Labelling IDS Clusters by Means of the Silhouette Index

Abstract. One of the most difficult problems in the design of an anomaly based intrusion detection system (IDS) that uses clustering is that of labelling the obtained clusters, i.e. determining which of them correspond to ”good” behaviour on the network/host and which to ”bad” behaviour. In this paper, a new clusters' labelling strategy, which makes use of the Silhouette clustering quality index is proposed for application in such an IDS. The aim of the new labelling algorithm is to detect compact clusters containing very similar vectors and these are highly likely to be attack vectors. The effectiveness of a multiple classifier IDS with the Silhouette index implemented is compared to the effectiveness of a system employing a classical cardinality-based labelling strategy. Experimental results show that the system using the Silhouette index produces much more accurate results than the system that uses the classical cardinality-based labelling. Possibilities of improving the overall efficiency of an IDS using the new labelling algorithm are also discussed.

Key words: Intrusion detection system, Anomaly detection, Clustering, Silhouette index.

1 Introduction

Intrusion detection systems (IDS) are security tools designed to detect and classify attacks against computer networks and hosts. They can operate in two ways: either by searching for specific patterns in data (misuse based IDS) or by recognising certain deviations from expected behaviour (anomaly based IDS). In anomaly based IDS, clustering algorithms are often used for recognition of ”abnormal” behaviour, especially if many previously unknown attacks appear on the monitored network/host [12]. They can be applied either directly on incoming data [4, 7, 17] or as a supporting technique in a stage posterior to data classification performed by means of other techniques [10, 21].

Anomaly based IDS classify input data into a number of categories, or classes. This number can be arbitrary, but as the essential goal of these systems is to distinguish between ”normal” and ”abnormal” behaviour, it is very common to partition the incoming resource access requests into two classes that correspond to these two types of

behaviour. The data are submitted to the system in the form of lists created at predefined time intervals or alternatively, upon a predefined number of incoming requests. Then the system makes the decision about whether abnormal behaviour occurred or not, based on the obtained classification results.

In this paper, we consider a Denial-of-Service (DoS) attack scenario in which attack resource access requests arrive to the monitored network/host in bursts. An anomaly based IDS analyzes N resource access requests at a time and if it detects that more than $N/2$ of these requests correspond to attacks then it should generate a special alert. We call such a scenario a *massive attack*. Sometimes, other network monitoring tools (e.g. firewalls) can detect such attacks, but the advantage of an anomaly based IDS regarding all kinds of attacks (including massive attacks as defined in this paper) is in the capability of detecting a completely new attack.

If clustering is used for classification of resource access requests in an IDS, the main problem is the interpretation of clustering results, so called ”labelling” of clusters. Namely, without additional information it is difficult to decide whether the data classified in one cluster correspond to ”normal” behaviour in the monitored network or to ”abnormal” behaviour. Cardinalities of clusters are often used as a decision parameter for this purpose (see, for example, [17]) because the mathematical expectation of ”normal” behaviour is considered greater than that of ”abnormal” behaviour. However, this approach fails to detect massive attacks. Solving this problem requires a more complex clusters' labelling algorithm. A labelling strategy capable of outperforming the clusters' cardinality based labelling strategy has been proposed and analyzed in [16].

We analyze an alternative clusters' labelling strategy based on application of the Silhouette clustering evaluation index and clusters' silhouettes [18]. The goal of such a combination is to respond adequately to the properties of attack vectors. We consider the compactness of the corresponding clusters and the separation between them the principal parameters that distinguish ”normal” from ”abnormal” behaviour in the analysed network. The Silhouette index takes into account these parameters and because of that we apply it in our IDS. In the experiments, we test the response of a multiple classifier IDS (see, for example, [5]) with the new labelling strategy to artificial data. We express the IDS quality through Receiver Operating Characteristics (ROC) curves. The effectiveness of the IDS that uses the Silhouette index is compared with that of a system that uses a classical cardinality-based labelling algorithm.

In the experiments, we tested our labelling algorithm on the well known KDD CUP 1999 artificial data set [3, 11], which was used as the traffic source. Although this source has been criticized in the literature (see, for example, [13, 20]), it is still being used for IDS benchmarking [1, 6, 15]. We found it convenient as a source of massive attacks, against which we have tested our labelling strategy. The experimental results show that the labelling strategy that uses the Silhouette index gives much more accurate results than the strategy that employs the classical cardinality-based labelling, especially if massive attacks are present in the input data.

The problem of application of the Silhouette index labelling in an IDS is in its computational complexity, which is quadratic in the number of vectors involved in the clustering. However, by reducing the number of analyzed vectors at a time, it is possible to improve the overall efficiency of an IDS using such labelling.

The structure of the paper is the following: In Section 2, a general description of the analyzed intrusion detection system is given. In Section 3, the new clusters' labelling method employing the Silhouette clustering quality index is described in detail. 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 multiple classifier IDS, whose elements we analyze in this paper, consists of the following components (Fig. 1):

1. $\mathcal{K}$ sensors, $\mathcal{P}_1, \ldots \mathcal{P}_{\mathcal{K}}$, which operate in parallel on the same data set $\mathbf{X}_\tau$, $\tau = 0, 1, 2, \ldots$ We limit ourselves to the case in which every sensor is merely a clustering algorithm that classifies the input data set into K clusters, without any interpretation of clustering results.
2. $\mathcal{L}$ assessors, $\mathcal{A}_1, \ldots, \mathcal{A}_{\mathcal{L}}$, whose task is to "label" the clusters obtained from the sensors, upon 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 sensor over the data set $\mathbf{X}_\tau$. A local extreme value (maximum or minimum) of this function determines the decision of the assessor on the following: an element of $\mathbf{X}_\tau$ belongs to a cluster that is interpreted as one of the "normal" clusters or it belongs to a cluster that is interpreted as an "abnormal" one.
3. The manager of the system adjusts the parameters of the sensors and the assessors in order to maximize the effectiveness of the system as a whole.

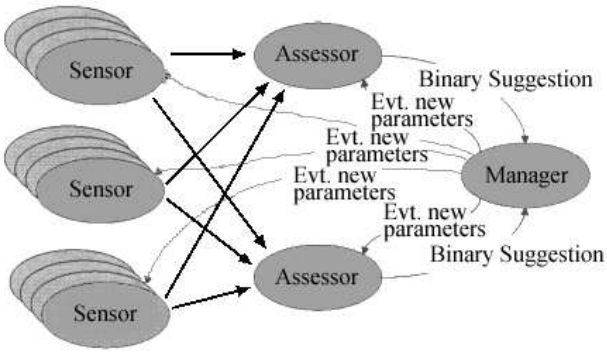


Fig. 1 - A multiple classifier IDS

In this paper, we concentrate on the basic sensor-assessor structure. The former actually performs the clustering of the incoming resource access requests, whereas the latter performs the clustering quality evaluation.

We have selected the well known K-means algorithm (see for example [9]) for implementation in the sensors of the IDS, because we consider this algorithm the best trade-off between accuracy and efficiency. The K-means algorithm is presented in the Fig. 2.

1. Initialization: Randomly choose *K* vectors from the data set and make them initial cluster centers.
2. Assignment: Assign each vector 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 input resource access requests are encoded in such a way that vectors of the same length are produced. The Euclidean metric, given by the following expression, is used in our system as a distance measure between vectors.

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^{n} \mid x_i - y_i \mid^2} \tag{1}$$

where n is the dimension of the vectors $\mathbf{X}$ and $\mathbf{Y}$.

3 The clusters' labelling algorithm

Having obtained clusters from the sensors, the task of the assessors is to label them, i.e. to determine which clusters correspond to "normal" behaviour, and which to "abnormal" behaviour. Since there is no learning on labelled data in the system, the assessors must use other criteria to decide on this. A common assumption is that few anomalies are expected in the clustering results, so a significant difference in cardinalities of the clusters naturally labels the cluster with the greatest cardinality as that corresponding to "normal" behaviour. However, there are at least two problems related to such a strategy [17]: 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*, that can mislead this labelling strategy by making the mathematical expectation of the attack much greater than that of a "normal" behaviour. To overcome the problems related to the labelling strategy described above, we propose the Silhouette clustering evaluation index to be used in the assessors of the IDS. This index opts for detecting well separated and compact clusters.

In our IDS assessing algorithm, the global Silhouette index of the clustering is combined with the comparison of the silhouettes of the clusters. We now present the formal definition of the Silhouette index.

Let $\mathbf{X}_\tau = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be the data set and let $\mathcal{C} = (C_1, \ldots, C_K)$ be its clustering into K clusters. Let $d(\mathbf{X}_k, \mathbf{X}_l)$ be the distance between $\mathbf{X}_k$ and $\mathbf{X}_l$. Let $\mathcal{C}_j = \{\mathbf{X}_1^j, \ldots, \mathbf{X}_{m_j}^j\}$ be the j-th cluster, $j = 1, \ldots, K$, where $m_j = \mid \mathcal{C}_j \mid$. The average distance a_i^j between the i-th vector in the cluster $\mathcal{C}_j$ and the other vectors in the same cluster is given by the following expression [2, 8, 18]:

$$a_i^j = \frac{1}{m_j - 1} \sum_{\substack{k=1 \\ k \neq i}}^{m_j} d(\mathbf{X}_i^j, \mathbf{X}_k^j), \quad i = 1, \ldots, m_j. \tag{2}$$

The minimum average distance between the i-th vector in the cluster $\mathcal{C}_j$ and all the vectors clustered in the clusters $\mathcal{C}_k$, $k = 1, \ldots, K$, $k \neq j$ is given by the following expression:

$$b_i^j = \min_{\substack{n=1,\ldots,K \\ n \neq j}} \left\{ \frac{1}{m_n} \sum_{k=1}^{m_n} d(\mathbf{X}_i^j, \mathbf{X}_k^n) \right\}, \quad i = 1, \ldots, m_j. \tag{3}$$

Then the silhouette width of the i-th vector in the cluster $\mathcal{C}_j$ is defined in the following way:

$$s_i^j = \frac{b_i^j - a_i^j}{\max\left\{a_i^j, b_i^j\right\}} \tag{4}$$

From the expression (4), it follows that $-1 \leq s_i^j \leq 1$. We can now define the silhouette of the cluster $\mathcal{C}_j$:

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} s_i^j \tag{5}$$

Finally, the global Silhouette index of the clustering is given by:

$$S = \frac{1}{K} \sum_{j=1}^{K} S_j \tag{6}$$

It is easy to see that both a cluster's silhouette and the global silhouette take values between -1 and 1 (both inclusive).

In the experiments, we compare the results obtained with our assessing algorithm with the results obtained with the clusters' cardinalities criterion, a common measure for assessing IDS clusters (see for example [17]). We define this criterion in the following way:

Let $\mathbf{X}_\tau = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be the current data set and let $\mathcal{C}_{k,\tau} = \{\mathbf{Y}_{k,\tau}, \mathbf{Z}_{k,\tau}\}$ be the partition of $\mathbf{X}_\tau$ into 2 clusters, obtained in the sensor $\mathcal{P}_k$. 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 a threshold 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.

For the remainder of this paper, we shall limit ourselves to studying the case with 2 clusters, of which one corresponds to "normal" and the other to "abnormal" behaviour in the corresponding network. The reason for this is that, whatever the number of clusters we use in the sensors, we must finally decide which of them will be considered "normal", leading us to a case with 2 "superclusters".

The main idea of our clusters' labelling algorithm, which uses the Silhouette clustering quality index is the following:

The attack vectors are often mutually very similar, if not identical. Because of that, we expect that the attack cluster in the case of a massive attack will be extremely compact. The value of the Silhouette index of such a clustering is either 1 or very close to

1. Having in mind the expected mutual similarity among attack vectors, the silhouette of the attack cluster is expected to be greater than the silhouette of the non-attack cluster. The case in which one of the clusters is empty must be treated in a special way: since the Silhouette index is used, relabeling of the clustering should be performed if the value of the global Silhouette index is -1 and the cluster labelled with "2" (the label reserved for the attack cluster) is empty. Higher values of the global Silhouette index indicate the presence of a massive attack, whereas higher values of clusters' silhouettes indicate attack clusters.

By contrast, when the global Silhouette index takes lower values, i.e. when there is no massive attack, the silhouette of the non attack cluster (labelled with "1") is expected to be higher than the silhouette of the attack cluster (labelled with "2"). This is because isolated attacks (non-massive) are expected to be less similar among themselves.

Example 1: In the KDD CUP 1999 data set, many attack vectors correspond to the so called "smurf" attack, which is a sort of DoS attack. Table 1 shows the differences between the coordinates of two attack vectors that correspond to the "smurf" attack. Table 2 shows the differences between two "normal" vectors. In this particular example it is easy to see that the difference between two attack vectors is much smaller than the difference between two "normal" vectors.

The study above gives rise to the following labelling algorithm:

Algorithm 1 - the Silhouette index labelling algorithm
Input:

- A clustering $\mathcal{C}$ of N vectors into 2 clusters, C_1 and C_2, in which the vectors assigned to the "non-attack" cluster C_1 take the label "1", and those assigned to the "attack" cluster C_2 take the label "2".
- The global Silhouette index threshold, Δ_S.
- The clusters' silhouette thresholds, Δ_{S_1} and Δ_{S_2}.

Output:

- The eventually relabelled input clustering, if relabelling conditions are met.

begin
$S \longleftarrow GlobalSilhouetteIndex(\mathcal{C})$;
$s_1 \longleftarrow Silhouette(C_1)$;
$s_2 \longleftarrow Silhouette(C_2)$;
/** Condition 1 **/
if $(S = -1)$ **and** $(IsEmpty(C_2))$ **then**
$Relabel(\mathcal{C})$
/** Condition 2 **/
else if $(S < \Delta_S)$ **and** $(s_1 < s_2 + \Delta_{S_1})$ **then**
$Relabel(\mathcal{C})$
/** Condition 3 **/
else if $(S > \Delta_S)$ **and** $(s_1 + \Delta_{S_2} > s_2)$ **then**
$Relabel(\mathcal{C})$;
end.

The relabelling procedure is given below:

procedure $Relabel$(Clustering $\mathcal{C}$)
begin
  forall vectors in $\mathcal{C}$
    if label of a vector is '1', set it to '2'
      and vice versa ;
end.

Example 2: The Algorithm 1 applied to the first 20000 records of the reduced (10%) KDD CUP data set, upon clustering by the 2-means algorithm where $N = 1000$, produces no labelling errors in spite of a very bad as-clustered labelling on average. The parameters of the Algorithm 1 are the following: $\Delta_S = 0.8$, $\Delta_{S_1} = 0.0001$ and $\Delta_{S_2} = 0.0001$. The results are summarized in the Table 3.

Table 1. The differences between two attack vectors in the KDD CUP 1999 data base (records 7635 and 7636 of the reduced (10%) data set). The rest of 41 coordinates are equal to 0.

Coord. id.	Rec. 7635	Rec. 7636
protocol_type	2	2
service	50001	50001
flag	10	10
src_bytes	1032	1032
count	511	511
srv_count	511	511
same_srv_rate	100	100
dst_host_count	228	238
dst_host_srv_count	83	93

Table 2. The differences between two "normal" vectors in the KDD CUP data base (records 6 and 7 of the reduced (10%) data set). The rest of 41 coordinates are equal to 0.

Coord. id.	Rec. 6	Rec. 7
service	80	80
flag	10	10
src_bytes	212	159
dst_bytes	1940	4087
logged_in	1	1
count	1	5
srv_count	2	5
same_srv_rate	100	100
srv_diff_host_rate	100	0
dst_host_count	1	11
dst_host_srv_count	69	79
dst_host_same_srv_rate	100	100
dst_host_same_src_port_rate	100	0

Table 3. Application of the Algorithm 1 to the first 20000 records of the reduced (10%) KDD CUP 1999 data set. The first row of the table corresponds to the records 1-1000 of the KDD CUP 1999 data base, the second row corresponds to the records 1001-2000 and so on.

S index	s_1	s_2	No. of attacks	Relab. cond.*
0.611	0.368	0.854	0	2
0.663	0.482	0.845	0	2
0.678	0.503	0.853	0	2
0.666	0.403	0.93	2	2
0.499	0.089	0.908	0	2
0.643	0.828	0.459	0	0
0.661	0.958	0.364	0	0
0.956	0.999	0.913	376	3
-1.0	-1.0	e	1000	1
-1.0	-1.0	e	1000	1
-1.0	-1.0	e	1000	1
0.953	0.998	0.907	321	3
0.582	0.290	0.875	0	2
0.627	0.863	0.391	0	0
0.568	0.160	0.977	0	2
0.589	0.428	0.750	21	2
0.614	0.397	0.831	0	2
0.592	0.849	0.335	0	0
0.621	0.356	0.885	0	2
0.945	0.895	0.995	99	0

* See Algorithm 1; 0 means that the initial (as clustered) labelling is correct so no relabelling is performed by the Algorithm 1

^{e} Empty cluster

The behaviour of the Algorithm 1 depends on the choice of the parameters. These should be determined in advance. One of the ways to do that is to use a network/dataset with known relevant characteristics, of which the most important ones are the base rate, i.e. the probability of the attack and the type of the attack. For example, the KDD CUP 1999 data set has the base rate of $\approx 80\%$ and of all the attacks in that dataset $\approx 99\%$ are denial-of-service (DoS) attacks [19]. For a dataset with such characteristics, the best performance of the Algorithm 1 has been obtained by setting the values of the parameters to those used in the Example 2. In a real network, one could start with the parameters of the labelling algorithm obtained in a controlled network scenario (e.g. with those obtained with the KDD CUP 1999 database) and then fine tune the parameters over time.

4 Experimental work

Extensive simulation of the basic sensor-assessor structure of a multiple classifier IDS has been carried out in order to study its response to the attack data. To this end, the following instance of this structure has been built:

1. In the sensor, the 2-means clustering algorithm has been implemented.
2. Two types of assessors have been tested:
   2.1 The assessor implementing the Silhouette index of the clustering and the silhouettes of the clusters, according to the Algorithm 1. The global Silhouette index threshold, Δ_S, and the clusters' silhouette thresholds, Δ_{S_1} and Δ_{S_2}, have been used as parameters of the assessing algorithm.
   2.2 Cardinality based assessor: the cluster of greater cardinality is considered "normal" and is labelled with '1'. The minimum difference $\mathcal{D}_C$ between clusters' cardinalities needed to relabel the clustering is used as a parameter of this assessing algorithm.

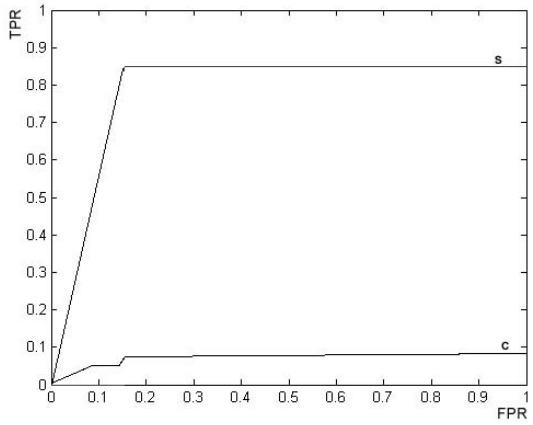
Next, an input data source had to be selected. According to [14], this is one of the challenges of IDS testing. In [14], 4 approaches to this problem are defined, according to the use of background traffic in the test data:

1. testing using no background traffic at all;
2. testing using real traffic/logs;
3. testing using sanitized traffic/logs;
4. testing by generating traffic on a tested network.

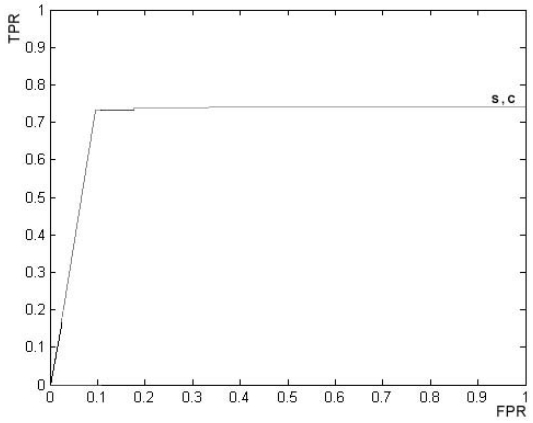
The main advantage of testing by generating traffic artificially is the possibility of accurate determination of the number of false alarms, since no unknown attacks can appear in the test data. The quality of such a simulated data source is a separate question. For example, the well known and widely used KDD CUP 1999 source [3, 11] has been criticized by various authors (see [13, 20], among others). However, the KDD CUP data set contains many massive attacks (which is typical for a military environment to which it corresponds) and this is a decisive characteristic needed for testing the labelling strategy proposed in this paper.

Thus, we have selected the KDD CUP 1999 database as the traffic source for our experiments. The aim was to compare the results obtained by applying the two variants of the proposed labelling strategy, with and without the presence of massive attacks. Because of that, the attacks from the KDD CUP database were filtered out in the same way as in [17]. The filtering percentage of 0%, 98% and 99% was used over all the resource access request records of the database. Without filtering out the attacks (0%), the database simulates many massive attacks, whereas if the filtering of 98% and 99% of attacks is applied it simulates a situation in which attacks are rare events. The effectiveness of the system was measured by means of the ROC (Receiver Operating Characteristic) curves for the filtered data set mentioned above. A ROC curve depicts the relationship between false positive rate FPR and true positive rate TPR, where:

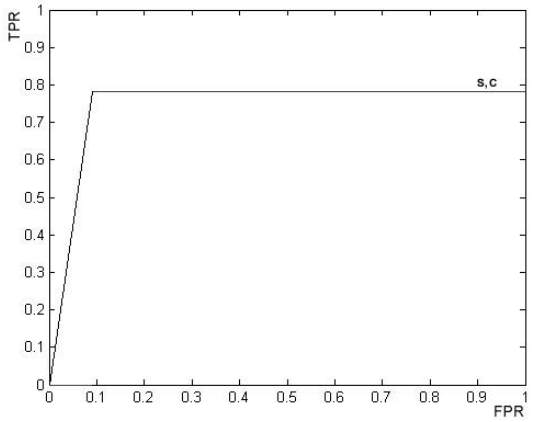
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \qquad \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$



a)



b)



c)

Fig. 3 - ROC curves of the IDS. S - labelling using the Silhouette index; C - labelling using clusters' cardinalities. Attack filtration: a) 0%, b) 98%, c) 99%

In the equation (7), FP is the number of false positive outcomes of the intrusion detection on a fixed data set, i.e. the number of decisions in which a non-existing attack is signalled, TP is the number of true positive outcomes, i.e. successful detections, TN is the number of true negative outcomes, i.e. the number of decisions, in which a non-existing attack is not signalled, and FN is the number of false negative outcomes, i.e. the number of decisions, in which an existing attack is not signalled.

The results concerning the effectiveness of the IDS using the Algorithm 1 are compared with those obtained using the classical cardinality-based labelling. The comparative results are presented in the Fig. 3. The best results with the Algorithm 1 over the KDD CUP '99 database were obtained with $\Delta_{S_1} = \Delta_{S_2} = 0.0001$.

The ROC curves labelled with S from the Fig. 3 were obtained by setting $\Delta_{S_1} = \Delta_{S_2} = 0.0001$ and by varying the global Silhouette index threshold Δ_S between 0.6 and 0.9. The cardinality of the data set for clustering was $N = 1000$ in all the experiments.

From the Fig. 3, it can be seen that without attack filtering (Fig. 3a), the Algorithm 1 gives much better results than the cardinality-based labelling, which is in this case completely useless. With 98% as well as 99% of the attacks from the KDD CUP 1999 database filtered out (Fig. 3b and 3c, respectively), the results obtained with the Algorithm 1 are approximately the same as those obtained with the cardinality-based labelling. This means that the labelling in the presence of massive attacks is solved in a satisfactory way with the Algorithm 1, whereas the correctness of the IDS decisions without the presence of massive attacks is not deteriorated.

However, the time complexity of the Silhouette index computation is quadratic in the number of vectors involved in the clustering. One way of improving the overall efficiency of an IDS that uses the Silhouette index for labelling is to reduce the number of processed vectors at a time, N. In that case, the accuracy of the results is deteriorated to some extent, but the overall efficiency is signifficantly improved.

5 Conclusion

In this paper, a new clusters' labelling strategy has been proposed for application in a multiple classifier intrusion detection system (IDS). That strategy combines the computation of the Silhouette clustering quality index and the comparison of silhouettes of the clusters. The aim of the new labelling algorithm is to detect compact clusters containing very similar vectors that are highly likely to be attack vectors. The response of an IDS using such a labelling strategy to a massive attack (for example, a Denial-of-Service attack) was tested. In the experiments, the KDD CUP 1999 database has been used as the traffic source, in spite of all the criticism, because it is a good source of massive attacks. It was shown experimentally, via ROC curves obtained by applying the IDS over the KDD CUP 1999 database, that in the presence of massive attacks the labelling algorithm that uses the Silhouette index produces much more accurate results than the one that uses the classical cardinality-based labelling. However, the time complexity of the Silhouette index computation is much greater than the time complexity of the cardinality-based labelling. The overall performance of a system with the Silhou-

ette index used for labelling may be improved by reducing the number of input vectors processed at a time, without sacrifising much of the system's accuracy.

References

1. Ben Amor N., Benferhat S. and Elouedi Z., "Naive Bayes vs. Decision Trees in Intrusion Detection Systems", *Proceedings of the 19th ACM Symposium on Applied Computing (SAC2004), ACM*, Nicosia, Cyprus, 2004, pp. 420-424.
2. Bolshakova N. and Azuaje F., "Cluster Validation Techniques for Genome Expression Data", *Signal Processing*, 83, 2003, pp. 825-833.
3. Elkan C., "Results of the KDD'99 Classifier Learning", *ACM SIGKDD Explorations*, Vol. 1, No. 2, 2000, pp. 63-64.
4. Frank J., "Artificial Intelligence and Intrusion Detection: Current and Future Directions", *Proceedings of the 17th National Computer Security Conference*, Baltimore, USA, 1994.
5. Giacinto G. and Roli F., "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.
6. Gomez J., Gonzalez F. and Dasgupta D., "An Imuno-Fuzzy Approach to Anomaly Detection", *Proceedings of the 12th IEEE International Conference on Fuzzy Systems (FUZZIEEE)*, Vol. 2, St. Louis, USA, May 2003, pp. 1219-1224.
7. Guan Y., Ghorbani A. and Belacel N., "Y-Means: a Clustering Method for Intrusion Detection", *Proceedings of Canadian Conference on Electrical and Computer Engineering*, Montreal, Canada, 2003.
8. Günter S. and Bunke H., "Validation Indices for Graph Clustering", J. Jolion, W. Kropatsch, M. Vento (Eds.) *Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, CUEN Ed., Italy, 2001, pp. 229-238.
9. Jain A., Murty M. and Flynn P., "Data Clustering: A Review", *ACM Computing Surveys*, Vol. 31, No. 3, 1999, pp. 264-323.
10. Julisch K., "Clustering Intrusion Detection Alarms to Support Root Cause Analysis", *ACM Transactions on Information and System Security*, Vol. 6, No. 4, 2003, pp. 443-471.
11. Lippman R., Haines J., Fried D., Korba J. and Das K., "The 1999 DARPA off-line intrusion detection evaluation", Computer Networks, 34, 2000, pp. 579-595.
12. Laskov P., Düssel P., Schäfer C. and Rieck K., "Learning Intrusion Detection: Supervised or Unsupervised", *Proceedings of the 13th International Conference on Image Analysis and Processing (ICIAP 2005)*, Cagliari, Italia, 2005, pp. 50-57.
13. McHugh J., "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory", *ACM Trans. on Information and System Security*, Vol. 3, No. 4, November 2000, pp. 262-294.
14. Mell P., Hu V., Lippman R., Haines J. and Zissman M., "An Overview of Issues in Testing Intrusion Detection Systems", *NIST interagency report 7007*, June 2003.
15. Ozyer T., Ahlaj L. and Barker K., "A Boosting Genetic Fuzzy Classifier for Intrusion Detection Using Data Mining Techniques for Rule Pre-Screening", *Design and Application of Hybrid Intelligent Systems*, 2003, pp. 983-992.
16. Petrović S., Álvarez G., Orfila A. and Carbó J., "Labelling Clusters in an Intrusion Detection System Using a Combination of Clustering Evaluation Techniques," *Proceedings of the 39th Hawaii International Conference on System Sciences* (CD ROM), (8 pages), IEEE Computer Society Press, 2006.
17. Portnoy L., Eskin E. and Stolfo S., "Intrusion Detection with Unlabeled Data Using Clustering", *Proceedings of the ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, Philadelphia, PA, November 5-8, 2001.

18. Rousseeuw P., ”Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis”, *J. Comput. Appl. Math.*, 20, 1987, pp. 53-65.
19. Sabhnani S. and Serpen G., ”Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context”, *Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA-2003)*, Las Vegas, USA, June 2003, 209-215.
20. Taylor C. and Alves-Foss J., ”An Empirical Analysis of NATE - Network Analysis of Anomalous Traffic Events”, *Proceedings of the 2002 workshop on New security paradigms*, Virginia Beach, Virginia, September 2002, pp. 18-26.
21. Tölle J. and Niggemann O., ”Supporting Intrusion Detection by Graph Clustering and Graph Drawing”, *Proceedings of 3rd International Workshop on Recent Advances in Intrusion Detection (RAID 2000)*, Toulouse, France, 2000.