Clustering for Intrusion Detection: Network Scans as a Case of Study

Raúl Sánchez[1], Álvaro Herrero[1], and Emilio Corchado[2]

[1]Department of Civil Engineering, University of Burgos, Spain
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio, rsarevalo}@ubu.es

[2]Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008 Salamanca, Spain
escorchado@usal.es

Abstract. MOVICAB-IDS has been previously proposed as a hybrid intelligent Intrusion Detection System (IDS). This on-going research aims to be one step towards adding automatic response to this visualization-based IDS by means of clustering techniques. As a sample case of study for the proposed clustering extension, it has been applied to the identification of different network scans. The aim is checking whether clustering and projection techniques could be compatible and consequently applied to a continuous network flow for intrusion detection. A comprehensive experimental study has been carried out on previously generated real-life data sets. Empirical results suggest that projection and clustering techniques could work in unison to enhance MOVICAB-IDS.

Keywords: Network Intrusion Detection, Computational Intelligence, Exploratory Projection Pursuit, Clustering, Automatic Response.

1 Introduction

The ever-changing nature of attack technologies and strategies is one of the most harmful issues of attacks and intrusions, which increases the difficulty of protecting computer systems. For that reason, among others, Intrusion Detection Systems (IDSs) [1-3] have become an essential asset in addition to the computer security infrastructure of most organizations.

In the context of computer networks, an IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack. Intrusion Detection (ID) is therefore a field that focuses on the identification of attempted or ongoing attacks on a computer system (Host IDS - HIDS) or network (Network IDS - NIDS).

MOVICAB-IDS (MObile VIsualisation Connectionist Agent-Based IDS) has been proposed [4, 5] as a novel IDS comprising a Hybrid Artificial Intelligent System (HAIS). It monitors the network activity to identify intrusive events. This hybrid intelligent IDS combines different AI paradigms to visualise network traffic for ID at packet level. Its main goal is to provide security personnel with an intuitive and informative visualization of network traffic to ease intrusion detection. The proposed MOVICAB-IDS applies an unsupervised neural projection model to extract interesting traffic dataset projections and to display them through a mobile visualisation interface

A port scan may be defined as a series of messages sent to different port numbers to gain information on their activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services the host is providing. A port scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity. This work focuses on the identification of network scans, in which the same port is the target for a number of computers. A network scan is one of the most common techniques used to identify services that might then be accessed without permission [6]. Because of that, the proposed extension of MOVICAB-IDS is faced up with this kind of simple but usual situations.

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis.

The remaining sections of this study are structured as follows: section 2 introduces the proposed framework and applied models and techniques. Experimental results are presented in section 3 while the conclusions of this study are discussed in section 4, as well as future work.

2 On the Network Data Visualization and Analysis

The general framework for the proposed projection-based intrusion detection taking part in MOVICAB-IDS is depicted in Fig. 1.

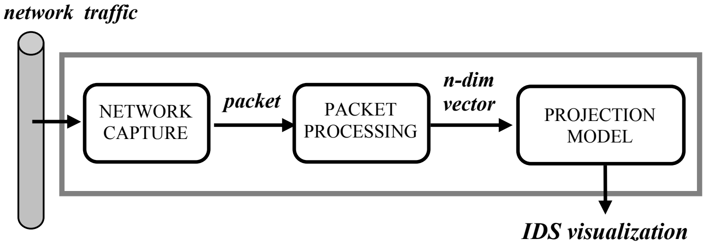


Fig. 1. MOVICAB-IDS general architecture.

This framework could be described as follows:

- packets traveling through the network are intercepted by a **capture device**;
- traffic is **coded by a set of features** spanning a multidimensional vector space;
- a **projection model** operates on feature vectors and yields as output a suitable representation of the network traffic. The projection model clearly is the actual

core of the overall IDS. That module is designed to yield an effective and intuitive representation of network traffic, thus providing a powerful tool for the security staff to visualize network traffic.

Present work focuses on the upgrading of the previous framework, to incorporate now new facilities as depicted in Fig. 2. It is now required and enhanced visualization by combining projection and clustering results to ease traffic by personnel. By doing so, further information on the nature of the travelling packets could be compressed in the visualization. On the other hand automatic response is an additional feature of some IDSs that could be incorporated in MOVICAB-IDS. Additionally to some classifiers for the automatic detection, clustering is proposed for those cases in which classifiers do usually fail (0-day attacks for example).

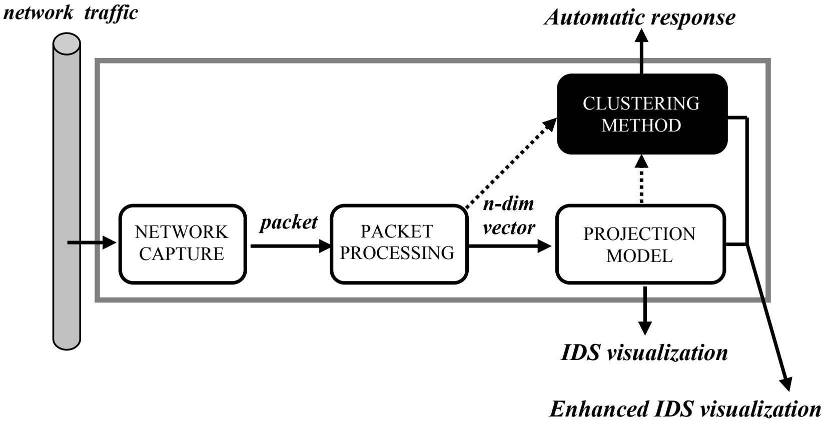


Fig. 2. Clustering extension of MOVICAB-IDS.

The following subsections describe the different techniques that take part in the proposed solution. For the dimensionality reduction as a projection method, Cooperative Maximum Likelihood Hebbian Learning [7] is explained as it proved to be the most informative one among many considered [5]. It is described in section 2.1. On the other hand, to test clustering performance some of the standard methods have been tested, namely: *k*-means and agglomerative clustering. They are described in sections 2.2 and 2.3 respectively.

2.1 Cooperative Maximum Likelihood Hebbian Learning

The standard statistical method of Exploratory Projection Pursuit (EPP) [8] provides a linear projection of a data set, but it projects the data onto a set of basis vectors which best reveal the interesting structure in data; interestingness is usually defined in terms of how far the distribution is from the Gaussian distribution.

One neural implementation of EPP is Maximum Likelihood Hebbian Learning (MLHL) [9], [10]. It identifies interestingness by maximising the probability of the residuals under specific probability density functions which are non-Gaussian.

One extended version of this model is the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [7] model. CMLHL is based on MLHL [9], [10] adding lateral connections [7], [11] which have been derived from the Rectified Gaussian Distribution [12]. The resultant net can find the independent factors of a data set but does so in a way that captures some type of global ordering in the data set.

Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed [7], [11] as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \forall i \ . \tag{1}$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \ . \tag{2}$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \ . \tag{3}$$

4. Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) | e_j |^{p-1} \ . \tag{4}$$

Where: η is the learning rate, τ is the "strength" of the lateral connections, b the bias parameter, p a parameter related to the energy function [9], [10], [7] and A a symmetric matrix used to modify the response to the data [7]. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.2 Clustering

Cluster analysis [13] is the organization of a collection of data items or patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Hence, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. This notion of similarity can be expressed in very different ways.

Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various communities [14], [15], [16]. A simple distance measure such as the Euclidean distance is often used to reflect dissimilarity between two patterns, whereas other similarity measures can be used to characterize the conceptual similarity between patterns [17], it depends on the type of data we want to analyse. Furthermore, the clustering output can be hard (allocates each pattern to a single cluster) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters). A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership.

There are different approaches to clustering data [13], [15], but given the high number and the strong diversity of the existent clustering methods, we have focused on the ones shown in Figure 3 based on the suggestions in [13].

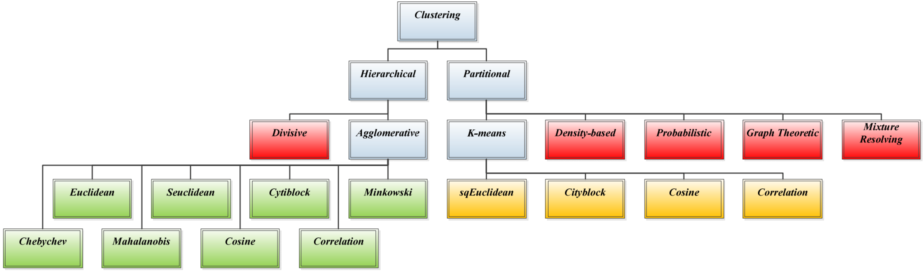


Fig. 3. Clustering methods used on this paper: one hierarchical (Agglomerative) and other partitional method (K-means).

Hierarchical methods generally fall into two types:

1. Agglomerative: an agglomerative approach begins with each pattern in a distinct cluster, and successively joins clusters together until a stopping criterion is satisfied or until a single cluster is formed.
2. Divisive: a divisive method begins with all patterns in a single cluster and performs splitting until a stopping criterion is met or every pattern is in a different cluster. This method is neither applied nor discussed in this paper.

Partitional clustering aims to directly obtain a single partition of the data instead of a clustering structure, such as the dendrogram produced by a hierarchical technique.

There is no clustering technique that is generally applicable in the clustering of the different structures presented in multidimensional data sets. Humans can competitively and automatically cluster data in two dimensions, but most real problems involve clustering in a higher dimensional space, that is the case of network security data sets. It is difficult to obtain an intuitive interpretation of data in those spaces.

Since similarity is fundamental to the definition of a cluster, a measure of the similarity is essential to most clustering methods and it must be chosen carefully. We will focus on the well-known distance measures used for patterns whose features are all continuous:

Table 1. Some of the well-known distance measures that are usually employed in clustering methods.

Metric	**Description**
Euclidean	Euclidean distance: $D_{ab} = \sqrt{\sum_{j=1}^{p} \left(x_{aj} - x_{bj}\right)^2}$ Where: • x_{aj} , x_{bj} values taken by the j^{th} variable for the objects *a* and *b*, respectively in the multi-variable space. • *p* number of dimensions.

sEuclidean	Standardized Euclidean distance. Each coordinate difference between rows in X is scaled by dividing by the corresponding element of the standard deviation.
Cityblock	City block metric also known as Manhattan distance: $D_{ab} = \sum_{j=1}^{p} \left\| x_{aj} - x_{bj} \right\|$ Where: • x_{aj} , x_{bj} values taken by the j^{th} variable for the objects a and b, respectively in the multi-variable space. • p number of dimensions.
Minkowski	Minkowski distance: $D_{ab} = \sqrt[\lambda]{\sum_{j=1}^{p} \left\| x_{aj} - x_{bj} \right\|^{\lambda}}$ • x_{aj} , x_{bj} values taken by the j^{th} variable for the objects a and b, respectively in the multi-variable space. • p number of dimensions. • λ=1 Cityblock distance. • λ=2 Euclidean distance.
Chebychev	Chebychev distance (maximum coordinate difference).
Mahalanobis	Mahalanobis distance, using the sample covariance of X: $D_{ab} = \sqrt{(x_a - x_b)^T S^{-1} (x_a - x_b)}$ • x_a , x_b values of the objects a and b, respectively in the multi-variable space. • S covariance matrix.
Cosine	One minus the cosine of the included angle between points (treated as vectors).
Correlation	One minus the sample correlation between points (treated as sequences of values).

The most popular metric for continuous features is the Euclidean distance which is a special case of the Minkowski metric (p =2). It works well when a data set has compact or isolated clusters [18]. The problem of using directly the Minkowski metrics is the tendency of the largest-scaled feature to dominate the others. Solutions to this problem include normalization of the continuous features (sEuclidean distance).

Linear correlation among features can also distort distance measures, it can be relieved by using the squared Mahalanobis distance that assigns different weights to different features based on their variances and pairwise linear correlations. The regularized Mahalanobis distance was used in [18] to extract hyperellipsoidal clusters.

2.2.1 *k*-means

K-means is the simplest and most commonly used partitional algorithm employing a squared error criterion [19], but it also can be used with other distance measures. It starts with a random initial partition of *k* clusters (centroids: is the point to which the sum of distances from all objects in that cluster is minimized) and assign the patterns to clusters based on the similarity between the pattern and the centroid until a convergence criterion is met (e.g. minimize the sum of point-to-centroid distances, summed over all *k* clusters). The *k*-means algorithm is popular because it is easy to implement, and its time complexity is O(n), where n is the number of patterns to cluster. The main problem of this algorithm is that it is sensitive to the selection of the initial partition and may conclude with a local minimum (not a global minimum) depending on the initial partition.

This study uses four different distance measures, the method have been tested on all of them and the best result can be seen on the results section.

Table 2. Distance measures employed for K-means in this study.

Metric	Description
sqEuclidean	Squared Euclidean distance. Each centroid is the mean of the points in that cluster.
Cityblock	Sum of absolute differences. Each centroid is the component-wise median of the points in that cluster.
Cosine	One minus the cosine of the included angle between points (treated as vectors). Each centroid is the mean of the points in that cluster, after normalizing those points to unit Euclidean length.
Correlation	One minus the sample correlation between points (treated as sequences of values). Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.

2.2.2 Agglomerative

A hierarchical algorithm produces a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. The dendrogram can be broken at different levels to produce different clusterings of the data. The hierarchical agglomerative clustering algorithm has three phases:

1. *First phase*: compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.
2. *Second phase*: find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.
3. *Third phase*: if all patterns are in one cluster, stop. Otherwise, go to step 2.

Based on the way the proximity matrix is updated in the second phase, a variety of linking methods can be designed (this study has been developed with the linking methods shown in Table 3).

Table 3. Linkage functions employed for agglomerative clustering in this study.

Method	Description
Single	Shortest distance. $d'(k,\{i,j\}) = \min\{d(k,i), d(k,j)\}$
Complete	Furthest distance. $d'(k,\{i,j\}) = \max\{d(k,i), d(k,j)\}$
Ward	Inner squared distance (minimum variance algorithm), appropriate for Euclidean distances only.
Median	Weighted center of mass distance (WPGMC: Weighted Pair Group Method with Centroid Averaging), appropriate for Euclidean distances only.
Average	Unweighted average distance (UPGMA: Unweighted Pair Group Method with Arithmetic Averaging).
Centroid	Centroid distance (UPGMC: Unweighted Pair Group Method with Centroid Averaging), appropriate for Euclidean distances only.
Weighted	Weighted average distance (WPGMA: Weighted Pair Group Method with Arithmetic Averaging).

3 Experimental Results

This section describes the dataset used for evaluating the proposed clustering methods and how they were generated. Then, the obtained results are also detailed.

3.1 Datasets

Five features were extracted from packet headers to form the data set:

- **Timestamp**: the time difference in relation to the first captured packet. Sequential integer nonlinear [0:262198].
- **Source Port**: the port of the source host from where the packet is sent. Discrete integer values {53, ..., 5353}.
- **Destination Port**: the port of the destination host to where the packet is sent. Discrete integer values {53, ..., 36546}.
- **Size**: total packet size (in Bytes). Discrete integer values {60, ..., 355}.
- **Protocol ID**: we have used values between 1 and 35 to identify the packet protocol. Discrete integer values {65, ..., 112}.

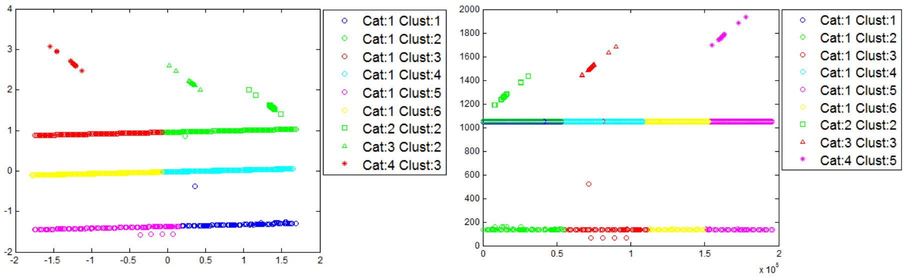
As an initial experiment to enhance MOVICAB-IDS detection capabilities, clustering techniques have been applied to a simple dataset containing three network scans aimed at port numbers 161, 162 and 3750. Additionally, it contains a great background of network traffic that may be considered as "normal".

As previously used in other experiments, further details on the data can be found in [4, 5].

3.2 Results

The best results obtained by applying the previously introduced techniques to the described datasets are shown in this section. The results are projected through CMLHL and further information about the clustering results is added to the projections, mainly by the glyph metaphor (colors and symbols). All the projections comprise a legend that states the color and symbol used to depict each packet, according to the original category: normal (Cat. 1), scan #1 (Cat. 2), scan #2 (Cat. 3) or scan #3 (Cat. 4), and the assigned cluster (Clust.).

Initially, the well-known *k*-means algorithm has been applied several times to the data by combining the different algorithm options. Best results are shown in Fig. 4.



4.a *k*-means on projected data (*k*=6 and sqEuclidean distance).

4.b *k*-means on original data (*k*=6 and sqEuclidean distance).

Fig. 4. Best clustering result through *k*-means under the frame of MOVICAB-IDS.

From Fig. 4 it can be seen that all the packets in each one of the scans (represented as non-horizontal small bars) are clustered in the same group. However, some other packets, regarded as normal, have been also included in those clusters. Apart from these two projections, some more experiments have been conducted, whose details (performance, true positive and false positive rates, values of *k* parameter, etc.) can be seen in Table 4.

Table 4. K-means experiments with different conditions.

Data	k	Distance criteria	False Positive	False Negative	Replicates/ Iterations	Sum of Distances
Projected	2	sqEuclidean	48,0186 %	0 %	5/4	1705,77
Original	2	sqEuclidean	46.6200 %	2.0979 %	5/5	9,75E+11
Projected	4	sqEuclidean	22,9604 %	0 %	5/8	643,352
Original	4	sqEuclidean	69,1143 %	0 %	5/8	4,38E+11
Projected	6	sqEuclidean	22,9604 %	0 %	5/8	301,218
Original	6	sqEuclidean	45,4545 %	0 %	5/24	2,91E+11
Projected	2	Cityblock	46,2704 %	0 %	5/7	1380,1
Original	2	Cityblock	49.6503 %	2.0979 %	5/9	3,50E+07
Projected	4	Cityblock	22,9604 %	0 %	5/8	710,545
Original	4	Cityblock	72,0249 %	0 %	5/15	2,15E+07
Projected	6	Cityblock	22,9604 %	0 %	5/14	526,885
Original	6	Cityblock	48,0187 %	0 %	5/10	1,41E+07
Projected	2	Cosine	47,9021 %	0 %	5/3	316,193
Original	2	Cosine	78,5548 %	0 %	5/5	15,4214
Projected	4	Cosine	22,9604 %	0 %	5/7	86,2315
Original	4	Cosine	46,8531 %	0 %	5/12	3,79324
Projected	6	Cosine	22,9604 %	0 %	5/5	35,9083
Original	6	Cosine	47,2028 %	0 %	5/24	2,51022
Projected	2	Correlation	52,0979 %	0 %	5/3	273,91
Original	2	Correlation	80,0699 %	0 %	5/6	20,6143
Projected	4	Correlation	51,8648 %	0 %	5/7	46,7877
Original	4	Correlation	47,2028 %	0 %	5/16	5,53442
Projected	6	Correlation	27,4876 %	0,3497 %	5/12	16,9416
Original	6	Correlation	47,3193 %	0 %	5/29	3,69279

The setting of the *k* parameter is one of the key points in applying *k*-means. For this experimental study, different values of *k* parameter were tested; the best of them (in terms of false positive and negative rates) are the ones in Table 4. One of the harmful points in computer security, in general terms, and intrusion detection, in particular, is the false negative rate (FNR). It can be easily seen in Table 4 that only in few of the experiments the FNR is not zero. For those cases, it keeps as a very low value as the number of packets in the network scans is much lower than those from normal traffic. On the other hand, there is not a clear difference (in terms of clustering error) between the experiments on original and projected data, although for a certain number of clusters, the results on projected data are better. Additionally, the number of needed iterations is lower for the projected data, as the dimensionality of the data has been previously reduced through CMLHL. By looking at the sum of distances (sum of point-to-centroid distances, summed over all *k* clusters), a clear conclusion can not be drawn as it depends on the distance method.

Given that *k*-means did not achieved satisfactory results on projected/original data, agglomerative clustering has been also used. Comprehensive details of the run experiments with no clustering error are shown in Table 5. Some of the results, with different values for distance criteria, linkage and number of clusters, are depicted in Fig. 5.

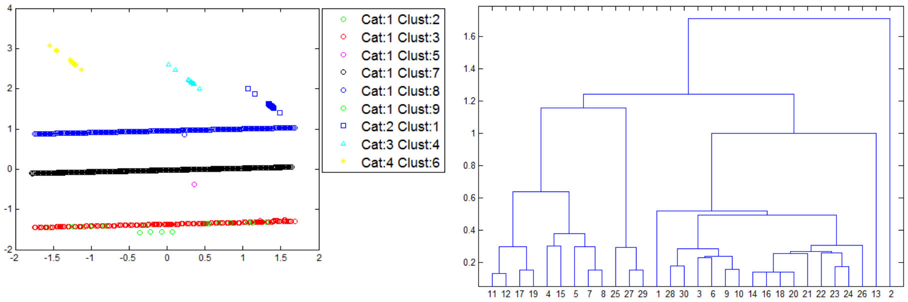
Table 5. Experimental setting of the agglomerative method.

Data	Distance	Linkage	Cutoff	Range	Cluster
Projected	Euclidean	Single	0,37	0,307 - 0,3803	9
Projected	sEuclidean	Single	0,37	0,3087 - 0,3824	9
Projected	Cityblock	Single	0,42	0,4125 - 0,443	9
Projected	Minkowski	Single	0,38	0,307 - 0,3803	9
Projected	Chebychev	Single	0,35	0,2902 - 0,366	9
Projected	Mahalanobis	Single	0,35	0,3084 - 0,3824	9
Original	sEuclidean	Single	1,80	1,533 - 1,813	5
Original	sEuclidean	Complete	4,62	4,62 - 4,628	4
Original	sEuclidean	Average	3,00	2,696 -3,271	4
Original	sEuclidean	Weighted	3,20	3 - 3,261	4
Original	Mahalanobis	Single	2,40	2,289 - 2,438	4
Original	Mahalanobis	Complete	6,00	5,35 - 6,553	3
Original	Mahalanobis	Average	4,00	3,141 - 4,624	3
Original	Mahalanobis	Weighted	4,00	3,504 - 4,536	3

As previously stated, Table 5 contains those results whit no clustering error. It can be seen that in the case of projected data, the minimum number of clusters without error is 9, while in the case of original data, it could be lowered to 3 with appropriate distance method. From the intrusion detection point of view, a higher number of clusters does not mean a higher error rate because more than one cluster can be assigned to both normal and attack traffic.

In the case of original data, the sEuclidean and Mahalanobis distances are minimizing the number of clusters without error. On the contrary, some other distances are applicable in the case of projected data with same performance regarding clustering error.

The results of one of the experiments from Table 5 are depicted on Fig. 5: traffic visualization and the dendrogram associated to agglomerative clustering. It has been selected to show how clustering results improve the visualization capabilities of MOVICAB-IDS. The following sample has been chosen: Euclidean distance, linkage single, cutoff: 0.37, 9 groups without error. It is shown that clusters 1, 4 and 6 are associated to the three network scans and the remaining ones are associated to normal traffic.



5.a Agglomerative clustering on projected data.

5.b Corresponding dendrogram.

Fig. 5. Best results of agglomerative clustering under the frame of MOVICAB-IDS.

4 Conclusions and Future Work

This paper has proposed the use of clustering technics to perform ID on numerical traffic data sets. Experimental results show that some of the applied clustering methods, mainly hierarchical ones, perform a good clustering in the analysed data, according to false positive and negative rates. It can then be concluded that the applied methods are able to properly detect new attacks when projected together with normal traffic. As an unsupervised process is proposed as a whole, the projections ease the task of labelling each one of the clusters as normal or attack traffic.

Future work will be based on the analysis of some other attack situations and the broadening of considered clustering methods. Moreover, new distance metrics would be developed to improve clustering results on projected data. By doing so, the automatic detection facilities of MOVICAB-IDS would be greatly improved.

Acknowledgments

This research has been partially supported through the project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT Project also funded by the MICINN, the Spanish Ministry of Science and Innovation PID 560300-2009-11 and the Junta de Castilla y León CCTT/10/BU/0002.

References

1. Computer Security Threat Monitoring and Surveillance. Technical Report. James P. Anderson Co (1980)

2. Denning, D.E.: An Intrusion-Detection Model. IEEE Transactions on Software Engineering 13 (1987) 222-232
3. Chih-Fong, T., Yu-Feng, H., Chia-Ying, L., Wei-Yang, L.: Intrusion Detection by Machine Learning: A Review. Expert Systems with Applications 36 (2009) 11994-12000
4. Herrero, Á., Corchado, E.: Mining Network Traffic Data for Attacks through MOVICAB-IDS. Foundations of Computational Intelligence, Vol. 4. Springer (2009) 377-394
5. Corchado, E., Herrero, Á.: Neural Visualization of Network Traffic Data for Intrusion Detection. Applied Soft Computing 11 (2011) 2042–2056
6. Abdullah, K., Lee, C., Conti, G., Copeland, J.A.: Visualizing Network Data for Intrusion Detection. Sixth Annual IEEE Information Assurance Workshop - Systems, Man and Cybernetics (2005) 100-108
7. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence 17 (2003) 1447-1466
8. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers 23 (1974) 881-890
9. Corchado, E., Corchado, J.M., Saiz, L., Lara, A.: Constructing a Global and Integral Model of Business Management Using a CBR System. In: Luo, Y. (ed.): CDVE 2004, Vol. 3190. Springer, Heidelberg (2004) 141-147
10. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. 10th European Symposium on Artificial Neural Networks (ESANN 2002) (2002) 143-148
11. Corchado, E., Han, Y., Fyfe, C.: Structuring Global Responses of Local Filters Using Lateral Connections. Journal of Experimental & Theoretical Artificial Intelligence 15 (2003) 473-487
12. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems 10 (1998) 350-356
13. A.K. Jain, M.N.M., P.J. Flynn: Data Clustering: A Review. ACM Computing Surveys 31 (1999)
14. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, Inc., New York, NY. (1973)
15. A.K. Jain, R.C.D.: Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ. (1988)
16. E. Diday, J.C.S.: Clustering Analysis. In Digital Pattern Recognition. K.S. Fu, Ed. Springer-Verlag, Secaucus, NJ. (1976) 47-94
17. R. Michalski, R.E.S., E. Diday: Automated construction of classifications: conceptual clustering versus numerical taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5, 5 (Sept.), (1983) 396-409
18. J. Mao, A.K.J.: A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Trans. Neural Netw. 7 (1996) 16-29
19. McQueen, J.: Some methods for classification and analysis of multivariate observacions. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. (1967) 281-297