Fast Predictor-Corrector Intrusion Detection System Based on Clustering

Slobodan Petrović, Gonzalo Álvarez,
Agustín Orfila and Javier Carbó

Instituto de Física Aplicada (C.S.I.C.)

Serrano 144, 28006 Madrid, España

E-mail: {slobodan, gonzalo}@iec.csic.es
{adiaz, jcarbo}@inf.uc3m.es

Fast Predictor-Corrector Intrusion Detection System Based on Clustering

Abstract. A predictor-corrector intrusion detection system is proposed, whose predictors are various clustering algorithms with different initial parameters that operate in parallel on the current data set. The decisions whether abnormal behaviour is detected in the current data set are made by a number of assessors that implement various clustering quality evaluation techniques. The manager of the system estimates the quality of decision making from the pieces of information obtained a posteriori and then varies the parameters of the predictors and/or the assessors in order to achieve better overall performance of the system. In such a way, the intelligence of the system is delegated to higher decision making levels, which improves the effectiveness. Experimental results regarding the effectiveness of the system are given with the KDD CUP 1999 test data as the reference data set. These results show that very good overall performance can be achieved by selecting properly various system parameters.

Key words: Intrusion detection system, Anomaly detection, Clustering, Decision making.

1 Introduction

Detection of intrusion has become one of the most important issues in contemporary computer networks. The general solution of this problem is difficult to find, since the normal and abnormal behaviour in these networks are hard to predict as the boundaries cannot be defined precisely. To address this, many general artificial intelligence and pattern recognition techniques have been used. Various artificial intelligence techniques are notorious for their inefficiency, which sharply limits the applications of such methods. To approach the on-line intrusion detection, a much more efficient system is needed. Thus, various statistical pattern recognition techniques seem to be more adequate for this purpose.

The idea of using various clustering methods in intrusion detection has always been very popular among researchers in this field, especially because of the efficiency of clustering algorithms compared with other techniques. The clustering methods have been used either directly on incoming data (see for example [7]) or as a supporting technique in a stage posterior to data classification [5]. The main problem of the use of clustering in intrusion detection is the interpretation of clustering results, so called "labeling" of clusters. Namely, without additional information (which is, by contrast, always present in the learning systems) it is difficult to decide whether the data classified in one cluster corresponds to "normal" behaviour in the monitored network or to "abnormal" one.

Cardinality of clusters is often used as the decision parameter for this purpose [7] because the mathematical expectation of "normal" behaviour is considered greater than that of "abnormal" behaviour. However, this approach has some serious drawbacks, which will be discussed later in this paper. Solving this problem and other problems regarding correctness of clustering results require a more complex system. Various multiple classifier systems have been proposed for this purpose, in which several classification algorithms (supervised or unsupervised) operate on the same data and later the final decision about whether an attack occured or not is reached by means of a decision fusion function (see for example [3]). Multiple classifying systems can also be implemented by means of adaptive agents [1] in which a correction of classification parameters is present based on information obtained a posteriori.

In this paper, we propose a multiple predictor-corrector intrusion detection scheme based on clustering of the incoming resource access requests. The system contains a number of predictors, which perform clustering of the incoming requests into two clusters that represent "abnormal" and "normal" behaviour. These requests are submitted to the system in the form of data sets created at predefined time intervals or alternatively, upon a predefined number of incoming requests. The predictors differ in the initial parameters of the clustering process and/or in the clustering methods implemented. There are also several assessors in the system that make the decisions whether an abnormal behaviour occured or not based on the clustering results of the predictors. The manager of the system monitors the decision making process and varies the decision making parameters of the assessors and the initial parameters of the clustering process performed in the predictors if necessary. In such a way, the intelligence of the system is delegated to the higher decision making levels, which improves the overall effectiveness. The functioning of the manager may include human interaction as well.

The system is very flexible because it permits varying of many parameters, such as the number of predictors, the clustering methods applied in the predictors, the decision making parameters of the assessors (evaluation criteria of the clusters' quality, various thresholds, etc.), the monitoring parameters of the manager and the criteria for varying of the clustering parameters, etc. Experimental results are given, in which the effectiveness of the system is expressed through Receiver Operating Characteristics (ROC) curves for various system parameters. The well known KDD CUP 1999 network resource requests data base [6] is used as the reference input data set.

The structure of the paper is the following: In Section 2, the general description of the predictor-corrector system is given. In Section 3, the details of the clustering methods for predictors and their parameters, the clustering evaluation criteria implemented in the assessors and the managing algorithm, are presented. In Section 4, the experimental work is described and the results of the experiments are given. Finally, Section 5 concludes the paper.

2 General description of the system

The global scheme of the predictor-corrector system is given in the Fig. 1. The system contains K predictors $\mathcal{P}_1, \ldots \mathcal{P}_K$ that operate in parallel on the same data set $\mathbf{X}_\tau$, $\tau = 0, 1, 2, \ldots$. Every predictor is merely a clustering algorithm that classifies the input data set into 2 clusters, without interpretation of the clustering results. Since any clustering algorithm can be considered a suboptimal algorithm for finding the extremal value (local maximum or minimum) of a clustering criterion function, the solution in general depends on some initial parameters such as the initial permutation of the input data set vectors, etc. By setting these parameters different for each predictor, the versatility of clustering results is achieved, which improves the chances to find the partition of the input data set closest to the optimal one.

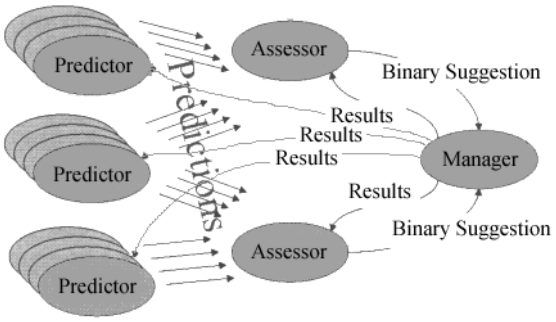

Fig. 1 - The global scheme of the predictor-corrector IDS

There are also $\mathcal{L}$ assessors in the system, $\mathcal{A}_1, \ldots, \mathcal{A}_\mathcal{L}$, whose task is to make decisions about whether an abnormal behaviour in the monitored network occured while processing the current data set $\mathbf{X}_\tau$. For this to be carried out, every assessor calculates the value of its own criterion function for every predictor over the data set $\mathbf{X}_\tau$. The local extremal value (maximum or minimum) of this function determines the binary decision of the assessor - the abnormal behaviour was detected (was not detected) for some element of the data set $\mathbf{X}_\tau$. More precisely, let us define possible outcomes of the decision making system implemented in the assessor $\mathcal{A}_l$:

H_0 - the abnormal behaviour was not detected for an element of $\mathbf{X}_\tau$;
H_1 - the abnormal behaviour was detected for an element of $\mathbf{X}_\tau$.

Let $\mathbf{X}_\tau = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_\tau}\}$ be the current input data set, $\tau = 0, 1, 2, \ldots$, where $\mathbf{X}_1, \ldots, \mathbf{X}_{N_\tau}$ are the vectors that encode the incoming resource access requests, $\mathbf{X}_i = (x_1, \ldots, x_{M_i})$, $i = 1, 2, \ldots, N_\tau$, $x_j \in \mathcal{R}$, $j = 1, 2, \ldots, M_i$. Let $\mathcal{C}_{k,\tau}$, $k = 1, \ldots, K$ be the partitions of $\mathbf{X}_\tau$ obtained by the predictors $\mathcal{P}_1, \ldots \mathcal{P}_K$, respectively:

$$\mathcal{C}_{k,\tau} = \{\mathbf{Y}_{k,\tau}, \mathbf{Z}_{k,\tau}\}, \qquad \mathbf{Y}_{k,\tau} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_k}\}, \qquad \mathbf{Z}_{k,\tau} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_{m_k}\},$$

$$\mathbf{Y}_{k,\tau} \cup \mathbf{Z}_{k,\tau} = \mathbf{X}_\tau, \qquad \mathbf{Y}_{k,\tau} \cap \mathbf{Z}_{k,\tau} = \varnothing, \qquad k = 1, \ldots, K.$$

Let $\mathcal{F}_{l,\tau} : \mathcal{C}_{k,\tau} \longrightarrow \mathcal{R}$ be the clustering quality criterion function of the assessor $\mathcal{A}_l$ for the data set $\mathbf{X}_\tau$ and let $T_{l,\tau} \in \mathcal{R}$ be a threshold value. Let $D_{l,\tau} : \mathcal{R} \times \mathcal{R} \longrightarrow \{0, 1\}$ be the decision function of the assessor $\mathcal{A}_l$. Then:

$$D_{l,\tau}(\mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}), T_{l,\tau}) = \begin{cases} 0, \mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}) \geq T_{l,\tau} \\ 1, \mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}) < T_{l,\tau} \end{cases} \quad \text{(minimization)} \qquad (1)$$

or, alternatively

$$D_{l,\tau}(\mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}), T_{l,\tau}) = \begin{cases} 0, \mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}) < T_{l,\tau} \\ 1, \mathcal{F}_{l,\tau}(\mathcal{C}_{k,\tau}) \geq T_{l,\tau} \end{cases} \quad \text{(maximization)}, \qquad (2)$$

where H_0 is accepted if $D_{l,\tau} = 0$ and rejected otherwise. The threshold $T_{l,\tau}$ is determined by the manager $\mathcal{M}$ so as to minimize the probability of missing the event, $p(H_0/H_1)$ and false alarm $p(H_1/H_0)$ in the assessor $\mathcal{A}_l$. After determining $p(H_0/H_1)$ and $p(H_1/H_0)$ for every assessor, the manager can change the parameters of the predictors and/or the assessors in order to improve the correctness of the decisions. For this to be carried out, the manager needs some information a posteriori.

3 Appropriate elements for implementation

Let us now concentrate on the selection of appropriate elements for the implementation of system components. We first briefly review some of the most often used clustering algorithms that can be implemented in the predictors as well as some proximity/distance measures. Then we discuss some clustering evaluation techniques that can be implemented in the assessors. Finally, we consider some system management strategies suitable for implementation in our IDS.

3.1 Clustering algorithms for predictors

It is important to select clustering methods for predictors such that both good efficiency and correctness of the clustering process are achieved. There are many clustering algorithms that are widely used in practice. These algorithms belong either to the group of so called aglomerative procedures or to the group of partitioning procedures. An excellent recent review of various clustering algorithms can be found in [4]. Various aglomerative procedures, also known as hierarchical (single linkage, complete linkage Ward's method, centroid method, etc.) are used very often because of the simplicity of their implementation and the fact that they produce a hierarchy on output, i.e. clustering into all the possible numbers of clusters. They have also been used in IDS (see for example [7]). But the results that the majority of them produce are too biased. The partitioning methods are

generally considered more accurate. The most often used method from the class of partitioning methods is the well known K-means algorithm (see for example [4]). We consider this algorithm the best trade-off between correctness and efficiency. The basic K-means algorithm is presented in the Fig. 2. This algorithm can also be modified to process vectors of different lengths.

1. Initialization: Randomly choose *K* instances from the data set and make them initial cluster centers.
2. Assignment: Assign each instance to its closest center.
3. Updating: Replace each center with the mean of its members.
4. Iteration: Repeat steps 2 and 3 until there is no more updating.

Fig. 2 - The K-means algorithm

The appropriate metric used for defining similarity/distance between input data set vectors depends on the character of their coordinates. For equal length vectors, whose coordinates take values from the set $\mathcal{R}$, the Minkowski metric is widely used as a distance measure:

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt[q]{\sum_{i=1}^{n} \mid x_i - y_i \mid^q} \tag{3}$$

where n is the dimension of the vectors $\mathbf{X}$ and $\mathbf{Y}$. Some important special cases of this metric are: Manhattan (or city block) distance, for $q = 1$, Euclidean distance, for $q = 2$, and Chebychev distance, for $q \longrightarrow \infty$. For equal length discrete vectors, the Hamming distance measure is often used. For unequal length discrete vectors, the edit distance, as the minimum number of elementary edit operations (deletions, insertions and substitutions of symbols) needed to transform one vector into another is widely used.

3.2 Cluster validation techniques for assessors

Having obtained clusters from the predictors, the next task of the IDS is to label them, i.e. to determine which cluster corresponds to "normal" behaviour, and which to "abnormal". Since there is no learning on labeled data in the system, the assessors must use additional criteria to decide on this. In [7], the first assumption is that few anomalies are expected in the clustering results, so the significant difference in cardinalities of the clusters naturally labels the cluster with greater cardinality as that corresponding to "normal" behaviour. However, there are two basic issues related with such a strategy: first, normal data transmitted by means of a less frequently used protocol (such as ftp or telnet) might produce clusters of very different cardinalities, which could mislead such an assessor. Second, there are some Denial-of-Service attacks, such as syn-Flood etc. that can mislead this labeling strategy by making the mathematical expectation of the attack much greater than that of the "normal" behaviour. To overcome the problems associated with the strategy described above, we propose the implementation of various cluster validation techniques, each of them in its own assessor. Keeping

in mind the initial assumption that the "abnormal" behaviour in the network should have "quite different" properties from the "normal" one in some labeling of clusters, by evaluating such labelings we hope to detect this difference.

The general clustering validation measures met in the literature, such as Silhouette index, Dunn's index, and Davies-Bouldin index (see for example [2]) can be used here because of ease of their implementation and reliability. All these measures opt for detecting well separated and compact clusters. However, in this paper we focus on two measures: the cluster cardinality measure, because it has already been used in IDS, and a pure graph theoretic measure, because we believe that the use of this measure can resolve the problem of massive attacks detection.

In the sequel we give formal definitions of the cluster cardinality criterion and the graph theoretic criterion that we propose for this purpose.

Cluster cardinality criterion Let $\mathbf{X}_\tau = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_\tau}\}$ be the current data set and let $\mathcal{C}_{k,\tau} = \{\mathbf{Y}_{k,\tau}, \mathbf{Z}_{k,\tau}\}$ be the partition of $\mathbf{X}_\tau$ obtained in the predictor $\mathcal{P}_k$. Let $\lambda_j \in \{1,2\}$ be the label of the vector $\mathbf{X}_j$ in the data set $\mathbf{X}_\tau$, where $\lambda_j = 1$ is interpreted as "normal" behaviour. If $\mid \mathbf{Y}_{k,\tau} \mid \geq \mid \mathbf{Z}_{k,\tau} \mid + \mathcal{D}_C$, where $\mathcal{D}_C$ is given in advance then $\lambda_j = 1$ for $\mathbf{X}_j \in \mathbf{Y}_{k,\tau}$ and $\lambda_j = 2$ otherwise. If $\mid \mathbf{Z}_{k,\tau} \mid \geq \mid \mathbf{Y}_{k,\tau} \mid + \mathcal{D}_C$, then $\lambda_j = 1$ for $\mathbf{X}_j \in \mathbf{Z}_{k,\tau}$ and $\lambda_j = 2$ otherwise.

Graph theoretic criterion To use this graph-theoretic criterion, a graph representation of clustering results is needed, for which an adequate definition of adjacency relation is to be defined.

Let $\mathbf{X}_\tau = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_\tau}\}$ be the current data set. Let $d(\mathbf{X}_k, \mathbf{X}_l)$ be the distance between $\mathbf{X}_k$ and $\mathbf{X}_l$, $\mathbf{X}_k, \mathbf{X}_l \in \mathbf{X}_\tau$, and let $\mathcal{D}$ be a threshold given in advance. Then $\mathbf{X}_k$ and $\mathbf{X}_l$ are in relation if and only if $d(\mathbf{X}_k, \mathbf{X}_l) \leq \mathcal{D}$. In such a way, we obtain the graph $\mathcal{G} = (V, E)$ whose set of vertices is V and the set of edges is E from the given set of input data vectors $\mathbf{X}$: the vectors $\mathbf{X}_i$, $i = 1, \ldots, N$, correspond to its vertices, and its edges are defined by the relation defined above. In general, the number of edges increases as $\mathcal{D}$ increases, so it is possible to increase $\mathcal{D}$ until the graph $\mathcal{G}$ is connected. Now let $\mathcal{C}_{k,\tau} = \{\mathbf{Y}_{k,\tau}, \mathbf{Z}_{k,\tau}\}$ be the partition of $\mathbf{X}_\tau$ obtained in the predictor $\mathcal{P}_k$. Consider the subgraphs $\mathcal{G}_Y$ and $\mathcal{G}_Z$ of $\mathcal{G}$ induced by $\mathbf{Y}_{k,\tau}$ and $\mathbf{Z}_{k,\tau}$, respectively. We define the compactness of the cluster $\mathcal{C}_i$, $i = 1,2$ in the following way:

$$Q(\mathcal{C}_i) = \frac{2\mathcal{N}_{\mathcal{C}_i}}{N_{\mathcal{C}_i}(N_{\mathcal{C}_i} - 1)} \tag{4}$$

where $\mathcal{N}_{\mathcal{C}_i} = \mid E_{\mathcal{C}_i} \mid$, i.e. the number of edges in the subgraph $\mathcal{G}_{\mathcal{C}_i}$ of $\mathcal{G}$ induced by the cluster $\mathcal{C}_i$ and $N_{\mathcal{C}_i} = \mid \mathcal{C}_i \mid$.

Then the graph theoretic clusters' quality criterion can be defined in the following way:

Let $Q = \max\{Q(\mathcal{C}_i)\}$, $i = 1,2$ and let i_Q be the index i for which the maximum is achieved. Let $\mathcal{D}_Q$ be a predefined threshold. Let $\lambda_j \in \{1,2\}$ be

the label of the vector $\mathbf{X}_j$ in the data set $\mathbf{X}_\tau$, where $\lambda_j = 1$ is interpreted as "normal" behaviour. If $Q \geq \mathcal{D}_Q$, then $\lambda_j = 2$ for $\mathbf{X}_j \in \mathcal{C}_{i_Q}$ and $\lambda_j = 1$ otherwise.

3.3 Managing the system

The first assumption that we make about the manager is that it is given some information a posteriori about the real situation. These pieces of information are not supplied for all τ, because it would be too expensive and in that case the use of such an IDS would not make sense. The manager keeps track of such an information in the memory called *attack history*. The attack history $\mathbf{AH} = (ah_\tau, ah_{\tau-1}, \ldots, ah_{\tau-w})$, $ah_i \in \{0, 1, *\}$ is the vector in which the real attack situation is recorded. For a data set i for which the information a posteriori has been obtained the value of the corresponding coordinate of the vector $\mathbf{AH}$ is 0 if no attack really occured in that data set and 1 otherwise. For all the data sets for which the information a posteriori is not present, the value of the corresponding coordinate of the vector $\mathbf{AH}$ is "*". The manager also keeps track of the last w decisions of all the assessors in the decision history matrix $\mathbf{DH}$, with $w+1$ rows and $\mathcal{L}$ columns. In the matrix $\mathbf{DH}$, the element $\mathbf{DH}(i, j) \in \{0, 1\}$ represents the decision made by the assessor $\mathcal{A}_j$ for the data set $\tau - i$, $i = 0, \ldots, w$, $j = 1, \ldots, \mathcal{L}$. The information stored in $\mathbf{AH}$ and $\mathbf{DH}$ is used for changing the manager's decision parameters.

Every assessor has a reliability coefficient assigned, which is updated upon the receipt of the information a posteriori about the quality of previous decisions. The coefficients are increased for the assessors, whose decisions coincided with the information obtained a posteriori, and decreased otherwise.

Let $\mathcal{A}_1, \ldots \mathcal{A}_\mathcal{L}$ be the assessors and let $p_1, \ldots p_\mathcal{L}$ be their reliability coefficients, respectively, $0 \leq p_i \leq 1$, $i = 1, \ldots \mathcal{L}$. Let $D_{l,\tau}$ be the decision function of the assessor $\mathcal{A}_l$. Then the basic decision making steps of the manager are the following:

1. If no external information is present at the moment of decision, go to step 2. Otherwise, compare the received external information with the corresponding element of history and update the reliability coefficients of the assessors in the following way:
   1.1 Let $\tau - k$ be the coordinate of the vector $\mathbf{AH}$ for which the information a posteriori has been obtained and let $\mathbf{AH}(\tau - k)$ be the new value at the coordinate $\tau - k$ of this vector. Let $p'_j = p_j$, $j = 1, \ldots, \mathcal{L}$. If $\mathbf{AH}(\tau - k) = \mathbf{DH}(\tau - k, j)$ then p_j is increased by δ_p, otherwise p_j is decreased by δ_p. The value δ_p is fixed and given in advance. If $p_j - p'_j < 0$ then increase the corresponding threshold ($\mathcal{D}_C$ or $\mathcal{D}_Q$) by a predefined value $\delta_\mathcal{D}$. If the maximum threshold value $\mathcal{D}_{\max}$ is reached then set the corresponding threshold to $\mathcal{D}_{\min}$, where $\mathcal{D}_{\min}$ and $\mathcal{D}_{\max}$ are given in advance.
   If the a posteriori information is received for more than one value of k, the process is repeated for every piece of the received information, starting from the greatest k.

1.2 Let $G_p = \sum_{j=1}^{\mathcal{L}} (p_j - p'_j)$. If $G_p < 0$ then change the parameters of the system by performing the following actions:
   a) Change N by δ_N, where δ_N is fixed and given in advance. If the maximum permitted value $N_{\max}$ of N is reached then set $N = N_{\min}$, where $N_{\min}$ is the minimum permitted value of N. $N_{\min}$ and $N_{\max}$ are given in advance.
   b) Change the distance measure used in the predictors. The change consists of picking the next distance measure from the circular list.

2. Calculate the value of the manager's decision function:

$$M_\tau = \sum_{i=1}^{\mathcal{L}} p_i D_{i,\tau} \tag{5}$$

3. Let $\mathcal{T}_\tau$ be the manager's threshold. The hypothesis H_1 is accepted if $M_\tau \geq \mathcal{T}_\tau$ and rejected otherwise.

4 Experimental work

Extensive simulation of the proposed system and its components has been carried out in order to study its effectiveness. To this end, the following instance of the system has been built:

1. The number of predictors $K = 5$, each performing the 2-means clustering algorithm with a different initial partition, and in which the total number of iterations has been limited to 3.
2. The number of assessors $\mathcal{L} = 3$; the first performs the computation of cardinality based decision making; the other two assessors calculate the graph theoretic criterion with 2 different sets of thresholds.
3. The dimension of the attack history vector $w = 10$, and the frequency of obtaining the a posteriori attack information varies between 2 and 5.

The input data set source that we used was the KDD CUP database [6], whose characteristic is that it has an unrealistically high percentage of attacks. Because of that, the attacks from this database were filtered out in the same way as in [7]. The filtering percentage of 0%, 98% and 99% was used over the 490000 resource requests records of the database. The effectiveness of the system was measured by means of the ROC (Receiver Operating Characteristic) curve for the filtered data set mentioned above. The ROC curve depicts the relationship between false positive rate FPR and true positive rate TPR, where:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

In the equations above, FP is the number of false positive outcomes of the intrusion detection on a fixed data set, TP is the number of true positive outcomes, i.e. successful detections, TN is the number of true negative outcomes and FN is the numnber of false negative outcomes.

The ROC curves are obtained by varying the parameters of the IDS (in this case, various thresholds) over the same data set and by determining the corresponding values of FPR and TPR.

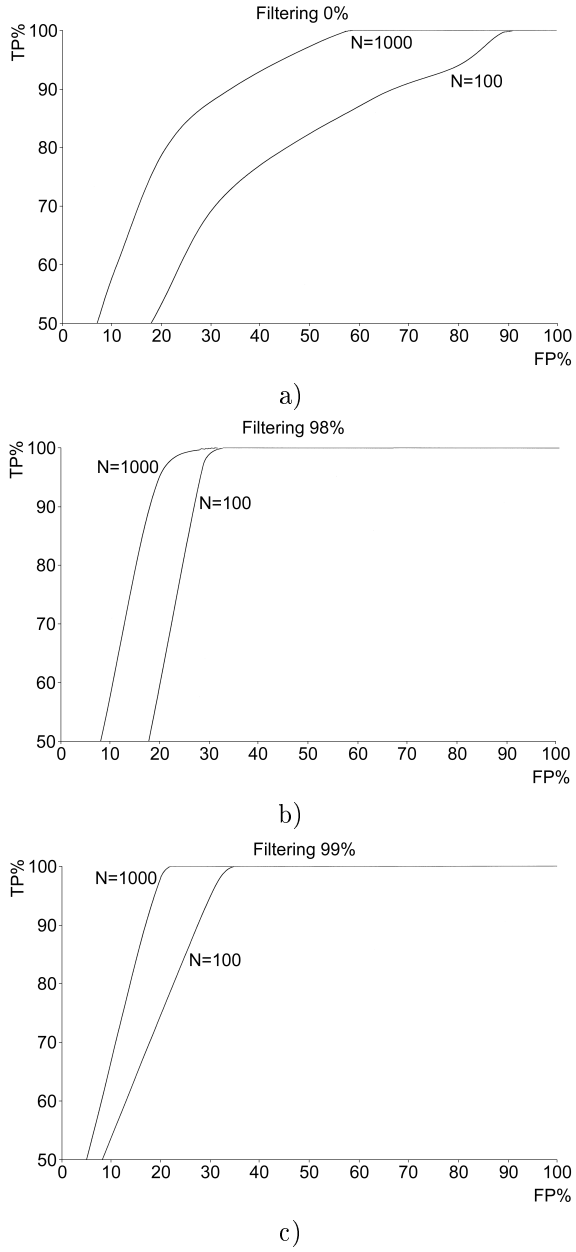

Fig. 3 - ROCs of the IDS with attack filtration: a) 0%, b) 98%, c) 99%

It can be seen from the Fig. 3 that the best results were achieved for $N = 1000$ and that the obtained results are comparable with those achieved by other IDS met in the literature, which is promising.

5 Conclusion

In this paper, a predictor-corrector intrusion detection system (IDS) is proposed, whose predictors are various clustering algorithms with different initial parameters that operate in parallel on the current set of resource access requests. The decisions whether abnormal behaviour is detected are made by a number of assessors, each of them implementing a different clustering evaluation algorithm in order to label the obtained clusters properly. The manager of the system estimates the quality of decision making from the pieces of information obtained a posteriori and then varies the parameters of predictors and/or assessors in order to achieve better overal performance. The cluster labeling problem was treated very carefully in order to overcome problems that usually arise in the IDS like this, such as the problem of detecting denial of service attacks, etc. With a combination of cluster cardinality labeling and graph theoretic labeling, promissing results expressed through the ROC curves have been obtained. These results show that very good overall performance can be achieved by selecting properly various system parameters.

References

1. J. Balasubramaniyan, J. García, D. Isacoff, E. Spafford and D. Zamboni, An Architecture for Intrusion Detection using Autonomous Agents, *Proceedings of the 14th Annual Computer Security Applications Conference*, IEEE Computer Society, 1998, pp. 13-24.
2. N. Bolshakova and F. Azuaje, Cluster Validation Techniques for Genome Expression Data, *Signal Processing*, 83, 2003, pp. 825-833.
3. G. Giacinto and F. Roli, Pattern Recognition for Intrusion Detection in Computer Networks, D Chen and X. Cheng (Eds.) *Pattern Recognition and String Matching*, Kluwer Academic Publishers, Dordrecht, 2002, pp. 187-209.
4. A. Jain, M. Murty and P. Flynn, Data Clustering: A Review, *ACM Computing Surveys*, Vol. 31, No. 3, 1999, pp. 264-323.
5. K. Julisch, Clustering Intrusion Detection Alarms to Support Root Cause Analysis, *ACM Transactions on Information and System Security*, Vol. 6, No. 4, 2003, pp. 443-471.
6. R. Lippman, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyshhogrod, R. Cunningham and M. Zissman, Evaluating Intrusion Detection Systems: the 1998 DARPA Off-line Intrusion Detection Evaluation, *Proceedings of the 2000 DARPA Information Survivality Conference and Exposition (DISCEX)*, Vol. 2, IEEE Press, January 2000.
7. L. Portnoy, E. Eskin and S. Stolfo, Intrusion Detection with Unlabeled Data Using Clustering, *Proceedings of the ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, Filadelfia, PA, November 5-8, 2001.