PCA filtering and Probabilistic SOM for Network Intrusion Detection

Eduardo De la Hoz, Emiro De La Hoz

Programa de Ingeniería de Sistemas, Universidad de la Costa. Barranquilla, Colombia

Andrés Ortiz*

Communications Engineering Department, University of Málaga

Julio Ortega, Beatriz Prieto

Computer Architecture and Technology Department. CITIC. University of Granada

Abstract

The growth of the internet and, consequently, the number of interconnected computers, has exposed significant amounts of information to intruders and attackers. Firewalls aim to detect violations according to a predefined rule-set and usually block potentially dangerous incoming traffic. However, with the evolution of attack techniques, it is more difficult to distinguish anomalies from normal traffic. Different detection approaches have been proposed, including the use of machine learning techniques based on neural models such as Self-Organizing Maps (SOM). In this paper, we present a classification approach that hybridizes statistical techniques and SOM for network anomaly detection. Thus, while Principal Component Analysis (PCA) and Fisher Discriminant Ratio (FDR) have been considered for feature selection and noise removal, Probabilistic Self-Organizing Maps (PSOM) aim to model the feature space and enable distinguishing between normal and anomalous connections. The detection capabilities of the proposed system can be modified without retraining the map, but only by modifying the units activation probabilities. This deals with fast implementations of Intrusion Detection Systems (IDS) necessary to cope with current link bandwidths.

Keywords: Probabilistic SOM, Bayesian SOM, IDS, Self-Organizing Maps, PCA filtering

1. Introduction

Nowadays, with the growth of Internet, not only the number of interconnected computers, but also the relevance of network applications, has increased considerably. At the same time, the trend to online services has exposed sensitive information to intruders and attackers [16, 3]. Common protection approaches do not react to attackers or intruders, but only suppose a passive position to reduce

*Corresponding Author. Tel: +34 952133353

Email addresses: edelahoz6@cuc.edu.co (Eduardo De la Hoz), edelahoz@cuc.edu.co (Emiro De La Hoz), aortiz@ic.uma.es (Andrés Ortiz*), jortega@ugr.es (Julio Ortega), beap@ugr.es (Beatriz Prieto)

exposure. On the other hand, the complexity of the newer attacks necessitates the use of elaborated techniques, such as pattern classification or artificial intelligence, for successfully detecting an attack or just to differentiate among normal and anomalous traffic. Other approaches that implement an active protection against real or potential attackers include firewall-like systems capable of inspecting data packets. IDS and Intrusion Protection Systems (IPS) are active systems that continuously monitor the network. These systems calculate some features from the monitored network in order to classify the traffic, detect abnormal behaviours and react according to predefined rules. This presents a classification problem, with some requirements needing specific approaches, thus contributing to the machine learning field.
There are two IDS design approaches [3, 15, 27, 17] depending on the detection philosophy. The first is the so-called *signature-based* IDS [27], which analyzes all the incoming packets looking for known patterns associated with intrusion attempts. These patterns are stored in a database and can be compared with patterns extracted from incoming network traffic. In other words, signature-based IDS works similarly to virus scanners as they also compare observed behaviours with stored ones. However, this method is not able to detect attacks whose signature is not in the database. Similarly, outdated databases or deficient signatures may cause false negatives or false positives (that is, missing an actual attack or misreading legitimate traffic as an attacker). The second searches for deviations from normal patterns to decide whether a connection is classified as anomalous, namely *anomaly-based* [17]. These systems usually characterize normal patterns by means of statistical learning techniques applied to network traffic. In addition, using complex features allows discovering not only an actual intrusion, but also a potential one, namely *anomaly-based* IDS. Nevertheless, in *anomaly-based* systems, sufficiently accurate models are necessary to distinguish normal from abnormal patterns. Otherwise, the IDS is likely to generate too many false positives or negatives. *Anomaly-based* IDS can be addressed, for instance, by discovering a misuse of the protocol flags or an abnormal number of certain events (such as the number of TCP connection attempts). Nevertheless, due to attack diversity, it is necessary to compute more complex features to improve detection.

In the last years, different intrusion detection approaches have been proposed, including the use of artificial intelligence techniques, such as neural networks [56].

As explained in the next section, anomaly detection is not a straightforward task in a real environment [42, 29, 49, 15] and poses interesting problems related to classification and feature selection.
This way, datasets such as KDD99 (and also the NSL-KDD) have been built to provide training and test subsets with different statistical distributions as expected in real anomaly detection tasks. As the KDD99 is a large-volume dataset, researchers usually take random samples for their experiments that explain the discrepancies observed in the literature [59]. In [37], an analysis is provided of the low level of industry adoption of intrusion detection procedures proposed in the academic literature, despite their reported high performance. Among those reasons, incorrect feature selection procedures and statistical analysis, as well as not having enough detailed experiments, are considered. In this paper, the IDS concerns expressed in [37] have been taken into account and addressed.

In this work, principal component analysis (PCA) is used to generate a new set

of non-correlated features in order to remove noise and to avoid using low variance variables (that is, almost single-valued variables). Moreover, these new features are selected according to their discriminative capability. Subsequently, feature space modelling and classification is addressed by means of Probabilistic SOM, a fuzzy version of classical SOM that allows measuring the activation probability of each unit. Nevertheless, detecting not only an attack but also the type is not a straightforward task, and previous approaches have not been able to obtain high per attack detection accuracy values [42, 29].

The rest of the paper describes the databases and methods used in this work, and the experimental results. Specifically, Section 2 is split into three subsections, feature filtering and selection procedures (Section 2.1 and 2.2), and the use of probabilistic SOM for modelling and classification (Section 2.3). Then Section 3 describes the dataset used for the experiments and the feature selection accomplished to distinguish between anomalies and normal traffic. Finally, Section 4 analyzes the previous work done in this line and section 5 summarizes the conclusions of the paper.

2. Proposed methods

The proposed feature selection and classification method for anomaly detection is presented in this section, which has been split into three subsections that summarize our approach. Figure 1 shows its corresponding block diagram

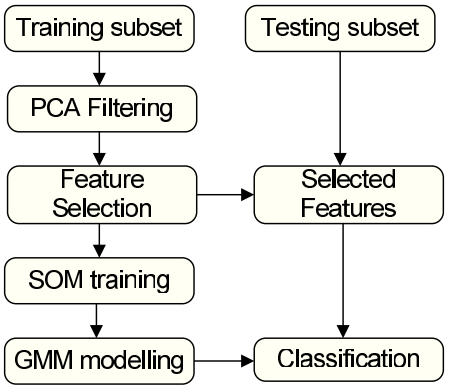

Figure 1: Block diagram of the proposed anomaly detection system

2.1. Feature generation and PCA filtering

Feature selection is a key step in classification problems as it contributes to removing redundant or irrelevant input features not only to reduce computing times for learning, but also to improve classifier accuracy [59]. The methods for feature selection can be classified into filter, wrapper and hybrid methods. Filter methods select the feature subset as a pre-processing step according to a chosen criterion and without taking into account the performance of the classifier. Thus, filter methods are usually less computational than expensive wrapper methods that use the classification outcomes to evaluate the feature selection methods. Although wrapper methods usually outperform filter methods with respect to classifier accuracy, the results obtained are usually not applicable whenever the classifier is changed. Thus, there are proposals (that is, hybrid methods) that

combine a wrapper method with a filter that guides the classifier. The approach considered in this paper can be included in the filter methods.

Principal Component Analysis (PCA) has been widely used in many applications for extracting the most relevant dataset information. In fact, it has been successfully used in face recognition applications [52]. In this case, PCA is used to derive a new set of uncorrelated features from a set of correlated ones. Thus, PCA generates a set of orthogonal basis vectors so that the data can be expressed as a linear combination of that basis. Some papers [14] have claimed that this method presents some classification task problems as it requires more processing whenever new data is added and it is not invariant under a transformation of the data.
The procedure can be explained as follows. Let $X = \{\mathbf{x}_1, ..., \mathbf{x}_{N_t}\}, \mathbf{x_i} = (x_i^1, ..., x_i^n)^T$ be the input data samples (training samples). A shifted version of the data manifold can be obtained by substracting the mean ($\overline{X}$), $Y = X - \overline{X}$, where $\mathbf{y}_i \in \mathbb{R}^n$, $\mathbf{y}_i = (y_i^1, ..., y_i^n)^T$, $i = 1, ..., N_t$. PCA searches for N_t orthonormal vectors $\mathbf{u}_k = (u_k^1, ..., u_k^n), k = 1, ..., N_t$ such that

$$\lambda_k = \frac{1}{M} \sum_{r=1}^{N_t} (\mathbf{u}_k^T \mathbf{y}_r)^2 \tag{1}$$

is maximum. Vectors $\mathbf{u}_k, k = 1, ..., N_t$ verify that $\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk}$ (δ_{lk} is the Kronecker delta). Vectors $\mathbf{u}_k$ and scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix computed as $C = YY^T$. It is worth noting the difference between the presented method and the *eigenconnections* approach [5], inspired by the face recognition method using eigenvectors (*eigenfaces*) [52] due to its appearance. In this case, eigenvectors are used to generate a new feature space that allows us to remove noise comprising the discriminative information in a reduced number of features. Thus, the training data samples are projected onto the space spanned by the *eigenvectors* to generate a set of uncorrelated features that best describe the data manifold. These features are further used to train the SOM-based classifier described in this section. In order to classify a new data instance $\mathbf{v}$, it has to be projected onto the *eigenvectors* space, obtaining its corresponding feature vector

$$\boldsymbol{\omega}_k = (\mathbf{v} - \overline{X}) * \mathbf{u}_k \tag{2}$$

On the other hand, the original data can be reconstructed from the principal components as

$$\mathbf{v}_k^{rec} = \overline{X} + \boldsymbol{\omega}_k * \mathbf{u}_k^T \tag{3}$$

where v_k^{rec} is the reconstruction of $\mathbf{v}$ using the eigenvector k. Although in the problem considered in this work, the two first principal components account for more than 95 percent of the variance, it does not ensure the discriminative capability of the projections. Thus, selected eigenvectors are sorted by their discriminative power according to their Fisher Discriminant Ratio (FDR) value, as is explained in what follows.

2.2. Fisher Discriminant Ratio for eigenvectors selection

In many problems (such as in NSL-KDD classification), the data set is noisy and contains columns with single or almost single values. Moreover, other columns containing not single values are not discriminative enough, contain redundant or not relevant information and result in misclassification. For this reason, a pre-processing stage to filter the input data is necessary [23, 8]. Although PCA sorts the principal components by its associated variance (eigenvalue), it is not guaranteed that eigenvectors with higher eigenvalues are the most discriminative ones. This can be solved by selecting the principal components according to the FDR criterion using the following method:

Let $X = \{\mathbf{x}_1, ..., \mathbf{x}_{N_t}\}$ be the training samples, Y their shifted versions and $U = \{\mathbf{u}_1, ..., \mathbf{u}_n\}$ the eigenvectors matrix. A new training dataset can be derived from X by projecting it into each eigenvector. Thus, defining

$$\boldsymbol{\psi}_i = \mathbf{u}_i^T * X \tag{4}$$

each column Ψ_i in $\boldsymbol{\Psi} = \{\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_n\}$ corresponds to the projection of X into the eigenvector $\mathbf{u}_i$. As PCA sorts the eigenvectors according to a decreasing variance order, first eigenvectors comprise the most part of the variance. However, it does not ensure a good enough class separation capability. This way, eigenvectors are ordered according to their FDR value.

$$FDR = \sum_{i}^{M} \sum_{j \neq i}^{M} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \tag{5}$$

for the multi-class case, where σ_i, μ_i are the variance and mean for the class i, respectively. Equation 5 can be simplified to

$$FDR = \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \tag{6}$$

for the 2 class separation case.

Our purpose is to remove noise from the dataset while using only the eigenvectors with maximum class separation capabilities. This is achieved by selecting projections that provide the r lower FDR values, and subtracting the reconstructing of the samples using these eigenvectors from the original dataset, as indicated in equation 7.

$$\hat{X} = X - \sum_{i=1}^{r} \mathbf{u}_i \boldsymbol{\psi}_i \tag{7}$$

Algorithm 1: PCA-FDR feature selection method

1. Subtract the mean from the training samples: $Y = X - \overline{X}$.
2. Compute the principal components u_i^T verifying equation 12 (Section 2.3).
3. Compute the projection ψ_i of the training samples onto the eigenvectors according to equation 4.
4. Sort the eigenvectors according to the discriminative power of the projections according to their FDR value.
5. Subtract the projection of the training samples onto the eigenvectors with lower FDR values from the data manifold as indicated in equation 7.

The method described above has been summarized in the pseudocode shown in algorithm 1. A variant of this method has been used in [31, 2]. The filtered dataset

$\hat{X}$ contains the same number of features as X, but their noise corresponds to the least discriminant eigenvectors. Subsequently, the filtered dataset is projected into the k eigenvectors with the higher FDR value (that is, the most discriminative eigenvectors), thus reducing the dataset dimension to k. This way, the described method allows selecting the k most discriminative eigenvectors and, subsequently, computing the features from attacks that enable their adequate classification.

2.3. Classification using Bayesian SOM

SOM [26] is one of the most popular neural network models for unsupervised learning. SOM groups similar data instances into a 2D or 3D lattice, namely, an output map. On the other hand, different data instances will be apart in the output map. Moreover, some important input space properties can be inferred from that output map [20].

1. *Input space modelling.* The prototypes $\boldsymbol{\omega}_i$ (where (i) refers to the unit index in the map) computed during the SOM training provide an approximation of the input space.
2. *Topological order.* Units on the output map are arranged into a 2D or 3D lattice, and their position depends on the specific input space features. This way, the index of a unit could be expressed as tuples (i_1, i_2) or (i_1, i_2, i_3), respectively for 2D or 3D lattices.
3. *Feature selection.* The SOM algorithm produces a number of prototypes from the input data space. Thus, the algorithm not only reduces the dimension, but also the input space size, as it is represented by the prototype vectors.

The SOM algorithm is briefly explained in what follows. Let $\mathrm{X} \in \mathbb{R}^n$ be a n-dimensional data manifold. The SOM map is composed of d units, each represented by an n-dimensional model vector $\boldsymbol{\omega}_i$. For each input data instance $\mathbf{v}$, the Best Matching Unit (BMU) is defined as the unit $\boldsymbol{\omega}_i$ closest to $\mathbf{v}$:

$$\|\boldsymbol{\omega}_i - \mathbf{v}\| \leq \|\boldsymbol{\omega}_j - \mathbf{v}\|, \forall \mathbf{v} \in X, i \neq j \tag{8}$$

where $\|\cdot\|$ is the Euclidean distance and X is the training dataset. Once the BMU is determined for the current iteration, the model vectors are updated according to the rule

$$\boldsymbol{\omega}_i(t+1) = \boldsymbol{\omega}_i(t) + \alpha(t) h_i(t)(\mathbf{v} - \boldsymbol{\omega}_i(t))^1 \tag{9}$$

where $\alpha(t)$ is the learning rate and $h_i(t)$ is a function that defines the neighbourhood around the BMU $\boldsymbol{\omega}_i$. Usually, $\alpha(t)$ diminishes following an exponential decay rule [26] and h_i is a Gaussian [26] hat whose width shrinks in time (iterations). In this work, the SOM has been initialized linearly as follows in order to avoid random effects [26]. Linear SOM prototype initialization aims to accommodate the training data eigenvalues and eigenvectors [26, 1, 53]. This initialisation method implies that the first dimension of the prototypes was arranged proportionally to the first principal component and that the second dimension was arranged proportionally to the second principal component. For example, in 2D maps, the initial values of the prototype vector with coordinates (i_1, i_2) on the output map

[1] Note that i is a linear index that identify the prototype vectors

were determined by equation 10, which defines the centroid of the subspace along the two principal input data components.

$$\boldsymbol{\omega}_i = \overline{X} + \frac{\sigma_1}{nd_1}\left(\mathbf{pc}_1\left(i_1 - \frac{nd_1}{2}\right) + \mathbf{pc}_2\left(i_2 - \frac{nd_2}{2}\right)\right)^{2} \tag{10}$$

In equation 10 $\overline{X}$ is the average vector of the input data along each dimension, σ_1 is the standard deviation of the first principal component, nd_1 and nd_2 are the number of units on the first and second map dimensions, respectively, and $\mathbf{pc_1}$ and $\mathbf{pc_2}$ are the first and second principal components of the input data, respectively.

Once the map is already trained, each prototype represents a set of input vectors (those input vectors that activate this vector as their BMU). In other words, SOM quantizes the data manifold in d n-dimensional prototypes or model vectors. This learning model activates a specific unit (that is, the corresponding BMU) whenever a new data instance is presented to the SOM.

However, it is possible to measure the response of the map units instead of calculating the BMU as the unit that is closest to the input data. This way, a probabilistic interpretation of the SOM could be obtained by modelling the prototypes using a Gaussian Mixture Model (GMM), which fuzzifies the SOM unit response [1]. The main goal behind using a GMM is to train the map only once, while a further tuning of the map response (that is, the classification results) can be achieved by modifying the prior activation probabilities of the SOM units such that they allow the activation level to be used to recognize the corresponding patterns associated to normal connections and network anomalies in this case. Thus, anomalies can be detected as they deviate from the normal activation patterns [45]. In this probabilistic SOM, the BMU is determined not only by computing the minimum distance to a given input vector, but also by taking into account the likelihood of an unit to be the BMU. The prior probability of each map unit i can be experimentally computed by taking into account the training set activation probability similarly to [1], as it is shown in equation 11

$$p_i = \frac{\#\widetilde{X_i}}{\#\widetilde{X}} \tag{11}$$

where $\#\widetilde{X}$ is the total number of input vectors and $\#\widetilde{X_i}$ is the number of vectors whose closest prototype is $\boldsymbol{\omega}_i$, as defined on equation 12 (Voronoi set of unit i). This is also called the *receptive field* of unit i [48].

$$\widetilde{X_i} = \{\mathbf{x} \in X \;/\; \|\mathbf{x} - \boldsymbol{\omega}_i\| \leq \|\mathbf{x} - \boldsymbol{\omega}_k\| \; k = 1, ...N, i \neq k\} \tag{12}$$

where X is the set of training patterns.

The GMM is built according to the equation 13 using N components (one for each SOM prototype), where the weights p_i for each Gaussian component corresponds to the prior probabilities computed in equation 11.

$$p(\mathbf{x}) = \sum_{i=1}^{N} p_i P_i(\mathbf{x}) \tag{13}$$

[2]Note that (i_1, i_2) is the subscript that identifies the prototype vector with index i in the SOM map)

In Equation 13, $P_i(\mathbf{x})$ is an $n-dimensional$ Gaussian distribution [1, 46] that serves as prototype vector and is computed as:

$$P_i(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n}|\Sigma_i|^{1/2}} e^{\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right)} \tag{14}$$

The mean of each individual Gaussian component $\boldsymbol{\mu_i}$, is the prototype vector of the corresponding unit itself, while the covariance matrix $\mathbf{\Sigma_i}$ for the i-component is given by the dispersion of the data samples around the model vector $\boldsymbol{\omega}_i$. In other words, each gaussian component models the distribution of the receptive field of the corresponding unit. Once the GMM model has been built, the response of the unit k for a given input $\mathbf{x}$ can be computed as the posterior probability by using the Bayes theorem.

$$p(\boldsymbol{\omega}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\omega}_k)p(\boldsymbol{\omega}_k)}{p(\mathbf{x})} \tag{15}$$

Consequently, in equation 15, $p(\boldsymbol{\omega}_k|\mathbf{x})$ represents the probability that a sample vector $\mathbf{x}$ belongs to class $\boldsymbol{\omega}_k$ (posterior probability), $p(\mathbf{x}|\boldsymbol{\omega}_k)$ is the probability density function of the prototype $\boldsymbol{\omega}_k$ computed from the GMM according to equation 13 (i.e. $P_k(x)$) and $p(\mathbf{x})$ is a normalization constant that makes the posterior probability density integrate equal to one and thus can be computed as:

$$p(\mathbf{x}) = \sum_{k=1}^{NC} p(\mathbf{x}|\boldsymbol{\omega}_k)p(\boldsymbol{\omega}_k) \tag{16}$$

where NC is the number of classes.

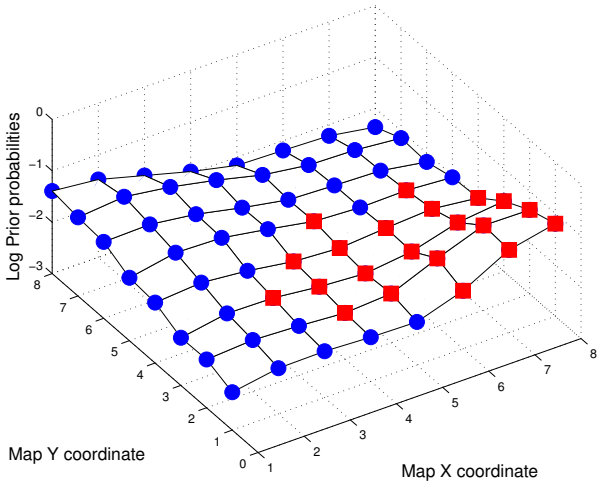

Figure 2: SOM *a priori* logarithmic activation probabilities for each unit computed according to equation 11, i.e. $log\frac{\#\widetilde{X}_i}{\#X}$. Circles and squares represent the label of each unit, corresponding to normal connections and network anomalies, respectively.

Figure 2 shows the *a priori* activation probabilities, p_i, for each map unit. In addition, normal connections are mapped in units represented as circles, while network anomalies are mapped in units represented as squares.

3. Experimental setup and Results

In this section, the database used to assess the proposed method and the experimental results obtained are presented. The issues vis--vis assuring correct experimental conclusions as described in [37] deal with: A detailed description of the dataset used as benchmark, including the training and testing sets that should be properly defined. A cross-validation performed by using the testing set to evaluate the IPS that has learnt by using the training set. The number of evaluation executions should guarantee statistical validity according to the central limit theorem. The evaluation metrics that should be considered. The use of the Receiver Operating Characteristic (ROC) curve alone, or detection rate and false positive rate could be incomplete [37], and the use of Area Under ROC Curve (AUC) in combination with ROC curve and accuracy would be interesting. The need for analyzing different types of attacks separately.

3.1. Database

In this paper, we use a revised version of the KDD99 dataset [37], which has been widely used in research works and contains about 4GB of compressed data from captures of *tcpdump* [30] in the DARPA98 IDS evaluation program [33]. This data corresponds to about seven weeks of network traffic, and three groups of features extracted for each connection. DARPA datasets [35] were generated in 1998, 1999 and 2000 in MIT Lincoln Laboratories, specifically for testing purposes. The sets consist of simulated host and network normal traffic and manually generated network-based attacks [49]. Training and testing subsets provided by the KDD99 dataset follow different probability distributions.

However, the KDD99 dataset has inherent representation problems due to the synthetic characteristic of the data [33, 37]. Thus, the Network Security Lab - Knowledge Discovery and Data Mining (NSL-KDD) dataset [38] was proposed to overcome most of the deficiencies of KDD99, as stated in [33]. Moreover, in NSL-KDD, redundant records were removed and the attacks were labelled and sorted by their level of detection difficulty. Then, as in the most recent works [22, 40, 39], we also use the NSL-KDD dataset. Moreover, in NSL-KDD, redundant records in KDD99 were removed and the attacks labelled and sorted by their level of detection difficulty. Taking into account all these characteristics, the NSL-KDD can be considered a good approximation to present known attacks. This way, NSL-KDD constitutes an adequate dataset to evaluate our procedure as the most recent IDS reference works [18, 40, 39, 55, 28].

As KDD'99, the NSL-KDD Dataset provides 41 features extracted from the network traffic to describe each connection. These features can be grouped into four types:

1. Basic features. These features summarize all the properties of a TCP/IP connection.
2. Traffic-based features. This class includes those features computed over a time interval (window) and contains information about the connections in which the destination port or the service remains the same after the corresponding time windows. In the KDD99, the time window is two seconds.
3. Content-based features. Since U2R or R2L attacks consist of repeatedly sending similar patterns on the packet payload, it is necessary to examine the contents of the packets to figure out these attacks.

Indeed, these features are used to classify the attacks enumerated in the introduction. Nevertheless, not all the features are discriminative enough and many are almost constant or even zero, thus necessitating devising filtering and selection techniques to improve classification outcomes.

3.1.1. Data preprocessing

Data pre-processing, which comprises encoding non-continuous variables and normalization, is an important stage that may determine the classification performance. Although this stage plays an important role in pre-processing, as it can determine the classification performance, few works pay sufficient attention to it [49]. As described in the preceding section, KDD99-based datasets consist of 41 features, which should be enough to characterize anomaly connections. These are classified into three groups: continuous, symbolic and binary features. As most classifiers only accept numeric values, the first issue is related to symbolic features. These are usually coded in several works by simply substituting each different feature with an integer [49, 18, 23]. Although this can be acceptable in many situations, it is not the best encoding solution for classifiers based on the Euclidean distance [12, 7]. This way, we adopt a different solution that maps each symbolic feature to an $\mathbb{R}^d$ subspace, where d is the number of possible values of the discrete variable. Although this solution increases the dimensionality of the data considerably (for instance, the *service* feature can take 65 different values), it is not critical for the classification method used in this work. Furthermore, dimensionality reduction techniques are used to compress relevant information with fewer features. Thus, a different value on these features contributes to $\sqrt{2}$ the distance measure [7]. This technique is used with *protocol* (three different values), *service* (65 different values) *port* (four different features). With the same philosophy, binary features are left *as is* in the feature space.

3.1.2. Data normalization

Data normalization ensures that all the features are in the same scale, in such a way that no feature contributes more than any other in the distance measure. There are different ways to normalize data [50, 3]. As previously explained, the KDD99 dataset contains numerical and categorical features. Thus, normalization has to be addressed differently in each case. Numeric and continuous variables are normalized to zero mean and unity variance using the equation 17

$$\hat{x} = \frac{x - \overline{x}}{\sigma} \tag{17}$$

where $\overline{x}$ and σ are the mean and the standard deviation of variable x, respectively. This is equivalent to expressing the variable x as the number of standard deviations away from the mean. However, categorical variables require a different treatment. Specifically, these variables need to be encoded before normalization according to the similarity measure [7]. Alternatives used in some works that use a simple category-to-numeric encoding fail when the similarity measurement is computed [49, 3]. For instance, the protocol feature cannot be encoded using consecutive numbers as it will state a similarity between different protocols (for example, TCP and UDP). For categories, we impose the following similarity measure: $s(x_k, y_k) = 1$ if $x_k = y_k$ and $s(x_k, y_k) = 0$ otherwise. This is addressed encoding these features as binary vectors, in which each component indicates the activation of

the corresponding feature (for example, whether the protocol is TCP or not in a specific connection). These features are not normalized.

3.2. Experimental results

The proposed method has been evaluated by using training and testing data provided by the NSL-KDD as separate subsets. Thus, it is not necessary to extract database subsets for cross-validation assessment; thus, the testing data does not participate in the training process in any way. Moreover, label information from the test set is only used for performance evaluation. Classification performance has been assessed by computing three statistical measures: sensitivity, specificity and accuracy. Sensitivity can be defined as the ability of the classifier to detect positive results (that is, the number of detected anomalies TP with respect to the total number of anomalies TP+FN) and it is described in equation 18. Specificity measures the ability to detect negative results (that is, the number of detected normal connections TN with respect to the total number of normal connections TN+FP) as described in equation 19. In addition, accuracy, as defined in equation 20, measures the percentage of samples correctly classified (the number of normal connections and anomalies TP+FN detected with respect to the number of received connections, either normal or anomalous TP+TN+FP+FN).

$$SENS = \frac{TP}{TP + FN} \tag{18}$$

$$SPEC = \frac{TN}{TN + FP} \tag{19}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

In equations 18, 19 and 20, TP, TN, FP and FN indicates *true positives, true negatives, false positives and false negatives*, respectively [50].
Two different experiments have been performed. The first one aims to determine the number of features that maximize classification performance. The results of these experiments for different feature selection and classification methods are shown in Figure 3. From these figures, the number of features providing the best performance in terms of sensitivity, specificity and accuracy can be figured out as the lowest number that allows a high enough performance level for each measure. The number of features indicated in Table 1 are specific features computed depending on the selection method. The optimum number of features can be easily obtained from the non-dominance concept applied over the classification outcomes [10], for instance, plotting accuracy vs. sensitivity. These plots for the discussed methods are shown in Figure 4, stating that maximum classification performance is achieved with the PSOM+PCA+FDR method in terms of accuracy and sensitivity, using the 15 features obtained by using the PSOM+PCA method. The PSOM+PCA+FDR method provides similar performance using eight features, but as fewer features implies high computational effectiveness, the PSOM+PCA+FDR method is considered to have better performance. Performance of these three methods in terms of accuracy, sensitivity and specificity can be statistically compared to show differences in the mean values. However, it is not possible to state clear superiority though hypothesis tests performed through ANOVA [36].

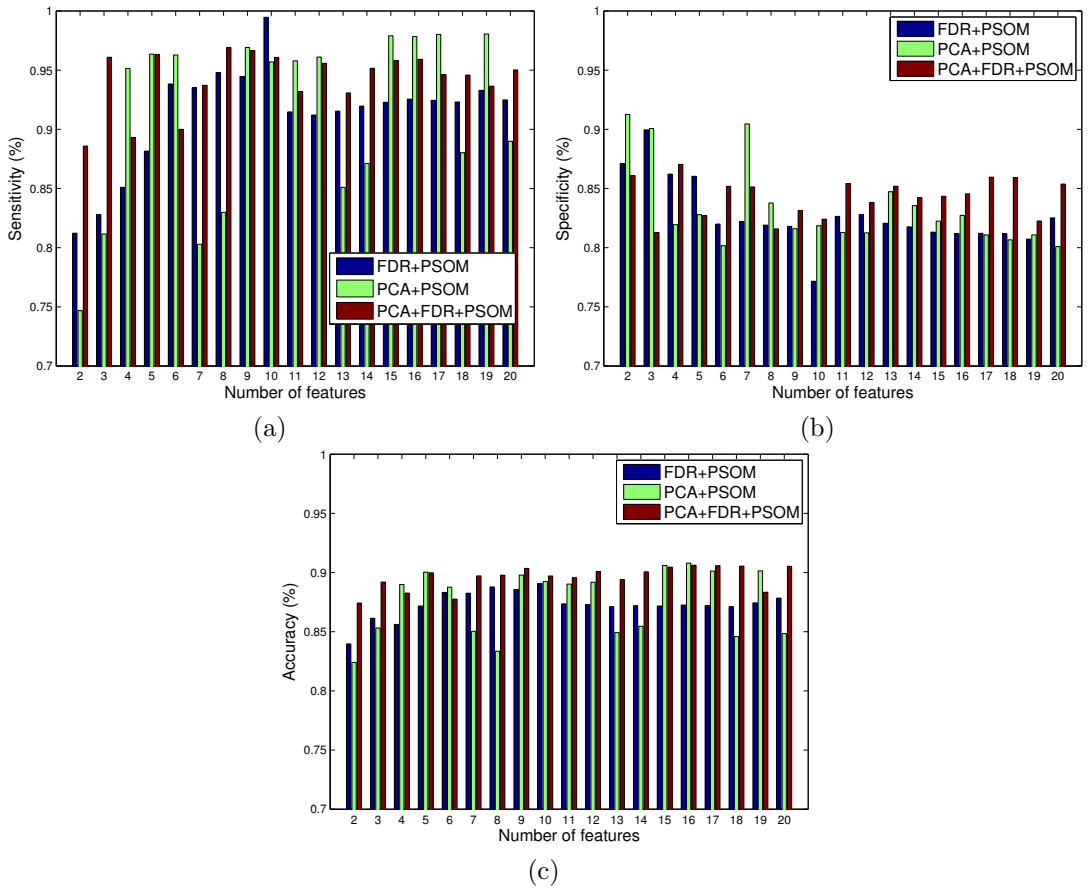

Figure 3: Classification results as a function of the number of selected features using. Sensitivity, specificity and accuracy statistics are shown in (a), (b) and (c), respectively

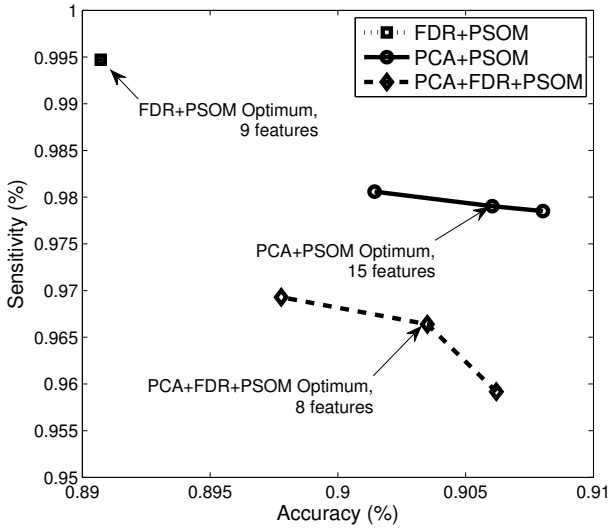

Figure 4: Non-dominated sets for different feature extraction methods. Optimum points are indicated in each plot

On the other hand, results obtained using other statistical methods such as the Relief method [24] or information theory-based algorithms, such as Conditional Mutual Information (CMI) [13], have been included in Table 1 for comparison.

Table 1: Best results obtained for different methods. Standard deviation for the NSL-KDD partition is computed along 50 executions. Standard deviation for cross-validation using merged train/test dataset is computed along 10 folds and 50 executions per fold.

Method	Number of features	Accuracy	Sensitivity	Specificity
NSL-KDD train/test partition				
PSOM+Relief	15	0.87 ± 0.03	0.85 ± 0.05	0.91 ± 0.04
PSOM+CMI	15	0.87 ± 0.02	0.85 ± 0.40	0.93 ± 0.05
PSOM+FDR	9	0.89 ± 0.05	0.98 ± 0.06	0.77 ± 0.06
PSOM+PCA	15	0.90 ± 0.05	0.97 ± 0.05	0.80 ± 0.08
PSOM+PCA+FDR	**8**	**0.90±0.05**	**0.97±0.05**	**0.93±0.06**
10-fold cross-validation (merged dataset)				
PSOM+Relief	15	0.91 ± 0.03	0.90 ± 0.03	0.94 ± 0.02
PSOM+CMI	15	0.90 ± 0.02	0.90 ± 0.04	0.93 ± 0.03
PSOM+FDR	9	0.88 ± 0.01	0.91 ± 0.01	0.90 ± 0.01
PSOM+PCA	15	0.92 ± 0.01	0.88 ± 0.03	0.94 ± 0.02
PSOM+PCA+FDR	**8**	**0.93±0.01**	**0.89±0.02**	**0.96±0.05**

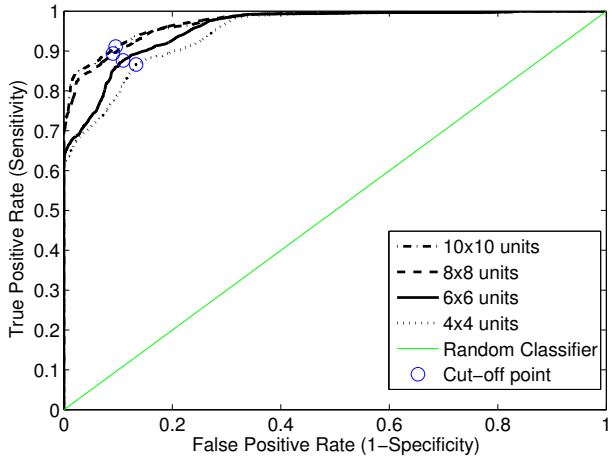

Figure 5: ROC curves for different map sizes for PSOM+PCA+FDR

It is worth mentioning that the number of features indicated in Figure 3 and Table 1 correspond to specific features selected depending on the method. Consequently, the nine features selected by the PSOM+FDR method refer to the ones providing the higher FDR value (that is, FDR has been used as selection criterion in this method). In the PSOM+PCA method, the features are generated by using the eigenvectors in increasing order of variance explained and the PSOM+PCA+FDR method selects the eigenvectors whose projection provides a higher FDR value (that is, the most discriminative eigenvectors according to the FDR criterion).

The second experiment aims to find out the number of SOM units for maximum performance. The number of SOM units plays an important role on the

quantization error and determines the quality of the auto-organization process. Furthermore, the distance between units belonging to different classes also depends on the map size. Thus, the *Receiver Operating Curves* (ROC) [32] have been computed measuring the difference between the mean probability activation for units labelled as *normal* and *anomaly* while varying the map size. The ROC curves are shown in Figure 5, where the maximum performance occurs for map size of 8x8 units (that is, maximum area under ROC curve). In this case, the cut-off point provides sensitivity values of 0.92 and specificity values of 0.95. Table 2 shows the area under ROC curve (AUC) for different map sizes.

Table 2: Area under ROC curve for different map sizes for PSOM+PCA+FDR

Map Size	AUC
4x4	0.94
6x6	0.96
8x8	**0.97**
10x10	0.95

Additionally, we show the activation of the map for both normal and anomaly samples. Thus, Figure 6 shows the log posterior activation probabilities of the SOM units for normal and anomaly samples in the left and right column, respectively. As shown in these figures, activation patterns for normal and anomaly connections can be figured out according to similar map unit probabilities. Moreover, normal samples activate most units, while anomaly units remain dormant.

Classification results with the SOM trained using the above-indicated parameters are provided in Table 1 for comparison with the best results obtained using other feature selection techniques. Additionally, Table 3 shows the results obtained using all the features for comparison with previous approaches. In this table, results for PSOM by using PCA+FDR and PCA as feature selection methods are the same since all the features are used (that is, eigenvectors are not filtered).

Table 3: Classification results for different classification methods. All features are used in all cases for comparison

Method	Number of features	Accuracy	Sensitivity	Specificity
Naïve Bayes [40]	41	$0.76 \pm *$	*	*
Random Forest [40]	41	$0.80 \pm *$	*	*
Decision Trees [40]	41	$0.81 \pm *$	*	*
PSOM+FDR	41	0.60 ± 0.01	0.45 ± 0.30	0.9 ± 0.3
PSOM+PCA	41	0.88 ± 0.01	0.92 ± 0.02	0.84 ± 0.01
PSOM+PCA+FDR	**41**	**0.88±0.01**	**0.92±0.02**	**0.84±0.01**

* Data not provided by the source.

4. Related Work

Works in anomaly detection based on classifying pattern deviations from normal network activity are usually focused on two issues. The first is related to select features to allow obtaining high classification rates, by means of *wrapper*

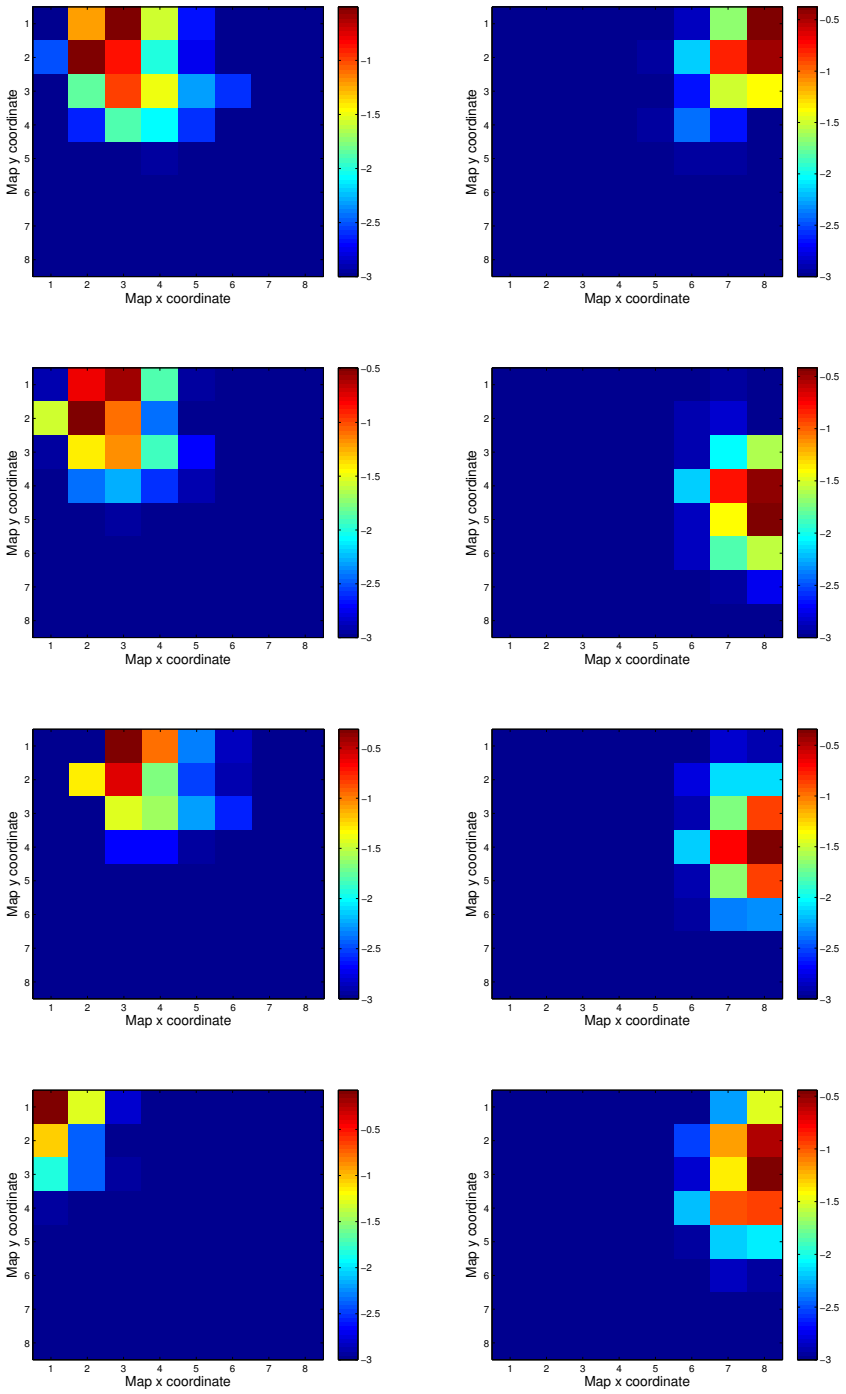

Figure 6: Activation patterns for normal (left column) and anomaly (right column) using random samples from the test data manifold. Figures shown 8x8 SOM trained with eight features as indicated in the text. Activation levels indicate logarithmic posterior probabilities as different colours according to the colorbar

or *filter* techniques. Unlike *filters*, *wrappers* use an objective function that returns a goodness measure of the selected feature. This feedback is obtained from the classifier performance (that is, classification accuracy or error) executed on the training set. However, *wrappers* are classifier-dependent, and require executing the training process in each iteration, being computationally expensive. On the other hand, *filters* do not involve the use of a classifier and rank the features according to their importance for separating classes using either statistical methods or information theory-based methods. Statistical methods include parametric or non-parametric hypothesis-testing, such as the Student's *t-test* or *Mann-Whitney U-test* [36, 51], respectively. Additionally, *relief* ranks the features using an instance-based learning method to assign a relevance weight to each feature for two-class classification problems. Other statistical methods, such as the Fisher Discriminant Ratio, quantify the discriminative power of individual features [51].

Information theory-based methods can use different metrics, such as entropy, Kullback-Leibler divergence [51] or the information gain measure [43] to rank the features. Hence, [13, 54] used the Conditional Mutual Information (CMI) to select feature subsets. Other *filter* algorithms use a correlation-based metric to evaluate feature usefulness. Specifically, the Correlation-Based Feature Selection (CFS) algorithm [19, 58] takes into account the worth of individual features based on the hypothesis that *good feature subsets are highly correlated with the class, yet uncorrelated with each other* [19]. *Filters* have been used for feature selection in intrusion detection applications in many works. This way, [6] used information theory *filters* and [6, 3, 60] used statistical techniques to de-correlate the feature set or for noise removal [60, 34]. Other works, such as [5] used PCA, as in the *eigenfaces* approach, and computed a reduced set of representative vectors called *eigenconnections* to model all types of connections present in the dataset. Other works make use of non-linear projection techniques to obtain a discriminative set of features in a lower dimension space [9]. However, not all of these works use the same dataset or follow the same evaluation protocol, making it difficult to compare the results [49]. In fact, [3] provides extensive information regarding the most recent works, techniques and datasets used in intrusion detection research. In the present work, we used the DARPA KDD99-based dataset to compare the proposed approach as it is one of the most-used datasets according to recent works [3, 40, 39, 28, 55]. Moreover, we propose a new feature selection method based on PCA, but in a different way than in [60, 6, 5, 34]. We select the principal components by the discriminative power of the projections and not by variance. This is addressed combining PCA with statistical techniques (FDR) to sort components by their maximum class separation capability.

Other works focus on the classification stage [3]. Thus, pure parametrical and non-parametrical statistical methods, clustering and outlier-based methods, soft computing methods (including neural networks, and optimization techniques), knowledge-based methods and hybrid approaches are used in different works [3]. Specifically, SOM is used in [41] to compute clusters associated with normal or anomaly connections. In [44], connections are classified by means of the distance to normal and anomaly clusters. In [21], a hierarchical structure of SOM maps is proposed to figure out clusters associated with different attacks. In our approach, a unique SOM map is used that implies a reduction in the computational burden. Moreover, we use the SOM to model the normal and anomaly connections by

measuring the activation probability of each map unit. This way, not only the BMU is used, but also the activation level of the SOM units. This is addressed by modelling the trained SOM map using a GMM in which each unit acts as a cluster centre. This way, posterior activation probabilities can be computed for new data instances, and these activation levels are used for classification.

Gaussian Mixture Models (GMM) have been applied for pattern recognition and clustering problems [57], and several methods have been proposed to determine their components [57, 47]. Among them, the EM (Expectation-Maximization) algorithm based on the maximum likelihood principle [11] has been frequently applied although it presents some drawbacks related to its convergence speed and initialization sensitivity [57]. Different probabilistic reformulations of Self-Organizing Maps have been also considered for learning the Gaussian mixtures. For example, in [47], the GMM was obtained by minimizing the Kullback-Leibler information metric; in [4], the generative topographic mapping (GTM) based on the use of radial basis functions and the EM algorithm was proposed. In S-Map [25], the GTM and the SOM learning algorithm were combined. Recently, the Naive Bayes SOM (NBSOM) [47] has been proposed. The current approach citeRiveiro2008 also uses the prior probabilities in the output map, although it sets similar values (one divided by the number of prototypes) for all of them. Moreover, the log-likelihood is used instead of likelihood [46], and the EM algorithm is applied to maximize the log-likelihood.
The main advantage of the proposed approach is related to the possibility of modifying the classification performance by adjusting the prior activation probabilities of each SOM unit, which eventually modifies the classification performance without re-training. In other words, the system can be trained once and can be further adjusted by means of the prior activation probabilities, avoiding the execution of the entire training process for new samples. Thus our main contributions in this paper can be summarized as follows:

- The use of a PCA/FDR approach for selecting features according to their discriminative power.
- The use of a hybrid SOM-GMM approach to model normal patterns and anomalies, providing a fuzzy response of the map units that allows measuring the posterior activation probabilities for new data instances according to the Bayes Theorem.
- The improvement of the classification performance by adjusting the prior activation probabilities, avoiding re-training when new instances diminish the classification accuracy or sensitivity

In addition, we used standard and public KDD99-based datasets and our results are compared with other works using the same dataset. Furthermore, provided datasets are used for training and testing tasks respectively, ensuring a correct validation of the results [49].

5. Conclusions

In this paper, we present a method for network intrusion detection based on SOM and PCA. Moreover, noise in the dataset and low variance features are

filtered by means of PCA and FDR. This procedure uses the most discriminative projections not only based on the variance explained by the eigenvectors, but also in their discriminative power. Subsequently, prototypes generated by the self-organizing process are modelled by d Gaussians where d is the number of SOM units. This allows the proposed system to be trained only once, as classification is performed by means of the mixture models, using the posterior activation level of each unit. On the other hand, experiments to optimize parameters of the classifier such as the map size have been performed by computing the ROC curves. Using the activation probabilities computed during the training stage, we obtained sensitivity, specificity and accuracy values up to 97 percent, 93 percent and 90 percent, respectively. In the proposed method, classification capabilities can be modified by varying the prior activation probabilities of the SOM units, avoiding training the SOM for new data. This way, the accuracy may be improved by tuning the detection threshold. Although these prior probabilities can be modified by the network administrator, it is also possible to adjust them automatically. Thus, as future work, we plan to improve the computation of the prior activation probabilities by means of multi-objective optimization. On the other hand, several SOMs could be combined in an SOM ensemble to build a hierarchical model in order to classify not only normal and abnormal connections, but also the four types of attacks described in the dataset.

Acknowledgements

This work has funded by the Ministerio de Ciencia e Innovación of the Spanish Government and FEDER funds under Project No. TIN2012-32039. The authors would like to thank the reviewers for their useful comments and suggestions.

References

[1] E. Alhoniemi, J. Himberg, and J Vesanto. Probabilistic measures for responses of self-organizing map units. In *Proc. of the International ICSC Congress on Computational Intelligence Methods and Applications. (CIMA)*, volume 1, pages 286–290, 1999.

[2] I. Alvarez, J.M. Gorriz, J. Ramirez, D. Salas-Gonzalez, M.M. Lopez, F. Segovia, R. Chaves, M. Gomez-Rio, and C. Garcia-Puntonet. 18f-fdg pet imaging analysis for computer aided Alzheimer's diagnosis. *Information Sciences*, 184(4):903–196, 2011.

[3] M. Bhuyan, D. Bhattacharyya, and J. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, PP(99), 2013.

[4] M.; Williams C.K.I. Bishop, G.M.; Svensn. Gtm: The generative topographic mapping. *Neural Computation*, 10:215–234, 1998.

[5] Y. Bouzida and S. Gombault. Eigenconnections to intrusion detection. In *In 19 th IFIP International Information Security Conference (SEC2004)*, pages 241–258. IEEE, 2004.

[6] Y. Chen, Y. Li, X. Chend, and L. Guo. Survey and taxonomy of feature selection algorithms in intrusion detection system. In *Lecture Notes in Computer Science*, volume 4318, pages 153–167, 2006.

[7] S. Choi, S. Cha, and C.C. Tappert. A survey of binary similarity and distance measures. *Cybernetics and Informatics*, 8(1):43–48, 2010.

[8] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.

[9] E. De la Hoz, A. Ortiz, J. Ortega, and E. De la Hoz. Network anomaly classification by support vector classifiers ensemble and non-linear projection techniques. In *International Work-Conference on Artificial Neural Networks (IWANN). Lecture Notes in Computes Science*, volume 8073, pages 103–111. Springer, 2013.

[10] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[11] A. Dempster, N. Laird, and D. Rubing. Maximum likelihood from incomplete data via em algorithm. *Journal of the royal society*, 39:1–38, 1977.

[12] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer, 2002.

[13] François Fleuret. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.*, 5:1531–1555, December 2004.

[14] S. Foithong, O. Pinngern, and B Attachoo. Feature subset selection wrapper based on mutual information and rough sets. *Expert Systems with Applications*, 39:674–584, 2012.

[15] A.A. Ghorbani, W. Lu, and M. Tavallae. *Network intrusion detection and prevention: concepts and techniques*. Springer-verlag, 2009.

[16] J. Ghosh, J. Wanken, and F. Charron. Detecting anomalous and unknown intrusions against programs. In *Proc. of the Annual Computer Security Applications Conference*, volume 1, pages 259–267, 1998.

[17] F. Gong. Deciphering detection techniques: Part ii anomaly-based intrusion detection. *McAfee Network Security Technologies Group. White paper*, 1:1–10, 2003.

[18] H. Gunes Kayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood. A hierarchical som-based intrusion detection system. *Eng. Appl. Artif. Intell.*, 20(4):439–451, June 2007.

[19] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[20] S Haykin. *Neural Networks*. Prentice-Hall, 2nd edition, 1999.

[21] D. Ippoliti and X. Zhou. A-ghsom: An adaptive growing hierarchical self organizing map for network anomaly detection. *Journal of Parallel and Distributed Computing*, 72(12), 2012.

[22] S. Joseph. Feature Reduction using Principal Component Analysis for Effective Anomaly Based Intrusion Detection on NSL-KDD. *International Journal of Engineering Science*, 2(6):1790–1799, 2010.

[23] H.G. Kayacik, A.N. Zincir-Heywood, and M.I. Heywood. Selecting features for intrusion detection: a feature relevance analysis on kdd99 intrusion detection datasets. In *Proceedings of the 3rd Conference on Privacy, Security and Trust*, 2005.

[24] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *In Proceedings of the Second Workshop on Hot Topics in Networks (HotNets-II)*, page 129134. AAAI Press, 1992.

[25] K. Kiviluoto and E. Oja. S-map: A network with a simple self-organization algorithm for generative topographic mappings. *Advances in Neural Processing Systems*, 10:549–555, 1995.

[26] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[27] C. Kreibich and J. Crowcroft. Honeycomb - creating intrusion detection signatures using honeypots. In *In Proceedings of the Second Workshop on Hot Topics in Networks (HotNets-II)*, 2003.

[28] S. Lakhina, S. Joseph, and B. Verma. Feature reduction using principal component analysis for effective AnomalyBased intrusion detection on NSL-KDD. *International Journal of Engineering Science and Technology*, 2(6):1790–1799, June 2010.

[29] I. Levin. KDD-99 Classifier Learning Contest, LLSoft's Results Overview. *SIGKDD explorations. ACM SIGKDD*, 1(2):65–66, 2000.

[30] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendball, D. McClung, D. Weber, S.E. Webster, D. Wyschgrod, R.K. Cuningham, and M.A. Zissman. Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. *Descex*, 2:1012–1027, 2000.

[31] M. López, J. Ramírez, J. Górriz, I. Álvarez, D. Salas-González, F. Segovia, R. Chaves, P. Padilla, and M. Gómez-Río. Principal component analysis-based techniques and supervised classification schemes for the early detection of Alzheimer's disease. *Neurocomputing*, 74(8):1260–1271, 2011.

[32] R.A. Maxion and R.R. Roberts. Proper use of roc curves in intrusion/anomaly detection. Technical Report CS-TR-871, School of Computing Science, University of Newcastle upon Tyne, 2004.

[33] J. McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa instrusion detection systems evaluation as performed by lyncoln laboratory. *ACMTransactions on Information and Systems Security*, 3(4):262–294, 2000.

[34] L. Mechtri, F. Djemili Tolba, and N. Ghoualmi. Intrusion detection using principal component analysis. In *Proc. of the Second International Conference on Engineering Systems Management and Its Applications (ICESMA). Sharjah*, pages 1–6. IEEE, 2010.

[35] MIT Lincoln Labs. Darpa intrusion detection evaluation. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html, 2008.

[36] L. Navidi. *Statistics for Engineers and Scientists*. Wiley, 2010.

[37] Network Security Lab - Knowledge Discovery and Data Mining (KDD). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2007.

[38] NSL-KDD dataset. http://nsl.cs.unb.ca/NSL-KDD/, 2007.

[39] J.P. Nziga. Minimal dataset for network intrusion detection systems via dimensionality reduction. In *6th International Conference on Digital Information Management (ICDIM)*, 2011.

[40] M. Panda, A. Abraham, and Patra M.R. Discriminative multinomial naïve bayes for network intrusion detection. In *6th Conference on Information Assurance and Security (IAS)*, 2010.

[41] S.K. Panigrahy, J.R. Mahapatra, J. Mohanty, and S.K. Jena. Anomaly detection in ethernet networks using self organizing maps. In *Lecture Notes in Computer Science*, volume 125, pages 300–305, 2011.

[42] B. Pfahringer. Winning the FDD99 clasification Cup: Bagged-Boosting. *SIGKDD explorations. ACM SIGKDD*, 1(2):67–75, 2000.

[43] J.R. Quinlan. Induction of decision trees. *Machine learning*, pages 81–106, 1986.

[44] M. Ramadas, S. Ostermann, and B. Tjaden. Detecting anomalous network traffic with self-organizing maps. In *Lecture Notes in Computer Science*, volume 2820, pages 36–54, 2003.

[45] M. Riveiro, F. Johansson, G. Falkman, and T. Ziemke. Supporting maritime situation awareness using self organizing maps and gaussian mixture models. In *Proceedings of the 2008 conference on Tenth Scandinavian Conference on Artificial Intelligence (SCAI)*, volume 1, pages 84–91, 2008.

[46] M. Riveiro, F. Johansson, G. Falkman, and T. Ziemke. Supporting maritime situation awareness using self organizing maps and gaussian mixture models. In *Proceedings of the 2008 conference on Tenth Scandinavian Conference on Artificial Intelligence: SCAI 2008*, 2008.

[47] G. Ruz and D.T. Pham. Nbsom: The naïve bayes self-organizing map. *Neural Comp. & Applic.*, 21:1319–1330, 2012.

[48] K. Tasdemir, P. Milenov, and B. Tapsall. Topology-based hierarchical clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, 22(3):474–485, 2011.

[49] M. Tavallaee, N. Stakhanova, and A.A. Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *Trans. Sys. Man Cyber Part C*, 40(5):516–524, September 2010.

[50] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

[51] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

[52] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1992.

[53] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. Som toolbox. Helsinki University of Technology, 2000.

[54] Gang Wang and Frederick H. Lochovsky. Feature selection with conditional mutual information maximin in text categorization. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 342–349, New York, NY, USA, 2004. ACM.

[55] Tuo Wang, S. Mabu, Nannan Lu, and K. Hirasawa. A novel intrusion detection system based on the 2-dimensional space distribution of average matching degree. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 2829–2834, Sept.

[56] S. Xiaonan and W. Banhzaf. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, 10(1):1–35, 2010.

[57] H. Yin and N.M. Allinson. Bayesian self-organizing map for gaussian mixtures. *IEEE Proc. Vis. Image Signal Processing*, 148(4):234–240, 2012.

[58] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, December 2004.

[59] S. Zargari and D. Voorhis. Feature selection in the corrected kdd-dataset. In *Proc. 3rd International Conference on Emerging Intelligent Data and Web Technologies*, pages 174–180, 2012.

[60] L. Zhao, H. Kang, and S. Kim. Improved clustering for intrusion detection by principal component analysis with effective noise reduction. In *Lecture Notes in Computer Science*, volume 7804, pages 490–495, 2013.