign weights to the observations according to their "outlyingness". The specification of the weighting scheme directly determines the statistical properties of the estimators. Note that although for computing the median most observations receive weight zero, these observations are not simply discarded from the data, because their information is still needed for sorting the data.

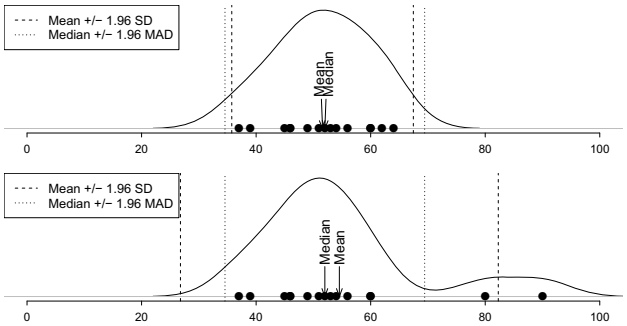

Fig. 1. Univariate outlier detection rules applied to clean (top) and contaminated (bottom) data. The classical rule is based on the arithmetic mean (`Mean`) and the empirical standard deviation (`SD`), while the robust rule uses the median (`Median`) and the median absolute deviation (`MAD`).

For anomaly detection there is usually multivariate information available. Hence, we are in the context of multivariate outlier detection, which is potentially different from the univariate case. A simulated example in Figure 2 demonstrates the difference. The bulk of the data comes from a bivariate normal distribution, while several data points deviate from this structure. Applying the robust univariate outlier detection rule along each coordinate results in the horizontal and vertical lines. Accordingly, outliers would be data points falling outside the inner rectangular region. Obviously, this approach is not useful, because the main data structure is elliptically symmetric, and the outliers are not extreme along the individual coordinates. The indicated ellipse results from a multivariate treatment of the problem. It corresponds to a threshold for the Mahalanobis distance, defined for a point $\boldsymbol{x}$ in the bivariate

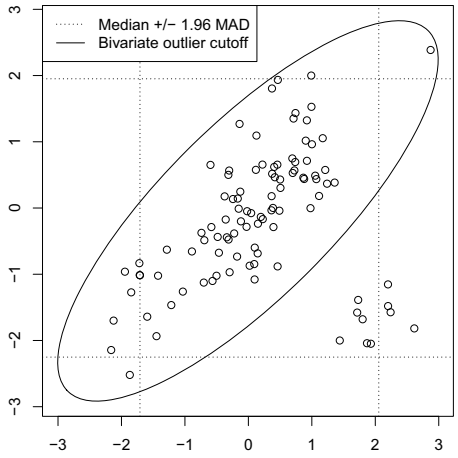

Fig. 2. Outlier detection for contaminated bivariate data: Robust univariate rules along the coordinates (dotted lines) result in a rectangular region separating regular points from outliers. Only the bivariate treatment of the problem leads to the elliptical region corresponding to the main data structure.

data space as

$$\mathrm{MD}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})},$$

with $\boldsymbol{\mu}$ the center and $\boldsymbol{\Sigma}$ the covariance matrix. Both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ need to be estimated from the available data, and here it is again important to use robust estimates [9]. For normally distributed (bivariate) data, the squared Mahalanobis distance follows approximately χ^2_2, a chisquare distribution with 2 degrees of freedom. Therefore, the outlier threshold for the Mahalanobis distances can be defined as $\sqrt{\chi^2_{2;0.975}}$, the square-root of the 0.975-quantile of this distribution; see also [18]. The concept of multivariate outlier detection based on Mahalanobis distances will also be used in the following. Note, however, that each variable (feature) contributes equally to the Mahalanobis distance. Especially in problems with many variables, this approach can be poor if many of the variables are non-informative for the outlier detection problem. Thus, in a first approach only those features that contain potential information about the outliers will be selected.

III. Feature selection

Feature selection using filter methods involves a measure of association between the variables (features) used to characterize objects and the classification classes they belong to, which in our case are only two (anomalous or regular) [19]. Our work is based on the Mutual Information (MI), an information-theoretic metric that captures both linear and non-linear dependencies, and has recently gained wide acceptance [20]. The MI between random variables X and Y is defined by

$$\mathrm{MI}(X,Y) = \iint f_{XY}(x,y) \log\left(\frac{f_{XY}(x,y)}{f_X(x) f_Y(y)}\right) \mathrm{d}x\,\mathrm{d}y$$

where $f_{XY}(x,y)$ denotes de probability density function of the random vector (X,Y) and $f_Z(z)$ the probability density function of a random variable Z.

Many estimators of MI have been proposed in the literature. In [21] we have performed a comparison of eleven different estimators and concluded that the empirical estimator with equal bins discretization achieves the best compromise between performance and complexity. This estimator is given by

$$\hat{\mathrm{MI}} = -\sum_{i=1}^{m_X} \hat{p}_i \log \hat{p}_i - \sum_{j=1}^{m_Y} \hat{q}_j \log \hat{q}_j + \sum_{i=1}^{m_X}\sum_{j=1}^{m_Y} \hat{p}_{ij} \log \hat{p}_{ij} \quad (1)$$

where m_X and m_Y are the number of bins considered for x- and y-sample, $\hat{p}_i$ and $\hat{q}_j$ are the proportions of observations from x- and y-sample that fall in the i-th and j-th bin, respectively, and finally $\hat{p}_{ij}$ is the proportions of the (x_k, y_k) observations such that x_k belongs to the i-th bin and y_k belongs to the j-th bin associated with the x- and y-sample, respectively.

A. Robustification of the mutual information estimator

The computation of the MI estimator requires a labeled dataset, indicating the class each object belongs to. The classical MI estimators, including (1), are all sensitive to errors in dataset labeling, as well as atypical observations in each class. As discussed in section I, in the context of anomaly detection, and probably more here than in any other machine learning application, it is very difficult to obtain a perfectly labeled dataset from real data. In order to overcome this problem, we use robust statistics to obtain a (much) less sensitive MI estimator. The basic idea is to use the robust thresholds for univariate data described in section II to remove (possible) outliers both from the set of regular observations and the set of anomalies.

Let X be a random variable representing a feature, Y the binary labeled random variable that represents the class, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^t$ a size n sample of X and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^t$ the n-dimensional vector where y_j is the label of observations x_j, $j = 1, \ldots, n$. We will represent the regular observations by label 0 and the anomalies by 1, on variable Y. The proposed algorithm to estimate MI is the following:

1. Separate the sample $\boldsymbol{x}$ into 2 subsets:
   a) $\mathcal{C}_0 = (x_{01}, x_{02}, \ldots, x_{0n_0})^t$, composed by the regular observations, i.e. with label 0;
   b) $\mathcal{C}_1 = (x_{11}, x_{12}, \ldots, x_{1n_1})^t$, composed by the anomalies, i.e. observations with label 1;
2. Exclude from $\mathcal{C}_0$ the observations outside the interval $[\mathrm{c_L}, \mathrm{c_U}]$ where $\mathrm{c_L} = \mathrm{med}\,(\mathcal{C}_0) - z_{1-\alpha/2}\mathrm{MAD}\,(\mathcal{C}_0)$ and $\mathrm{c_U} = \mathrm{med}\,(\mathcal{C}_0) + z_{1-\alpha/2}\mathrm{MAD}\,(\mathcal{C}_0)$, obtaining $\mathcal{C}_0^*$. Here z_γ denotes the γ-quantile of the standard normal distribution. In this work we chose $\alpha = 0.05$;
3. Exclude from $\mathcal{C}_1$ the observations within the interval $[\mathrm{c_L}, \mathrm{c_U}]$, obtaining $\mathcal{C}_1^*$;
4. Estimate MI based on (x_j, y_j) where $x_j \in \mathcal{C}_0^* \cup \mathcal{C}_1^*$.

The thresholds used in our algorithm assume that regular observations are approximately normally distributed. We also

assume a single interval, obtained from the regular observations: any regular observation outside the interval $[c_L, c_U]$ is considered an outlier among the regular observations; and any anomaly inside this interval is considered an outlier among the anomalies. Two possible extensions can be considered. First, there can be two distinct intervals, one for the set of regular observations and another for the set of anomalies. This is only feasible when the number of anomalies is sufficiently large, to allow obtaining good estimates. Second, if the normal assumption is not adequate either for anomalies or regular observations, e.g. data is skewed, alternative ways to estimate the intervals $[c_L, c_U]$, like the ones proposed in [22], may be considered.

B. Automatic method to select relevant features

Let X_i represent feature i, with $i = 1, \ldots, d$, and d the number of features we have to select from. Once an association measure between each feature and the class is available, the usual method to select the relevant features has been a user-defined threshold [23], [24]. In this case, a feature X_i would be considered relevant iff $\hat{\text{MI}}(X_i, Y) \geq \eta$, where η is fixed in advance. Another alternative is to plot the ordered MI estimates and look for a sharp drop, similar to the process of choosing the number of Principal Components (PCs) to retain in PCA. If the feature selection method is to be used in dissimilar environments, with distinct background traffic, the necessity to fine-tune manually any parameter should be avoided. That is why we have devised an automatic procedure to select the relevant features. The basic idea is to partition the ordered set of features in two subsets, where one includes the features with highest MI estimates, and then search for the partition that achieves the highest separation among subsets, using the t-test statistic as a separation metric. t-tests are used to evaluate whether or not the separation between subsets is statistically significant. Our algorithm is the following:

1. Estimate $\text{MI}(X_i, Y)$, $i = 1, \ldots, d$;
2. Sort the features X_i, $i = 1, \ldots, d$, according to the descending order of their estimated MI, obtaining the set of sorted features $\boldsymbol{X}^* = (X_1^*, X_2^*, \ldots, X_d^*)^t$;
3. Choose an initial subset $\mathcal{S}_0 = (X_1^*, X_2^*, \ldots, X_k^*)^t$ of relevant features composed by the k features with the highest estimated MI. Let $\mathcal{S}_1 = \left(X_{k+1}^*, X_{k+2}^*, \ldots, X_d^*\right)^t$ be the subset of the (remaining) non-relevant features;
4. Test H_0 : on average the MI of the relevant ($\mathcal{S}_0$) and non-relevant ($\mathcal{S}_1$) features are equal, against H_1 : on average the relevant features have larger MI than the non-relevant features. To do this, calculate the t-test statistics $t = (\overline{w}_{\bullet 0} - \overline{w}_{\bullet 1}) / \sqrt{\frac{s_0^2}{k} + \frac{s_1^2}{d-k}}$, where $w_{ij} = \hat{\text{MI}}(X_i^*, Y)$, $X_i^* \in \mathcal{S}_j$, $j = 0, 1$, and s_j^2 is the sample variance of the estimated MI associated with $\mathcal{S}_j$. Consider the Welch modification for the degrees of freedom of the t-distribution.
5. Update $\mathcal{S}_0 = \left(X_1^*, X_2^*, \ldots, X_k^*, X_{k+1}^*\right)^t$ and $\mathcal{S}_1 = \left(X_{k+2}^*, \ldots, X_d^*\right)^t$;
6. Repeat step 4. and 5. until only two features are left in $\mathcal{S}_1$.
7. For the $(d - k - 1)$ tests performed, select the highest t-test statistic, $t_{\max}$, and verify if it is statistically significant in the following way. Fix a global significance level, α, for all tests, and use the Bonferroni correction. The associated null hypothesis is considered statistically significant if $t_{\max}$ is larger than the $(1 - \alpha/(d - k - 1))$ −quantile of the t-distribution, with the corresponding Welch modified degrees of freedom.
8. If the statistical test associated with $t_{\max}$ is statistically significant then the features in set $\mathcal{S}_0$ from which this statistic is calculated are marked as the relevant ones. Otherwise, all features are considered equally relevant.

Note that both the robustification process and the method for automating the feature selection apply equally to association metrics other than the MI, e.g. the symmetrical uncertainty used in [23]. These methods do not account for possible associations and redundancies among the initial features. Such enhancements are left for further study.

C. Evaluation

We developed a simulation study to assess the performance of the algorithm with and without robustification of the MI measure, using in both cases the automatic method to select relevant features. We generated samples of size 300 for each of 80 features, 50 relevant and 30 non-relevant. Among the relevant ones, two groups were considered: 30 features more associated to Y (MI=0.279), called *m*-relevant, and 20 less associated (MI=0.217), called *l*-relevant. To simulate the data we use a setup similar to the one described in [25]. The relevant features are pairwise generated such that: (i) exactly 10% of the generated observations are anomalies, (ii) a regular pair of features has bivariate normal distribution with mean vector $\boldsymbol{\mu}_0 = (0, 0)^t$ and covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{2} \begin{pmatrix} v+1 & v-1 \\ v-1 & v+1 \end{pmatrix},$$

where $v = \sqrt{40}$; and (iii) an anomaly has also a bivariate normal distribution with the same covariance matrix and mean vector $\boldsymbol{\mu}_1 = 10\boldsymbol{e}$, $\boldsymbol{e} = \sqrt{2}/2(1, -1)^t$ for the *m*-relevant features, and $\boldsymbol{\mu}_2 = 7.5\boldsymbol{e}$ for the *l*-relevant ones. The non-relevant features have independent normal distribution with null mean, $v/\sqrt{2}$ variance, and are independent from Y, which means a null MI. To simulate an imperfect ground-truth, with wrongly labeled data, we changed $\tau/2 \times 100\%$ of all observations from regular to anomalies, and the same percentage otherwise, for $\tau \in \{0, 0.07, 0.10\}$. Each simulation setup was replicated 100 times.

Since we are interested in the features selected as relevant, we consider as performance metrics the proportion of simulation runs where all *m*-relevant features are selected as relevant (p_m) and the same proportion for the *l*-relevant features (p_l). The results are shown in Table I. When the contamination is

TABLE I
PROPORTION OF SELECTED FEATURES FOR DIFFERENT CONTAMINATIONS

	Non-robust		Robust	
τ	$p_{m\times 100\%}$	$p_{l\times 100\%}$	$p_{m\times 100\%}$	$p_{l\times 100\%}$
0	100	100	100	100
0.07	100	95	100	95
0.10	99	27	100	67

low ($\tau = 0$ or $\tau = 0.07$) the classical and robust estimators detect all the *m*-relevant features as relevant and the majority (if not all) *l*-relevant features also as relevant. However, for a contamination of $\tau = 0.10$, the robust estimator performs much better: in the case of *l*-relevant features, the robust estimator selects all as relevant in 67% of simulation runs, but the classical one selects only in 27%. Moreover, in the worst case, the classical estimator selects 28 non-relevant features as relevant, while the robust one selects only 2. These results clearly show the resilience of the robust estimator to errors in data labeling. We will return to this issue in section V, using a dataset obtained under real network conditions.

IV. OUTLIER DETECTION

Feature selection using the robust estimation of the MI allows to reduce the d features to the relevant ones. In the following, we assume that $p \leq d$ relevant features are remaining, and are characterized by the p random variables $\boldsymbol{X} = (X_1, \ldots, X_p)^t$. This possible reduction of the features will be important for the performance of the outlier detection procedure.

For multivariate outlier detection it is common to use the Mahalanobis distance, see Section II. For robustly estimating the Mahalanobis distances it is necessary to first estimate the covariance matrix of the data in a robust manner. There are two difficulties with this approach: (a) the number p of relevant features might still be too high in order to estimate robustly the covariance matrix; (b) a condensed data information rather than the complete covariance information might be preferable for reliable outlier detection in this context. Compressing information is most conveniently done by using PCA, because this method has statistically appealing properties, and it is known as a valuable tool for detecting atypical observations in the transformed space spanned by the first principal components [26]. Reliable outlier detection is, however, only possible with a robust version of PCA. Both, classical and robust PCA are described briefly in the following.

A. Classical PCA

PCA is a mathematically simple procedure, implemented in a variety of software packages, and highly successful in the analysis of real data. It is mainly used to reduce the dimensionality of the problem under study, which hopefully reveals interesting structure among the data and simplifies the analysis and the interpretation [1], [27]. Nevertheless, the method also has limitations and disadvantages: it is not scale invariant; multiple criteria exist for choosing the number of PCs to be retained; the method is based on the assumption of linearity, which can be an unrealistic hypothesis to model some real problems [13]. Another important limitation is its high sensitivity to outliers or anomalies, whose presence can lead to completely wrong conclusions about the data [1], [13], [18], [22], [26], [28]. Jolliffe [27] gives a more complete discussion about this method.

From the mathematical point of view, PCA seeks to maximize the variance of uncorrelated linear combinations of the original variables [27], called principal components. If a small number of PCs explain a large proportion of the total variance of the p original variables, then PCA can be successfully used as a dimension reduction technique. In the following we assume that k ($1 \leq k \leq p$) PCs are retained. Given the random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^t$, with mean vector $\boldsymbol{\mu}$, the j-th principal component is defined as the linear combination, $Z_j = \boldsymbol{\gamma}_j^t(\boldsymbol{X} - \boldsymbol{\mu})$, such that $\boldsymbol{\gamma}_j^t\boldsymbol{\gamma}_j = 1$, Z_j has maximum variance and is uncorrelated with the previous PCs (for $2 \leq j \leq k$). It can easily be proved that the loadings, $\boldsymbol{\gamma}_j$, and the variances of the PCs, i.e. $\lambda_j = \mathrm{Var}(Z_j)$, are, respectively, the eigenvectors and eigenvalues of the covariance matrix of $\boldsymbol{X}$, where the eigenvalues are arranged in decreasing order of magnitude.

In real applications, $\boldsymbol{\gamma}_j$ and λ_j have to be estimated. Given $\boldsymbol{x}_i$, the values associated with the random vector $\boldsymbol{X}$ on subject i, the score of subject i on the j^{th} PC is given by $z_{ij} = \hat{\boldsymbol{\gamma}}_j^t(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})$ where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\gamma}}_j$ denote the estimates of the mean vector and the loadings, and $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ik})^t$ represents the subject i in the PCA sub-space, spanned by the first k PCs.

B. Robust PCA

In classical PCA, the eigenvectors and eigenvalues are estimated from the sample covariance matrix, which is very sensitive to outliers. A straightforward robustification of PCA is to use a robust covariance estimation for the eigen-decomposition. However, robust covariance estimation is not trivial for high-dimensional data, and thus other approaches are considered here. The Projection Pursuit (PP) approach to PCA [15], [29] seeks for univariate projections of the multivariate data that maximize a robust scale. Subsequent directions are again found by maximizing a robust scale, but orthogonality constraints to the previous directions are imposed [15], [29]. Several variants of this idea have been explored leading to different estimation methods. Here we will use the method PCAGRID proposed in [15] because of its high robustness and precision. Another approach, called ROBPCA [16], combines PP ideas with robust scatter matrix estimation. It is known that the method produces accurate estimates for non-contaminated data sets and robust ones for contaminated data. Besides this, it is computationally fast and can be applied to datasets with more variables than observations. A simulation study [30] covering five robust PCA methods, as well as the classical one, confirmed PCAGRID and ROBPCA as the best options.

The robust PCs now represent the data effectively in a lower-dimensional space. In this space we need to identify

outliers, which means that appropriate distance measures need to be developed that support a decision on outlyingness. There are two important distance measures in the context of PCA: the *score distance* $\mathrm{SD}(\boldsymbol{x}_i) = \left(\sum_{j=1}^{k} z_{ij}^2/\hat{\lambda}_j\right)^{1/2}$ corresponding to the Mahalanobis distance in the PCA space, and the *orthogonal distance* $\mathrm{OD}(\boldsymbol{x}_i) = \|(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}) - \boldsymbol{P}_{p,k}\boldsymbol{z}_i\|$, which is the distance of an observation to the PCA space. $\boldsymbol{P}_{p,k}$ is the $p \times k$ matrix containing the loadings of the first k robust PCs in its columns [16]. For both distance measures it is necessary to define thresholds that allow to distinguish between regular observations and outliers. We have used the thresholds suggested in [16]. For the orthogonal distance, we used $\mathrm{c_{OD}} = (\hat{\mu} + z_{p_1}\hat{\sigma})^{3/2}$, where z_{p_1} is the p_1-quantile of the standard normal distribution and $\hat{\mu}$ and $\hat{\sigma}$ are, respectively, location and scale estimators. For the score distance, we used $\mathrm{c_{SD}} = \sqrt{\chi^2_{k,p_2}}$, where χ^2_{k,p_2} denotes the p_2-quantile of the χ^2 distribution with k degrees of freedom and k is the number of retained PCs. We considered $p_1 = 0.999$ and $p_2 = 0.05$. For a robust approach, the location μ and the scale σ are estimated by the univariate MCD estimators, and in the classical case by the corresponding sample ones.

The classification method proposed in this work is described as:

1) Estimate the first k robust PCs, as well as their variances, λ_j, $j = 1, \ldots, k$, using the p informative variables selected from the training set;
2) Calculate the Score Distance, $\mathrm{SD}(\boldsymbol{x}_i)$, and the Orthogonal Distance, $\mathrm{OD}(\boldsymbol{x}_i)$, $i = 1, \ldots, n$, of the training set observations projected into the space spanned by the first k PCs determined in 1.
3) Calculate the thresholds for the Score and Orthogonal Distances, $\mathrm{c_{OD}}$ and $\mathrm{c_{SD}}$;
4) Classify a new observation $\boldsymbol{x}_0$ as an anomaly if $\mathrm{SD}(\boldsymbol{x}_0) > \mathrm{c_{SD}}$ or $\mathrm{OD}(\boldsymbol{x}_0) > \mathrm{c_{OD}}$.

The number k of principal components is chosen in order to maximize the estimate of the Recall (probability that an observation is classified as anomaly when in fact it is an anomaly) obtained for the classification of the observations of the training set.

V. Results

A. Dataset, traffic object definition and features

In order to obtain a perfect ground-truth we arranged a small private laboratory network, with a set of PCs interconnected through a switch, and Internet access provided through a (strong) router/firewall. There were 17 PCs, 10 for users producing licit traffic, one for the server, one for measurements, and 5 assigned to the attacks.

Multiple measures were adopted to assure that the licit traffic objects were free of anomalies. We configured the PCs with a minimal Linux Ubuntu distribution running from live CDs to assure that no virus, worm, Trojan horse, or active Botnet were present. We also configured the forwarding table of the switch with static entries only, and disabled its learning capability, to prevent any local spoofing attack. Note that the IP and MAC addresses of the PCs are all known. Internet access is provided by router/firewall (Cisco ASA 5510), which performs traffic inspection to allow the entrance of external traffic only when in response to traffic generated from within the private network, and contains access lists configured to prevent remote IP spoofing attacks.

The licit traffic is a mixture of file sharing (BitTorrent), video streaming (IPTV over TCP) and Web browsing (HTTP), which are the predominant Internet applications. In order to generate this traffic, we asked a set of 10 users to use these applications from inside our lab network following their normal behavior. The users were restricted to an (access) list of (safe) Web sites, configured in the router/firewall, to avoid the possibility of becoming infected while navigating.

The attacks were produced only within the lab network and only to and from dedicated PCs. We concentrated on two broad classes of attacks, tightly related with Botnets, currently the main security threat in the Internet: port-scans and snapshots. Port-scans are usually the first activity of an infected Bot and are used to identify other targets vulnerable to infection and Botnet spreading. Snapshots, performed in the second phase of the attack, are a type of identity theft aimed at stealing personal information. Other classes of Botnet attacks are: (i) infection by spyware, malware and adware, (ii) denial of service and (iii) E-mail spam. In this work, we do not focus on the later attacks since they happen uniquely at the host level or can be detected and mitigated by traditional network mechanisms (e.g. host and server anti-virus, monitoring and rate control at routers/firewalls, and e-mail authentication and scanning). In our setup, the port-scans were produced by NMAP, with one second interval between SYN probes. Snapshots were emulated by sending small files, from FTP clients installed at the attacking PCs to an FTP server. The files had 120 Kbytes, to simulate a small screen area of 335x180 pixels around the cursor, and were sent with inter-arrival times following an exponential distribution with mean 120 seconds, to simulate the user clicks.

The dataset was collected by mirroring all traffic passing through the switch to a PC dedicated to measurements. The measurements were performed on March 2, 2011, for approximately 8 hours (some users were not always active). We captured the first 64 bytes of all packets.

The packets were aggregated in traffic objects that we call *datastreams*. A datastream aggregates all packets observed in a 5 minutes interval that have the same IP source address and one of the TCP port numbers equal. This object definition is not typical, but we claim it is very well suited to the detection of Internet attacks (and eventually to other machine learning applications, e.g. identification of Internet applications). It aggregates on a single object all traffic of one application that enters or leaves one machine. In this way, the traffic of applications and attacks that open several TCP sessions, e.g. HTTP, BitTorrent, and port-scans, is placed in the same traffic object, which would not occur with (more common) 5 tuple definition (source and destination IP address, source and destination port number, and protocol type). In this way, the

TABLE II
PERCENTAGE OF DATASTREAMS IN EACH SCENARIO

		%				
Scenario	n	P-Scan	Snap	HTTP	Stream	BitT
B1-Train	59	1.7	5.1	93.2	0	0
B1-Test	34	0.0	5.9	94.1	0	0
B2-Train	59	8.5	18.6	72.9	0	0
B2-Test	34	2.9	14.7	82.4	0	0
B3-Train	59	0.0	33.9	66.1	0	0
B3-Test	34	8.8	23.5	67.7	0	0
R1-Train	120	0.8	0.8	33.4	20.0	45.0
R1-Test	63	1.6	3.2	30.1	11.1	54.0
R2-Train	103	1.0	5.8	32.0	13.6	47.6
R2-Test	62	8.1	11.3	27.4	12.9	40.3
R3-Train	95	4.2	17.9	28.4	12.6	36.9
R3-Test	58	3.5	24.1	24.1	12.1	36.2

datastreams are able to capture user and high-level application behavior, which in many cases has a better discriminating power.

The capture file was processed by tshark to extract the datastreams and obtain 5 traffic characteristics computed in 0.1 seconds intervals: number of upstream packets (PUp), number of downstream packets (PDw), number of upstream bytes (BUp), number of downstream bytes (BDw), and number of active TCP sessions (Ses). Then, for each characteristic we computed 8 summary statistics: minimum (min), 1st quartile (Q_1), median (med), mean (m), 3rd quartile (Q_3), maximum (max), standard deviation (sd) and median absolute deviation (MAD). This gives a total of 40 features for each datastream, which is the starting point of our feature selection algorithm.

B. Performance evaluation

In order to evaluate the performance of our method we have considered two types of network scenarios and 4 distinct customer usage profiles. The network scenarios are Business (B) and Residential (R). The customer usage profiles are (a) soft browsing (HTTP only), (b) file sharing machine (BitTorrent only), (c) file sharing user (BitTorrent and HTTP), and (d) heavy user (HTTP, BitTorrent and streaming). For the Residential scenario, we considered that 30% of users were of type (b), 40% of type (c), and 30% of type (d). For the Business one, we considered that 100% of users were of type (a). We have considered 3 attack intensities, described in terms of total datastream percentages: (1) 6% (5% snapshot and 1% port-scan), (2) 20% (15% snapshot and 5% port-scan), and (3) 35% (30% snapshot and 5% port-scan). Each case study will be denoted by the network scenario (B or R) followed by the attack intensity (1, 2, or 3), e.g., B1 refers to an atack of intensity 1 in a Business scenario.

Finally, we have separated the datastreams in two sets, one used for training and the other for testing, with 60% of the users assigned to the training set and 40% to the testing one. Table II indicates the number of datastreams of each type used for training and testing. In [21] we report a larger number of cases.

TABLE III
PERFORMANCE MEASURES

			Performance measures		
Scenario	Detector	Nr Ftrs	Recall	FPR	Precision
B1	$\emptyset$-NR	-	1	0.063	0.500
	$\emptyset$-R	-	1	0.063	0.500
	NR-NR	17	1	0	1
	NR-R	17	1	0	1
	R-NR	5	1	0	1
	R-R	5	1	0	1
B2	$\emptyset$-NR	-	0.167	0	1
	$\emptyset$-R	-	1	0.679	0.240
	NR-NR	13	0.167	0	1
	NR-R	13	1	0	1
	R-NR	7	0.167	0	1
	R-R	7	1	0	1
B3	$\emptyset$-NR	-	0.273	0	1
	$\emptyset$-R	-	0.818	0.783	0.333
	NR-NR	18	0.273	0	1
	NR-R	18	0.454	0.522	0.294
	R-NR	7	0.273	0.044	0.750
	R-R	7	1	0	1
R1	$\emptyset$-NR	-	1	0.033	0.600
	$\emptyset$-R	-	1	0.100	0.333
	NR-NR	15	1	0.067	0.429
	NR-R	15	1	0.100	0.333
	R-NR	10	1	0	1
	R-R	10	1	0.017	0.750
R2	$\emptyset$-NR	-	0.417	0	1
	$\emptyset$-R	-	1	0.600	0.286
	NR-NR	5	1	0.020	0.923
	NR-R	5	1	0.040	0.857
	R-NR	10	1	0	1
	R-R	10	1	0.040	0.857
R3	$\emptyset$-NR	-	0.125	0	1
	$\emptyset$-R	-	0.875	0.786	0.298
	NR-NR	6	0.125	0	1
	NR-R	6	0.688	0.476	0.355
	R-NR	7	0.125	0	1
	R-R	7	1	0	1

Table III summarizes our results. We have compared 6 different types of anomaly detectors, with and without feature selection, and with and without robustification of the feature selection and outlier detection methods. We denote each detector by A-B, where A represents the feature selection method, which can be of type R (robust), NR (non-robust) or $\emptyset$ (no method), and B represents the outlier detection method, which can be of type R or NR. For example, an anomaly detector using robust feature selection and non-robust outlier detection will be represented by R-NR, and the same detector but with no feature selection, will be represented by $\emptyset$-NR.

The non-robust version of the feature selection algorithm just skips the robustification of the mutual information estimator proposed in section III, but keeps the procedure for automatic selection of the relevant features. The non-robust version of the outlier detection method uses the classical PCA algorithm described in section IV-A. Both classical and robust PCA use the same procedure to compute the threshold values as described in section IV-B.

For each case in Table III, we have computed the number of selected features (Nr Ftrs), Recall, the probability that an observation is classified as anomaly when in fact it is an anomaly, False Positive Rate (FPR), the probability that an observation is classified as anomaly, when in fact it is a regular observation, and Precision, the probability of having an anomalous observation given that it is classified as an anomaly.

The first observation is that the R-R detector achieved the best results in most cases: the Recall is always 1, in B1, B2, B3, and R3 the performance is the maximum possible, and in the remaining scenarios FPR and Precision are close to their optimal. In most cases, the improvements over the non-robust versions are very high. For example, in the B2, B3, and R3 the Recall improved from 0.167, 0.273, and 0.125 to 1, without degrading the FPR and Precision. Such low values of Recall mean that, without robust methods, a large percentage of anomalies are not correctly identified. The benefits of employing feature selection and using robust statistics in both the feature selection and the outlier detection methods become clear from these results. Note that the Recall is high in many cases since the number of principal components was selected in order to maximize this parameter.

To analyze the performance gains associated with the addition of the feature selection method we concentrate on the 3 cases using a robust outlier detection, i.e., $\emptyset$-R, NR-R, and R-R. We leave out the cases where outlier detection is non-robust since even a few outliers can damage their performance, making it hard to interpret the results. It can be seen that robust feature selection (R-R) always achieves the best results. Regarding the choice between non-robust (NR-R) and no ($\emptyset$-R) feature selection, there are two cases, B3 and R3, where the later achieves better results. Indeed, non-robust feature selection may be undesirable, due to the possibility of having features with an important discriminating power removed by selection algorithm.

We also note that feature selection achieved a significant reduction in the number of features, which in some cases reached 87.5%, e.g., in NR-R detector of R2.

Table IV shows the features selected by the robust algorithm. In bold, we highlight the features that are common to the 3 attack intensities of each scenario. These are: in the Business scenario, mean, max, and Q_3 of active TCP sessions, and mean of bytes down; in the Residential scenario, sd, MAD, max, and Q_3 of active TCP sessions, and sd and max of bytes up. The first observation from these results is that the features associated with the number of active TCP sessions are predominant. This is not surprising since several applications and attacks differ considerably on the number of TCP sessions they open or attempt to open: port-scans attempt to open a very large number of TCP sessions, snapshots and streaming open only one session, while HTTP and BitTorrent can open many sessions, but much less than port-scans. Note also that bytes down is selected in the Business scenario and bytes up in the Residential one. This is essentially due to the fact that, in the Business scenario, the licit traffic is only HTTP, which mainly produces download data, while the Residential scenario also includes BitTorrent, which generates a large amount of upload data.

TABLE IV
FEATURES SELECTED BY ROBUST ALGORITHM

Scenario	Nr Ftrs	Features
B1	5	**mean(BDw)**, **mean(Ses)**, **Q_3(Ses)**, **max(Ses)**, sd(BDw),
B2	7	**mean(BDw)**, **mean(Ses)**, **Q_3(Ses)**, **max(Ses)**, sd(BUp), median(Ses), MAD(Ses)
B3	7	**mean(BDw)**, **mean(Ses)**, **Q_3(Ses)**, **max(Ses)**, sd(BUp), median(Ses), MAD(Ses)
R1	10	**max(BUp)**, **sd(BUp)**, **Q_3(Ses)**, **max(Ses)**, **sd(Ses)**, **MAD(Ses)**, mean(BDw), Q_3(BDw), max(BDw), sd(BDw),
R2	10	**max(BUp)**, **sd(BUp)**, **Q_3(Ses)**, **max(Ses)**, **sd(Ses)**, **MAD(Ses)**, Q_3(BDw), sd(BDw), Q_3(BUp), mean(Ses),
R3	7	**max(BUp)**, **sd(BUp)**, **Q_3(Ses)**, **max(Ses)**, **sd(Ses)**, **MAD(Ses)**, mean(BDw)

We also observe that the set of features varied a lot between the Residential and Business scenarios. Only two of them are the same: Q_3 and max of active TCP sessions. This again is of no surprise since the licit traffic differs significantly between the two scenarios. But the implication of this result is very important. It shows that the anomaly detector must be adaptive to different traffic conditions, and again that feature selection plays a key role.

In Fig. 3 we represent the score versus orthogonal distance, called diagnostic plot, for the dataset associated with the R3 case, when the anomaly detectors are R-NR and R-R. The diagnostic plots also include the threshold lines: any point with a score or orthogonal distance larger than a threshold is considered an anomaly. Fig. 3 (bottom) shows that only the actual anomalies are classified as anomalies, in the case of R-R. However, in the NR-R case, there is a lot of confusion around snapshots, as can be seen in Fig. 3 (top): all snapshots are classified as licit traffic, which explains the poor Recall value (0.125); all licit traffic is classified as such, which explains the FPR of 0 and Precision of 1. In these cases, while the selected features are the same (both detectors use robust feature selection), the proximity in behavior between snapshots and some HTTP and BitTorrent interactions fools the non-robust outlier detector. Indeed, snapshots correspond to small file uploads, which can be also observed in BitTorrent, when uploading files to nodes with low available bandwidth using small data chunks, and in HTTP, when sending consecutive e-mails or chat messages using a Web interface.

VI. CONCLUSIONS

This paper proposed a novel anomaly detector for Internet traffic that has two fundamental characteristics. First, it includes feature selection as a preprocessing step. Second, it uses robust statistics in both the feature selection algorithm and the outlier detection method. Feature selection includes a novel procedure to determine automatically the set of relevant features. In order to evaluate the detector, we devised a network

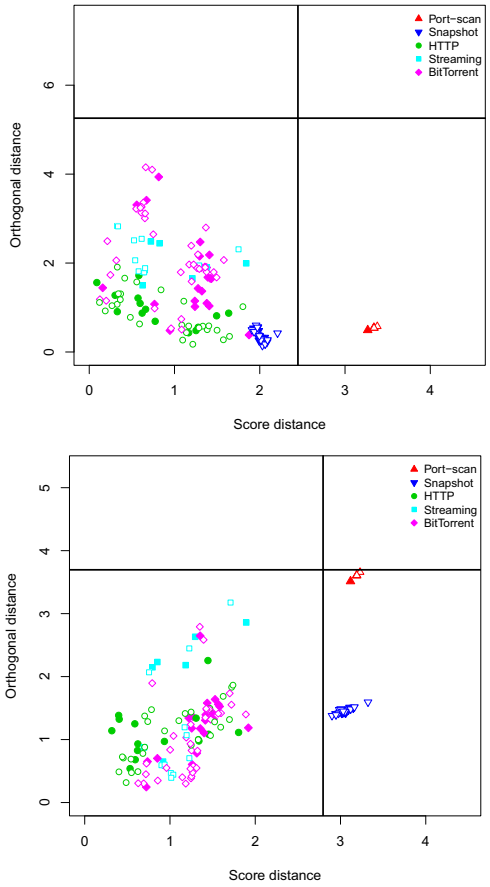

Fig. 3. Diagnostic plots, Residential scenario with attack intensity 3 (R3), and anomaly detectors R-NR (top) and R-R (bottom).

setup to obtain a perfect ground-truth. Our results show that the combination of robust feature selection and outlier detection based on robust PCA achieves very high performance and is adaptive to different traffic conditions. Moreover, robust feature selection induces significant performance gains, and is an essential preprocessing step for anomaly detection.

ACKNOWLEDGMENT

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) through projects PTDC/EIA-EIA/115988/2009 and PTDC/EEA-TEL/101880/2008. Cláudia Pascoal also acknowledges the support of FCT via PhD grant SFRH/BD/42547/2007.

REFERENCES

[1] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. of the IEEE Found. and New Direc. of Data Mining*, 2003, pp. 172–179.

[2] W. Hu, Y. Liao, and V. Vemuri, "Robust anomaly detection using support vector machines," in *Proc. of the Int. Conf. on Machine Learning*, 2003.

[3] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *Perform. Eval. Rev.*, vol. 32, pp. 61–72, 2004.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, 2009.

[5] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: Problems and solutions," in *INFOCOM*, 2009, pp. 2866–2870.

[6] B. Rubinstein, B. Nelson, L. Huang, A. Joseph, S. Lau, S. Rao, N. Taft, and J. Tygar, "ANTIDOTE: understanding and defending against poisoning of anomaly detectors," in *Proc. of the* 9^{th} *ACM SIGCOMM Internet Measurement Conference*, 2009, pp. 1–14.

[7] K. Nyalkalkar, S. Sinha, M. Bailey, and F. Jahanian, "A Comparative Study of Two Network-based Anomaly Detection Methods," in *(mini-conference at) INFOCOM*, Shanghai, China, April 2011.

[8] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.

[9] R. Maronna, R. Martin, and V. Yohai, *Robust statistics: theory and methods*. J. Wiley, 2006.

[10] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proc. of the 2008 ACM CoNEXT Conference*, 2008, pp. 11:1–11:12.

[11] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *Perform. Eval. Rev.*, vol. 33, pp. 50–60, 2005.

[12] A. Este, F. Gringoli, and L. Salgarelli, "On the stability of the information carried by traffic flow features at the packet level," *Comput. Commun. Rev.*, vol. 39, pp. 13–18, 2009.

[13] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, pp. 109–120, 2007.

[14] H. Ringberg, M. Roughan, and J. Rexford, "The need for simulation in evaluating anomaly detectors," *Comput. Commun. Rev.*, vol. 38, pp. 55–59, 2008.

[15] C. Croux, P. Filzmoser, and M. R. Oliveira, "Algorithms for projection-pursuit robust principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 87, pp. 218–225, 2007.

[16] M. Hubert, P. Rousseeuw, and K. Branden, "ROBPCA: a new approach to robust principal component analysis," *Technometrics*, vol. 47, pp. 64–79, 2005.

[17] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Trans. on Networking*, vol. 9, 2001.

[18] P. Filzmoser, R. Maronna, and M. Werner, "Outlier identification in high dimensions," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1694–1711, 2008.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[20] J. Walters-Williams and Y. Li, "Estimation of mutual information: A survey," in *Proc. of the* 4^{th} *Int. Conf. on Rough Sets and Knowledge Technology*, 2009, pp. 389–396.

[21] C. Pascoal, "Robust feature selection and robust PCA for Internet traffic anomaly detection," UTL/IST and CEMAT, Technical Report, 2011.

[22] M. Hubert, P. Rousseeuw, and T. Verdonck, "Robust PCA for skewed data and its outlier map," *Computational Statistics and Data Analysis*, vol. 53, pp. 2264–2274, 2009.

[23] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Int. Conf. on Machine Learning*, 2003, pp. 856–863.

[24] W. Gomez, L. Leija, and A. Diaz-Perez, "Mutual information and intrinsic dimensionality for feature selection," in *Proc. of the* 7^{th} *Int. Conf. on Elec. Eng. Comp. Sci. and Aut. Control*, 2010, pp. 339 – 344.

[25] C. Lai, M. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recogn. Lett.*, vol. 27, pp. 1067–1076, 2006.

[26] P. Filzmoser and H. Fritz, "Exploring high-dimensional data with robust principal components," in *Proc. of the* 8^{th} *International Conference on Computer Data Analysis and Modeling*, vol. 1, 2007, pp. 43–50.

[27] I. Jolliffe, *Principal Component Analysis*, 2^{nd} ed. Springer Verlag, 2002.

[28] M. Hubert and S. Engelen, "Robust PCA and classification in biosciences," *Bioinformatics*, vol. 20, pp. 1728–1736, 2004.

[29] C. Croux and A. Ruiz-Gazen, "High breakdown estimators for principal components: the projection-pursuit approach revisited," *Journal of Multivariate Analysis*, vol. 95, pp. 206–226, 2005.

[30] C. Pascoal, M. R. Oliveira, A. Pacheco, and R. Valadas, "Detection of outliers using robust principal component analysis: A simulation study," in *Combining Soft Computing and Statistical Methods in Data Analysis*, vol. 77. Springer-Verlag, 2010, pp. 499–507.