Clustering extension of MOVICAB-IDS to identify SNMP community searches

RAÚL SÁNCHEZ* and ÁLVARO HERRERO†, *Department of Civil Engineering, University of Burgos, Spain Avda. Cantabria s/n, 09006 Burgos, Spain*

EMILIO CORCHADO‡, *Departamento de Informática y Automática, Universidad de Salamanca Plaza de la Merced, s/n, 37008 Salamanca, Spain*

Abstract

There are many security systems to protect information resources, but we are still not free from possible successful attacks. This study aims at being one step towards the proposal of an intrusion detection system (IDS) that faces those attacks not previously seen (zero-day attacks), by studying the combination of clustering and neural visualization techniques. To do that, MObile VIsualization Connectionist Agent-Based IDS (MOVICAB-IDS), previously proposed as a hybrid intelligent IDS based on a visualization approach, is upgraded by adding clustering methods. One of the main drawbacks of MOVICAB-IDS was its dependence on human processing; it could not automatically raise an alarm to warn about attacks. Additionally, human users could fail to detect an intrusion even when visualized as an anomalous one. To overcome this limitation, present work proposes the application of clustering techniques to provide automatic response to MOVICAB-IDS to quickly abort intrusive actions while happening. To check the validity of the proposed clustering extension, it faces now an anomalous situation related to the Simple Network Management Protocol: a community search. This attack to get the community string (password guessing) is analysed by clustering and neural tools, individually and in conjunction. Through the experimental stage, it is shown that the combination of clustering and neural projection improves the detection capability on a continuous network flow.

Keywords: Network intrusion detection, exploratory projection pursuit, custering, *k*-means, automatic response, SNMP.

1 Introduction

The ever-changing nature of attack technologies and strategies is one of the most harmful issues of attacks and intrusions, increasing the difficulty of protecting computer systems. It means that new ways of attacking information systems and networks are being developed on a daily basis. For that reason, among others, it is not only prevention tool but also intrusion detection systems (IDS) [1–3] that have become an essential asset in addition to the computer security infrastructure of most organizations. An IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack, in the context of computer networks. Intrusion detection (ID) is, therefore, a field that focuses on the identification of attempted or ongoing attacks.

MObile VIsualisation Connectionist Agent-Based IDS (MOVICAB-IDS) was proposed [4, 5] as a novel IDS comprising a Hybrid Artificial Intelligent System (HAIS). It monitored the network

*E-mail: ahcosio@ubu.es
†E-mail: rsarevalo@ubu.es
‡E-mail: escorchado@usal.es

Vol. 23 No. 1, © The Author 2014. Published by Oxford University Press. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/jigpal/jzu035 Advance Access published 24 December 2014

activity to identify intrusive events. This hybrid intelligent IDS combined different Artificial Intelligence (AI) paradigms to visualize network traffic for ID at packet level. Its main goal was to provide security personnel with an intuitive and informative visualization of network traffic to ease intrusion detection. This IDS was proposed according to the need for visual support for ID that was identified in [6]: '*visualisation tools need to be designed so that anomalies can be easily flagged for later analysis by more experienced analysts*'. The proposed MOVICAB-IDS then applied an unsupervised neural projection model to extract interesting traffic dataset projections and to display them through a mobile visualization interface. One of its main drawbacks was its dependence on human processing; MOVICAB-IDS could not automatically raise an alarm to warn about attacks. Hence, human supervision was required to identify the anomalous situations. Additionally, human users could fail to detect an intrusion even when visualized as an anomalous one due to the limitations of human beings when visually processing big amounts of data [7].

Among all the implemented network protocols, several can be considered highly dangerous in terms of network security. That is the case of Simple Network Management Protocol (SNMP)[8, 9], which was ranked as one of the top five most vulnerable services by CISCO [10]. Specially the two first versions [8, 11] of this protocol that still are the most widely used at present time. SNMP attacks were also listed by the SANS Institute as one of the top 10 most critical Internet security threats [12, 13]. Those were the reasons for MOVICAB-IDS to be focused on the anomalous situations related to this protocol.

SNMP was oriented to manage nodes in the Internet community [8]. It is an application layer protocol that supports the exchange of management information (operating system, version, routing tables and default TTL) between network devices. SNMP offers the capability to poll networked devices and monitor data such as utilization and errors. SNMP is also capable of changing the configurations on the host, allowing the remote management of the network device. This protocol enables network administrators to manage network performance and is used to control network elements such as routers, bridges and switches. As a result, SNMP data are quite sensitive and liable to potential attacks. Indeed, an attack based on this protocol may severely compromise system security [14]. SNMP uses a community string for authentication from the client to the agent on the managed device. The default community string that provides the monitoring or read capability is often 'public'. The default management or write community string is often 'private'. These default community strings allow an attacker to gain information about a device using the read community string 'public', and the attacker can change a systems configuration using the write community string 'private'. The opportunity for this exploit is increased because the SNMP agent is often installed on a system by default without the administrator's knowledge.

Community searches are attacks generated by a hacker sending SNMP queries to the same port number of different hosts, and by using different strategies (brute force, dictionary, etc.) to guess the SNMP community string. Once the community string has been obtained, all the information stored in the Management Information Base (MIB) is available for the intruder. The unencrypted community string can be seen as the SNMP password for versions 1 and 2. In fact, it is the only SNMP authentication mechanism. As reported in [12] This eases SNMP intrusions, as intruders can guess the community string with little effort.

To overcome the limitations of MOVICAB-IDS, present work proposes the application of clustering techniques in conjunction with other mechanisms previously applied in MOVICAB-IDS. 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 experimental study tries to know whether clustering could be more

informative applied over the projected data rather than the original data captured from the network. That is, present work focus on how the detection rate and alarm capabilities of MOVICAB-IDS could be improved by clustering the traffic packets.

Clustering and neural visualization have been previously applied to the identification of different anomalous situations (network scans and MIB transfer) related to the SNMP network protocol [15]. Based on results from this previous study, present work advances that by focussing on a new attack situation related to SNMP: community string searches. Several well-known clustering techniques are analysed, compared and combined with powerful neural visualization models to identify this new kind of attacks.

1.1 Related previous work

Present work aims to provide automatic response to MOVICAB-IDS applying clustering methods, that have been previously applied to ID: [16] proposes an alert aggregation method, clustering similar alerts into a hyper alert based on category and feature similarity. From a similar perspective, [17] proposes a two-stage clustering algorithm to analyse the spatial and temporal relation of the network intrusion behaviours' alert sequence. [18] describes a classification of network traces through an improved nearest neighbour method, while [19] applies data mining algorithms for the same purpose and the results of preformatted data are visually displayed. Finally [20] discusses on how the clustering algorithm is applied to intrusion detection and analyses intrusion detection algorithm based on clustering problems.

The above-mentioned studies apply clustering methods directly to raw network data, whereas the proposed system applies clustering to previously projected data (processed by neural models). Present work focuses on the upgrading of MOVICAB-IDS, to incorporate new facilities. It is now required an enhanced visualization by combining projection and clustering results to ease traffic analysis by security personnel. That is, both simple and accumulated segments are now processed by neural projection and clustering techniques. By doing so, further information on the nature of the packets travelling along the network could be compressed in the visualization. Additionally, automatic response could be incorporated in MOVICAB-IDS to quickly abort intrusive actions while happening.

The remaining sections of this study are structured as follows: Section 2 discusses the combination of visualization and clustering techniques and describes the applied ones. Experimental setting and results are presented in Section 3 while the conclusions of this study are discussed in Section 4.

2 Combining visualization and clustering

MOVICAB-IDS was based on the application of different AI paradigms to process the continuous data flow of network traffic. To do so, MOVICAB-IDS splits massive traffic data into limited datasets and visualizes them, thereby providing security personnel with an intuitive snapshot to monitor the events taking place in the observed computer network. The following paradigms were combined within MOVICAB-IDS:

- **Multiagent system (MAS)**: some of the components are wrapped as deliberative agents capable of learning and evolving with the environment [21].
- **Case-based reasoning (CBR)**: some of the agents contained in the MAS are known as CBR-BDI agents [22] because they integrate the Beliefs, Desires and Intentions (BDI) [23] model and the CBR paradigm.

- **Artificial neural networks (ANNs)**: the connectionist approach fits the ID problem mainly because it allows a system to learn, in an empirical way, the input-output relationship between traffic data and its subsequent interpretation [24]. The previously described CBR-BDI agents incorporate the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) neural model (described in Section 2.1) to generate projections of network traffic.

The combination of these paradigms allowed the user to benefit from certain properties of ANN (generalization that allows the identification of previously unseen attacks), CBR (learning from past experiences) and agents (reactivity, proactivity, sociability and intelligence), which greatly facilitates the ID task.

The general framework for MOVICAB-IDS, to perform the tasks depicted in Figure 1, could be described as follows:

- packets travelling through the network are intercepted by a **capture device**;
- traffic is **coded by a set of features** spanning a multidimensional vector space; and
- 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.

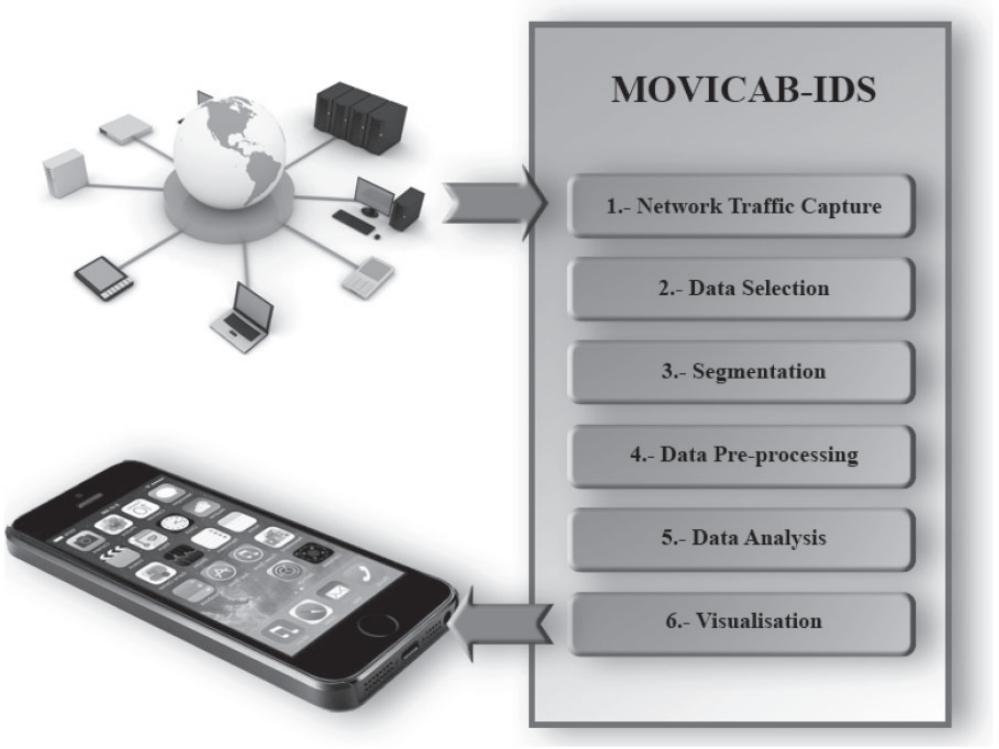

FIG. 1. MOVICAB-IDS task organization.

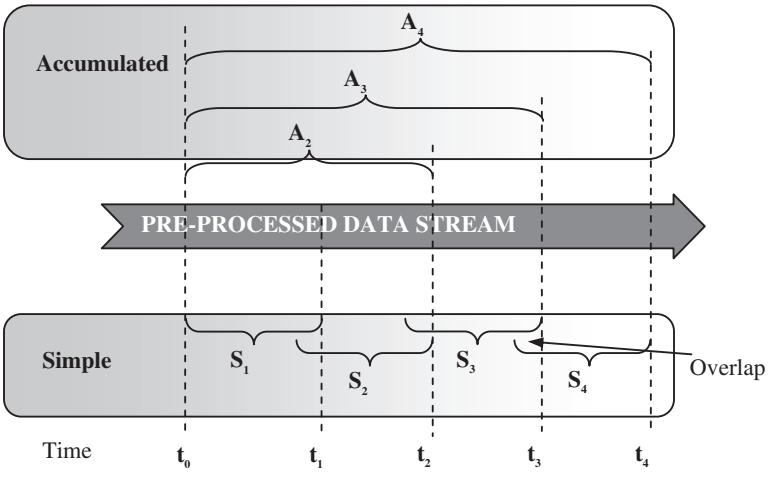

FIG. 2. MOVICAB-IDS segmentation of pre-processed data.

To process the continuous flow of network traffic, MOVICAB-IDS split the pre-processed data stream into simple and accumulated segments as depicted in Figure 2. These segments are defined as follows:

- **Equal simple segments (S_x)**: each simple segment contains all the packets with timestamps between the initial and final time limits of the segment. There must be a time overlap between each pair of consecutive simple segments because anomalous situations could conceivably take place between simple segment S_x and S_{x+1} (where S_{x+1} is the next segment following S_x).
- **Accumulated segments (A_x)**: each one of these segments contains several consecutive simple ones. To avoid duplicated packets, time overlap is removed in accumulated segments.

One of the main reasons for such a partitioning was to present a long-term picture of the evolution of network traffic to the network administrator, as it allows the visualization of attacks lasting longer than the length of a simple segment. To avoid confusion on the part of the analyst, accumulated segments were visualized at the same time. This prompted the network administrator to realize that there is only one anomalous situation being visualized twice.

Present work focuses on the upgrading of the previously introduced framework, to incorporate new facilities, as described below. The initial architecture of MOVICAB-IDS is extended to combine clustering methods and projection models, as depicted in Figure 3. The following subsections describe the different techniques that take part in the proposed solution. For the dimensionality reduction as a projection method, CMLHL [25] 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, described in Section 2.2.

2.1 CMLHL

The standard statistical method of Exploratory Projection Pursuit (EPP) [26] provides a linear projection of a dataset, but it projects the data onto a set of basis vectors which best reveal the interesting

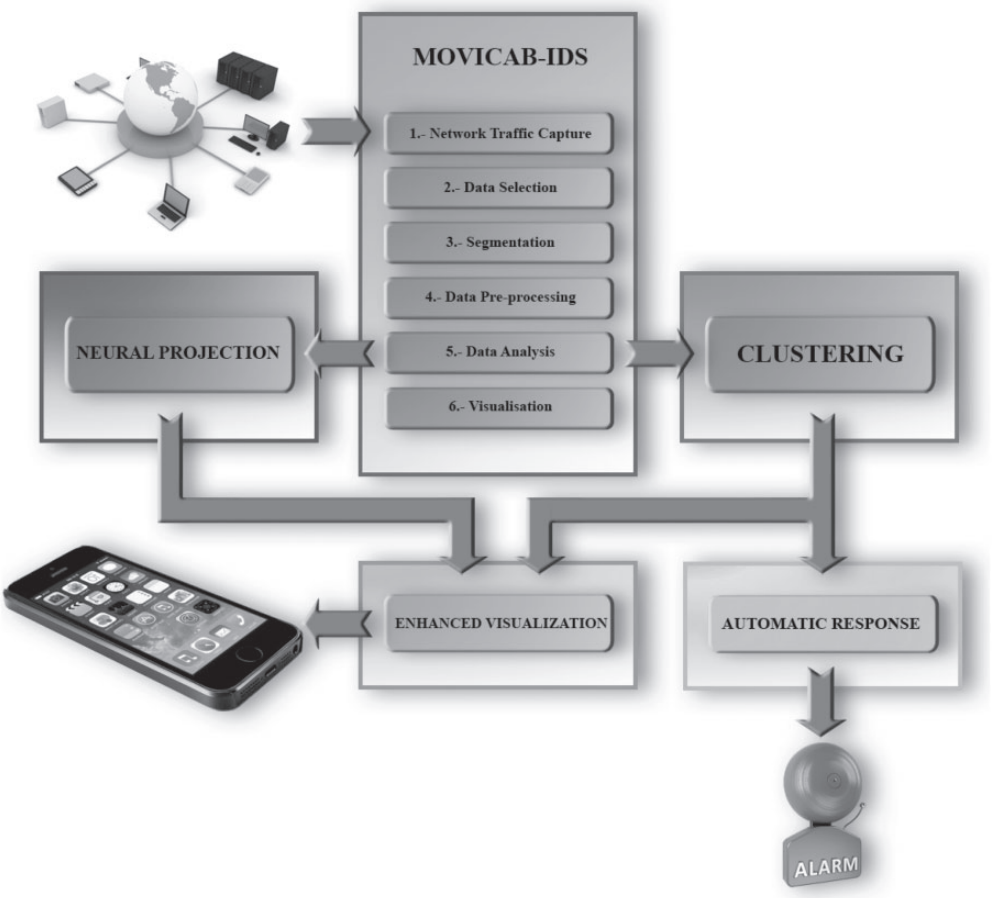

FIG. 3. Clustering extension of MOVICAB-IDS.

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) [27, 28]. It identifies interestingness by maximizing the probability of the residuals under specific probability density functions which are non-Gaussian.

One extended version of this model is the CMLHL [25] model. CMLHL is based on MLHL [27, 28] adding lateral connections [25, 29] which have been derived from the Rectified Gaussian Distribution [30]. The resultant net can find the independent factors of a dataset but does so in a way that captures some type of global ordering in the dataset.

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 [25, 29] 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:

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

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 [25, 27, 28] and A a symmetric matrix used to modify the response to the data [25]. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.2 Clustering

Cluster analysis [31, 32], 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.

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 [33, 34]. 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 [31], but given the high number and the strong diversity of the existent clustering methods, we have focused on the ones shown in Figure 4 based on the suggestions in [31].

At the top level, there is a distinction between hierarchical and partitional approaches. Hierarchical methods produce a nested series of partitions (illustrated on a dendrogram which is a tree diagram)

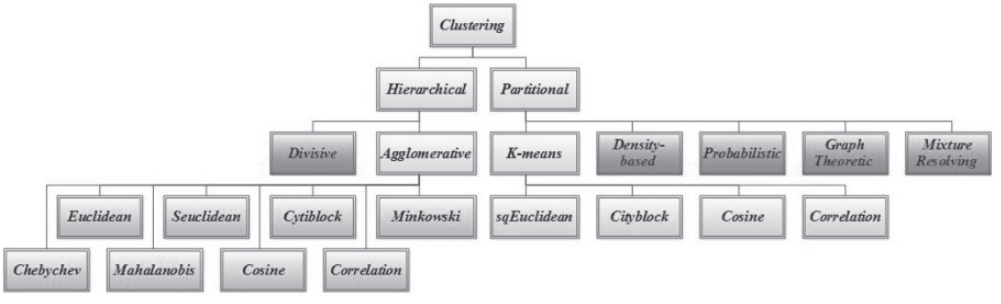

FIG. 4. Clustering methods used in this paper: one hierarchical (agglomerative) and other partitional method (*K*-means).

based on a similarity for merging or splitting clusters, while partitional methods identify the partition that optimizes (usually locally) a clustering criterion. Hence, obtaining a hierarchy of clusters can provide more flexibility than other methods. A partition of the data can be obtained from a hierarchy by cutting the tree of clusters at certain level.

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 article.

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. Many of these methods are based on the iterative optimization of a criterion function that reflects the similarity between a new data and the each of the initial patterns selected for a specific iteration. Partitional methods have advantages in applications involving large datasets for which the construction of a dendrogram is computationally prohibitive. The problem of these algorithms is the need of the number of desired output clusters [31]. Exhaustive search over all the set of possible initial labeling for an optimum output is clearly computationally prohibitive. Therefore, in practice, the algorithm is typically run a number of times with different starting states, and the best configuration obtained from all of the runs is used as the output clustering. Hence, we can meet different results depending on the initial labelling chosen (usually random). Additional techniques for the grouping operation include density-based [35], probabilistic [36], graph-theoretic [37] and mixture-resolving [31] clustering methods, but they are not used in this article.

3 Experiments and results

The main idea behind this experimental study is 2-fold: on the one hand, it is aimed at discovering whish simple clustering techniques can be successfully applied to the SNMP intrusion detection problem. On the other hand, it tries to check whether clustering and projection could work in unison to ease intrusion detection. Additionally, as previously stated, the experimental study tries to show whether clustering could be more informative applied over the projected data rather than the original data captured from the network. That is, the same data have been analysed for intrusion detection, in two different ways, according to what is depicted in Figure 3. The two considered alternatives are: clustering on projected (dimensionality-reduced) data, and clustering on original (five-dimensional) data.

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

3.1 Datasets

Five features were extracted from the headers of packets travelling along the network to form the dataset, and one more (the first one) was added to identify each single packet:

- **Packet ID**: sequential integer nonlinear.

- **Timestamp**: the time difference in relation to the first captured packet. Sequential integer nonlinear.
- **Source Port**: the port of the source host from where the packet is sent. Discrete integer values.
- **Destination Port**: the port of the destination host to where the packet is sent. Discrete integer values.
- **Size**: total packet size (in bytes).
- **Protocol ID**: we have used values between 1 and 35 to identify the packet protocol. Discrete integer values.

After initial experiments, it was decided to apply MOVICAB-IDS to the anomalous situations related to SNMP. The Management Information Base (MIB) can be defined in broad terms as the database used by SNMP to store information about the elements that it controls. Like a dictionary, an MIB defines a textual name for a managed object and explains its meaning.

As previously stated, consideration should be given to SNMP from a security standpoint, due to its very limited security mechanisms and the security sensitive data that is stored in the MIB. Attackers can exploit these vulnerabilities in the SNMP for network reconnaissance and remote reconfiguration or shut down of SNMP devices. Thus, MOVICAB-IDS focuses on the most commonly reported types of attacks that target SNMP:

- **SNMP network scan**: three types of scans (or sweeps) have been defined: network scans, port scans and their hybrid block scans [38]. Unlike other attacks, scans must use a real-source IP address, because the results of the scan (open ports or responding IP addresses) must be returned to the attacker [39]. A port scan (or sweep) may be defined as series of messages sent to different port numbers to gain information on its activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services that this host is providing. So, a scan is an attempt to count the services running on a machine (or a set of machines) by probing each port for a response, providing information on where to probe for weaknesses. Thus, scanning generally precedes any further intrusive activity. This work focuses on the identification of network scans, in which the same port (the SNMP port) is the target for a number of computers in an IP address range. A network scan is one of the most common techniques used to identify services that might then be accessed without permission [40].
- **MIB information transfer**: this situation involves a transfer of some (or all the) information contained in the SNMP MIB, generally through the *Get* command or similar primitives such as *GetBulk* [41, 42]. This kind of transfer is potentially quite a dangerous situation because anybody possessing some free tools, some basic SNMP knowledge and the community string (in SNMP versions 1 and 2), will be able to access all sorts of interesting and sometimes useful information. As specified by the Internet Activities Board, the SNMP is used to access MIB objects. Thus, protecting a network from malicious MIB information transfer is crucial. However, the'normal' behaviour of a network may include queries to the MIB. This is a situation in which visualization-based IDSs are quite useful; these situations may be visualized as anomalous by an IDS but it is the responsibility of the network administrator to decide whether or not it constitutes an intrusion.
- **Community search**: this attack is generated by a hacker sending SNMP queries to the same port number of different hosts, and by using different strategies (brute force, dictionary, etc.) to guess the SNMP community string. Once the community string has been obtained, all the information stored in the MIB is available for the intruder. The unencrypted community string can be seen as the SNMP password for versions 1 and 2. In fact, it is the only SNMP authentication mechanism.

As reported in [12], this eases SNMP intrusions, as intruders can guess the community string with little effort.

In addition to the previously mentioned SNMP situations, the analysed datasets contain a great background of network traffic that may be considered as 'normal'. Information about the packets was gathered from a middle-size university network. As used in previous experiments, further details on the data can be found in [4, 5]. The first two attack situations previously introduced have been already studied in previous work by the authors under the frame of the clustering extension of MOVICAB-IDS [43]. The datasets studied in present work contain an additional, and hence new for this clustering-extension, anomalous situation: community searches. Two different segments are analysed to check the validity of the proposed extension when identifying community searches: dataset 1 contains simple segment of normal traffic and dataset 2 contains accumulated segments.

The simple segment contains all the packets whose timestamp is between the segment initial and final time limit. There is a slight time overlap between each pair of consecutive simple segments. The main reason for overlapping simple segments is that anomalous situations could conceivably take place between simple segment. In this case, it would be necessary to consider some packets twice in order to visualize the end of the anomalous situation and the evolution between simple segments. To prevent confusion of the analyst (e.g., the same anomalous situation is visualized in two different simple segments), accumulated segments are visualized at the same time. Each one of these segments contains several consecutive simple ones. This will lead the network administrator to realize that there is only one anomalous situation being visualized twice. They can be described as follows:

- Dataset 1: contains a simple segment of 'normal' traffic, two network scans that target port numbers 1434 and 65,788, and three SNMP community searches that target port numbers 161, 1161 and 2161 of all the machines within an IP address range. Three different community names were used for each one of these port numbers. In total, 3235 packets are contained in this dataset.
- Dataset 2: contains an accumulated segment of 'normal' traffic, two network scans that target port numbers 1434 and 65,788, and three community searches that target port numbers 161, 1161, 2161. Three different community names were used for each one of these port numbers. In total, 8219 packets are contained in this dataset.

3.2 Details of applied clustering techniques

As similarity is fundamental to the definition of a cluster, a measure of the similarity is essential to most clustering methods and it must be carefully chosen. Present study applies well-known distance criteria used for examples whose features are all continuous.

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 dataset has compact or isolated clusters [44]. 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

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

Metric	Description
Euclidean	Euclidean distance.
sEuclidean	Standardized Euclidean distance. Each coordinate difference between rows is scaled by dividing by the corresponding element of the standard deviation.
Cityblock	City block metric also known as Manhattan distance.
Minkowski	Minkowski distance.
Chebychev	Chebychev distance (maximum coordinate difference).
Mahalanobis	Mahalanobis distance, using the sample covariance.
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).

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.

variances and pairwise linear correlations. The regularized Mahalanobis distance was used in [44] to extract hyperellipsoidal clusters.

3.2.1 K-Means algorithm

Four different distance measures are applied in present study for k-means algorithm, as described in Table 2. The proposed solution has been tested on all of them and the best result can be seen on Section 3.3.

3.2.2 Agglomerative clustering

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).

Most hierarchical clustering algorithms are based on the single link and complete link. These two algorithms differ in the way they characterize the similarity between a pair

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 centre 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).

of clusters:

1. *Single-link algorithm:* the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from each of the clusters.
2. *Complete-link algorithm:* the distance between two clusters is the maximum of all pairwise distances between patterns in each of the clusters.

In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-link algorithm produces compact clusters [45]. The single-link algorithm, by contrast, suffers from a chaining effect. It has a tendency to produce clusters that are straggly or elongated. The clusters obtained by the complete-link algorithm are more compact than those obtained by the single-link algorithm. The single-link algorithm is more versatile than the complete-link algorithm. However, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications than the single-link algorithm [31].

3.3 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 (colours and symbols). The projections comprise a legend that states the colour and symbol used to depict each packet, according to the original category of the data.

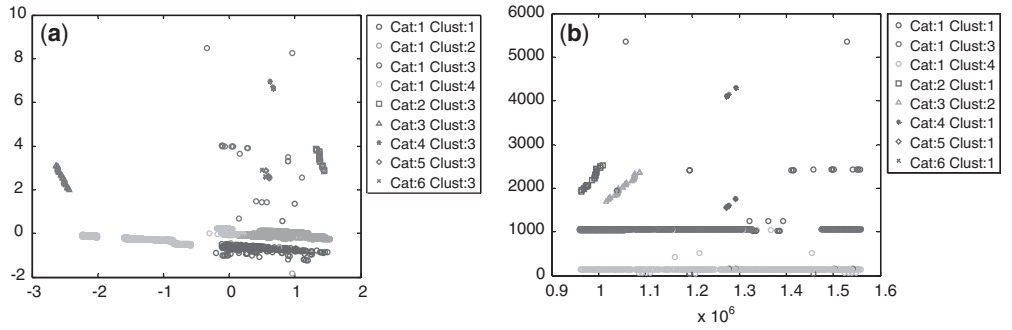

FIG. 5. Best clustering result under the frame of MOVICAB-IDS through k-means for Dataset 1. (a) K-means on projected data: $k=4$, correlation distance. (b) K-means on original data: $k=4$, correlation distance.

The clustering methods have been applied several times to the analysed datasets by combining different values for the algorithm options. The following sections show the best results, whose parameter settings and performance are also detailed. The following subsections comprise the results obtained by the projection and clustering technique for each one of the datasets.

3.3.1 Dataset 1

Figure 5 shows the results obtained by k-means on this data. The data has been labelled as follows: normal traffic (Cat. 1), network scan that target port number 1434 (Cat. 2), network scan that target port number 65788 (Cat. 3), community search that target port number 161 (Cat. 4), community search that target port number 1161 (Cat. 5), community search that target port number 2161 (Cat. 6).

For the normal traffic on projected data (Cat. 1 in Fig. 5a), the clustering does not group data without mistakes as it mixes all the anomalous situations, grouped in the same group (Clust. 3 in Fig. 5a), with some of the normal traffic (Cat. 1 Clust. 3). For the original data, although the number of clusters (k parameter) is the same, data are more mixed, hence the clustering is worst. In this case, most of the anomalous situations are grouped in the same group (Clust. 1), together with some of the normal traffic (Cat. 1 Clust. 1), except the packets from the second network scan that target port number 65,788 (Cat. 3 Clust. 2), that are grouped alone in cluster 2. Apart from these two projections, some more experiments have been conducted, whose details: performance, false positive rates (FPRs) and false negative rates (FNRs), values of k parameter, etc., can be seen in Table 4.

The run experiments for the agglomerative method with very little FPR or no clustering error are shown in Table 5.

It can be seen that, in the case of projected data, the minimum number of clusters with a very little error is 4, while in the case of original data is 7 with appropriate distance method. In the case of original data, the sEuclidean distance minimizes the number of clusters with very low error rate, and in the case of projected data there are some other methods applicable providing a non-zero, although very low, value of FPR.

The results of one of the best experiments from Table 5 are depicted in Figure 6, including traffic visualization and the associated dendrogram on projected data. The chosen experiment details are: correlation distance, weighted linkage, cutoff: 0.5 and 4 groups with FPR = 0.2479%.

TABLE 4. *k*-Means experiments with different conditions for Dataset

Data	*k*	Distance criteria	FPR (%)	FNR (%)	Replicates/Iterations Iterations	Sum of Distances
Projected	2	sqEuclidean	81.0201	0	5/5	6749.85
Original	2	sqEuclidean	37.7125	2.6892	5/7	2.60673E+13
Projected	4	sqEuclidean	00.5255	0	5/5	1936.62
Original	4	sqEuclidean	51.4684	0	5/17	6.37068E+12
Projected	6	sqEuclidean	0.5255	0	5/10	1157.53
Original	6	sqEuclidean	36.7542	0	5/34	3.18335E+12
Projected	2	Cityblock	67.2334	1.8547	5/3	4161.98
Original	2	Cityblock	32.2720	2.6892	5/5	2.82857E+08
Projected	4	Cityblock	0.6491	0	5/5	2341.34
Original	4	Cityblock	45.4714	0	5/6	1.48534E+08
Projected	6	Cityblock	0.6491	0	5/5	1883.25
Original	6	Cityblock	36.4452	0	5/29	1.05724E+08
Projected	2	Cosine	53.7249	1.8547	5/4	760.895
Original	2	Cosine	67.2952	1.8547	5/2	0.0373789
Projected	4	Cosine	0.7110	0	5/13	233.695
Original	4	Cosine	67.2952	0	5/12	0.00313724
Projected	6	Cosine	0.6801	0	5/13	74.5217
Original	6	Cosine	14.8995	0	5/10	0.00811161
Projected	2	Correlation	37.8362	0	5/3	727.484
Original	2	Correlation	67.2952	1.8547	5/2	0.0357083
Projected	4	Correlation	0.6491	0	5/10	169.474
Original	4	Correlation	14.8995	0	5/5	0.00769754
Projected	6	Correlation	0.6491	0	5/17	49.5901
Original	6	Correlation	14.8995	0	5/11	0.0025605

3.3.2 Dataset 2

Figure 7 shows the results obtained by *k*-means on this data. The data has been labelled as follows: normal traffic (Cat. 1), network scan that target port number 1434 (Cat. 2), network scan that target port number 65,788 (Cat. 3), community search that target port number 161 (Cat. 4), community search that target port number 1161 (Cat. 5), community search that target port number 2161 (Cat. 6). Although this dataset contains an accumulated segment with a great amount of packets, the number of categories for attacks is not proportionally increased.

For the normal traffic on projected data (Cat. 1 in Fig. 7a), the clustering does not group data without mistakes as it mixes most of the anomalous situations, grouped in the same group (Clust. 5 in Fig. 7a), with some of the normal traffic (Cat. 1 Clust. 5). For the original data, although the number of clusters (*k* parameter) is the same, data are more mixed, hence the clustering is worst. In this case, the two network scans are grouped in the same group (Clust. 6), together with some of the normal traffic (Cat. 1 Clust. 6), and the three community searches are clustered together (Clust. 2) with some of normal traffic's packets (Cat. 1 Clust. 2). Apart from these two projections, some more experiments have been conducted, whose details (performance, FPR and FNR, values of *k* parameter, etc.) can be seen in Table 6.

TABLE 5. Experimental setting of the agglomerative method for Dataset 1

Data	Distance	Linkage	Cutoff	FPR (%)	Cluster
Projected	Euclidean	Single	0.85	0.4637	11
Projected	sEuclidean	Average	2.5	0.5255	7
Projected	Cityblock	Single	1.25	0.4637	9
Projected	Minkowski $p=3$	Single	1	0.4637	8
Projected	Cosine	Average	0.6	0.7110	5
Projected	Cosine	Weighted	0.7	0.7110	5
Projected	Correlation	Complete	0.8	0.6801	5
Projected	Correlation	Average	0.2	0.6491	7
Projected	Correlation	Weighted	0.5	0.6801	4
Original	sEuclidean	Average	3	0.6491	7
Original	sEuclidean	Weighted	3	0.6800	7
Original	Cityblock	Complete	150,000	22.4111	8
Original	Mahalanobis	Complete	5	0.5255	7
Original	Mahalanobis	Average	3	0.6491	8

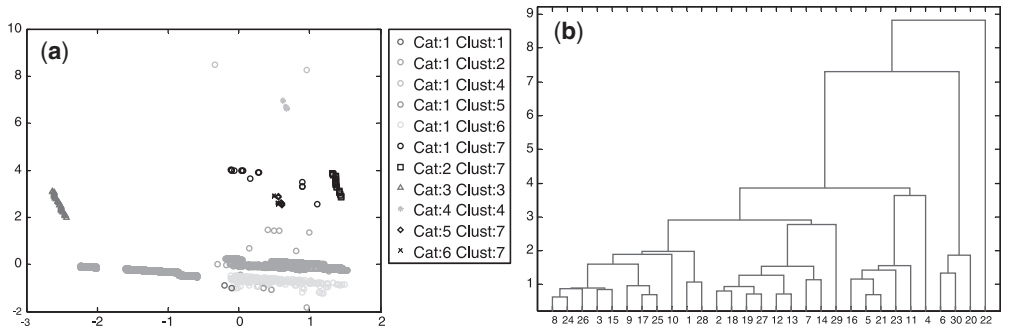

FIG. 6. Best results of agglomerative clustering under the frame of MOVICAB-IDS for Dataset 1. (a) Agglomerative clustering on projected data: sEuclidean, linkage: average, cutoff: 2.5. (b) Corresponding dendrogram.

The run experiments for the agglomerative method with very little FPR error are shown in Table 7.

It can be seen that in both cases (projected and original data), the minimum number of clusters with very little error is 6, although, in the case of projected data, it is obtained with a wide range of methods, while there is only one method (sEuclidean linkage complete) that can obtain that number of clusters. For original data, the sEuclidean distance minimizes the number of clusters with very low error rate, and in the case of projected data there are some other methods applicable providing a non-zero, although very low, value of FPR.

The results of one of the best experiments from Table 7 are depicted in Figure 8 including traffic visualization and the associated dendrogram on projected data. The chosen experiment details are: Mahalanobis distance, weighted linkage, cutoff: 3 and 6 groups with FPR = 0.5719%.

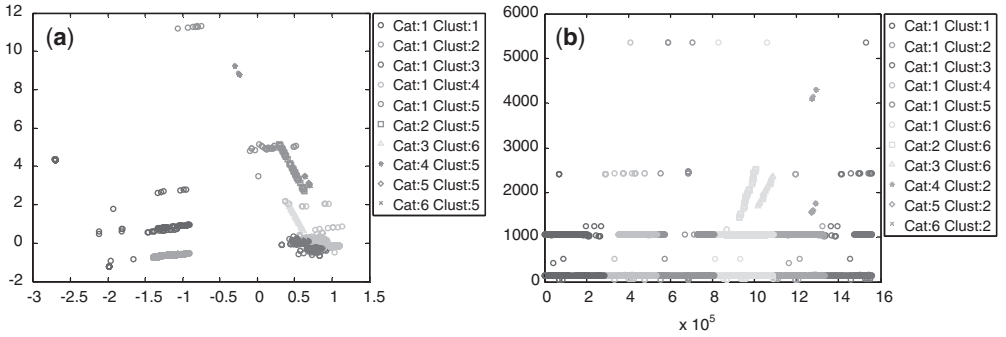

FIG. 7. Best clustering result under the frame of MOVICAB-IDS through k-means for Dataset 2. (a) k-means on projected data: $k=6$, sqEuclidean distance. (b) k-means on original data: $k=6$, sqEuclidean distance.

TABLE 6. k-means experiments with different conditions for Dataset 2

Data	k	Distance Criteria	FPR (%)	FNR (%)	Replicates/ Iterations	Sum of distances
Projected	2	sqEuclidean	80.5873	0	5/5	6749.85
Original	2	sqEuclidean	56.3524	2.8748	5/8	2.60673E+13
Projected	4	sqEuclidean	53.4158	0	5/7	1936.62
Original	4	sqEuclidean	72.8594	0.0309	5/12	6.37068E+12
Projected	6	sqEuclidean	53.4158	0	5/9	1219.47
Original	6	sqEuclidean	78.3617	0.0309	5/25	3.18335E+12
Projected	2	Cityblock	71.0388	1.8547	5/4	4161.98
Original	2	Cityblock	61.7929	2.8748	5/6	2.82857E+08
Projected	4	Cityblock	72.0866	0	5/11	2610.15
Original	4	Cityblock	64.8841	0.1855	5/7	1.48534E+08
Projected	6	Cityblock	0.2164	0	5/18	1882.77
Original	6	Cityblock	77.9907	0.0309	5/26	1.05723E+08
Projected	2	Cosine	53.2921	1.8547	5/2	760.895
Original	2	Cosine	66.8624	1.8547	5/3	0.0373789
Projected	4	Cosine	0.2782	0	5/8	195.351
Original	4	Cosine	66.8624	0	5/7	0.00811376
Projected	6	Cosine	0.2473	0	5/14	74.5217
Original	6	Cosine	66.8623	0	5/9	0.00798351
Projected	2	Correlation	37.4343	0.0309	5/2	727.484
Original	2	Correlation	66.8624	1.8547	5/2	0.0357083
Projected	4	Correlation	13.8176	0	5/5	169.474
Original	4	Correlation	66.8624	0	5/2	0.00769754
Projected	6	Correlation	13.7249	0	5/12	49.5901
Original	6	Correlation	66.8624	0	5/8	0.00100529

TABLE 7. Experimental setting of the agglomerative method for Dataset 2

Data	Distance	Linkage	Cutoff	FPR (%)	Cluster
Projected	Euclidean	Complete	5	0.7179	6
Projected	Euclidean	Average	3	0.7179	6
Projected	Euclidean	Weighted	3	0.5719	6
Projected	sEuclidean	Complete	5	0.7179	6
Projected	sEuclidean	Average	5	0.7179	6
Projected	sEuclidean	Weighted	3	0.5719	6
Projected	Cityblock	Complete	5	0.5719	8
Projected	Cityblock	Average	5	0.7179	6
Projected	Cityblock	Weighted	5	0.5719	6
Projected	Minkowski $p=3$	Average	4	0.7179	6
Projected	Chebychev	Weighted	3	0.7179	6
Projected	Mahalanobis	Weighted	3	0.5719	6
Projected	Cosine	Average	0.25	0.7178	6
Projected	Cosine	Weighted	0.45	0.9004	6
Original	sEuclidean	Complete	5	0.6692	6
Original	sEuclidean	Average	3	0.6692	9
Original	sEuclidean	Weighted	3	0.5719	9
Original	Mahalanobis	Complete	5	0.5719	8
Original	Mahalanobis	Average	4	0.7179	7
Original	Mahalanobis	Weighted	4	0.6692	9

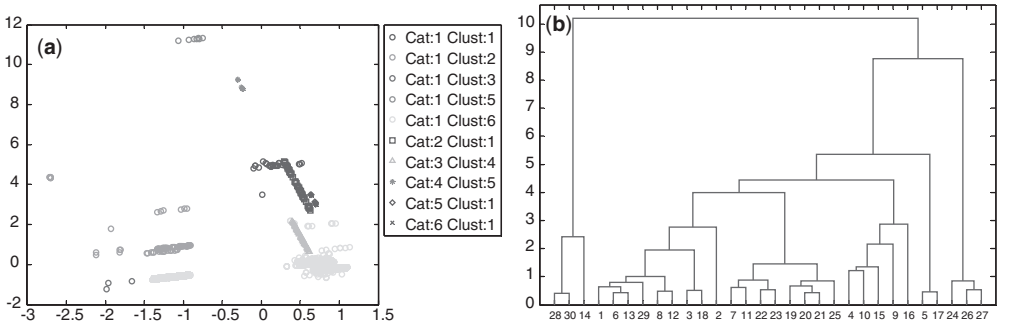

FIG. 8. Best results of agglomerative clustering under the frame of MOVICAB-IDS for Dataset 2. (a) Agglomerative clustering on projected data: Mahalanobis, linkage: weighted, cutoff: 3. (b) Corresponding dendrogram.

4 Conclusions

A clustering extension of MOVICAB-IDS has been proposed in present work as simple clustering methods and has been applied to early stages of an SNMP attack: network scans and SNMP community searches. It is worth highlighting that community searches are analysed in combination with these other anomalous situations, what makes its identification even more difficult.

Detailed conclusions about experiments on the different datasets and with several different clustering techniques and criteria, can be found in Section 3. Experimental results show that some of the applied clustering methods obtain a good clustering performance on the analysed data, according to FPR and FNR. The obtained results vary from the different analysed datasets and the behaviour of the applied clustering techniques. These results are consistent with those previously obtained for other SNMP anomalous situations [15].

Regarding the distance criteria, none of them is clearly the best one, so its selection will depend on the analysed data. Finally, by considering projected versus original data, it can be said that the results over projected data in the case of community searches are better (fewer number of groups with the same FNR and FPR) and other advantage is the smaller execution time, what is not covered in present work.

Finally, it can be concluded that the applied methods are able to properly detect anomalous situations when projected together with normal traffic. It has been proved that clustering methods could help in ID not only by applying them to the same data that is projected but in a subsequent way. On the other hand, by clustering packets, automatic response could be added to MOVICAB-IDS, to quickly abort intrusive actions while happening.

References

[1] Computer Security Threat Monitoring and Surveillance. *Technical Report*. James P. Anderson Co, 1980.

[2] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering* **13**, 222–232, 1987.

[3] T. Chih-Fong, H. Yu-Feng, L. Chia-Ying and L. Wei-Yang. Intrusion detection by machine learning: a review. *Expert Systems with Applications* **36**, 11994–12000, 2009.

[4] Á. Herrero and E. Corchado: mining network traffic data for attacks through MOVICAB-IDS. In *Foundations of Computational Intelligence*, vol. 4. pp. 377–394. Springer 2009.

[5] E. Corchado and Á. Herrero. Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing* **11**, 2042–2056, 2011.

[6] J. R. Goodall, W. G. Lutters and A. Komlodi. The work of intrusion detection: rethinking the role of security analysts. In *Americas Conference on Information Systems*, pp. 1421–1427, 2004.

[7] Á. Herrero and E. Corchado. *Mobile Hybrid Intrusion Detection: the MOVICAB-IDS System*, Vol. 334. Springer, 2011.

[8] J. Case, M. S. Fedor, M. L. Schoffstall and C. Davin. *Simple Network Management Protocol (SNMP)*. IETF RFC 1157, 1990.

[9] J. Davin, J. Galvin and K. McCloghrie. *SNMP Administrative Model*. IETF RFC 1351, 1992.

[10] *Vulnerability Statistics Report*. Cisco Secure Consulting, 2000.

[11] J. Case, K. McCloghrie, M. Rose and S. Waldbusse. *Introduction to Version 2 of the Internet-standard Network Management Framework*. IETF RFC 1441, 1993.

[12] *The Top 10 Most Critical Internet Security Threats (2000-2001 Archive)*. SANS Institute, 2001.

[13] S. Northcutt, M. Cooper, K. Fredericks, M. Fearnow and J. Riley. *Intrusion Signatures and Analysis. New Riders Publishing*, 2001.

[14] J. M. Myerson. Identifying enterprise network vulnerabilities. *International Journal of Network Management* **12**, 135–144, 2002.

[15] R. Sánchez, Á. Herrero and E. Corchado. Visualization and clustering for SNMP intrusion detection. *Cybernetics and Systems: An International Journal* **44**, 505–532, 2013.
[16] Q. H. Zheng, Y. G. Xuan and W. H. Hu. An IDS alert aggregation method based on clustering. In: H. Zhang, G. Shen and D. Jin (eds.) *Advanced Research on Information Science, Automation and Material System*, pts 1-6, vol. 219-220. pp. 156-159. Trans Tech Publications Ltd, 2011.
[17] L. B. Qiao, B. F. Zhang, Z. Q. Lai, J. S. Su. IEEE: Mining of Attack Models in IDS Alerts from Network Backbone by a Two-stage Clustering Method. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & Phd Forum*. pp. 1263–1269. IEEE, New York, 2012.
[18] Jiang, S., Song, X., Wang, H., Han, J.-J., Li, Q.-H.: A clustering-based method for unsupervised intrusion detections. *Pattern Recognition Letters* **27**, 802–810, 2006.
[19] K. Y. Cui. IEEE: Research On Clustering Technique In Network Intrusion Detection. Ieee Computer Soc, 2012.
[20] L. Ge, C. Q. Zhang. The application of clustering algorithm in intrusion detection system. In D. Jin and S. Lin (eds.) *Advances in Future Computer and Control Systems*, vol. 159. pp. 77–82. Springer, 2012.
[21] Á. Herrero, E. Corchado, M. A. Pellicer and A. Abraham. MOVIH-IDS: a mobile-visualization hybrid intrusion detection system. *Neurocomputing* **72**, 2775–2784, 2009.
[22] C. Carrascosa, J. Bajo, V. Julián, J. M. Corchado and V. Botti. Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Systems with Applications: An International Journal* **34**, 2–17, 2008.
[23] M. E. Bratman. *Intentions, Plans and Practical Reason*. Harvard University Press, 1987.
[24] Á. Herrero, E. Corchado, P. Gastaldo, R. Zunino. Neural projection techniques for the visual inspection of network traffic. *Neurocomputing* **72**, 3649–3658, 2009.
[25] E. Corchado and C. Fyfe. Connectionist techniques for the identification and suppression of interfering underlying factors. *International Journal of Pattern Recognition and Artificial Intelligence* **17**, 1447–1466, 2003.
[26] J. H. Friedman, J. W. Tukey. A projection pursuit algorithm for exploratory data-analysis. *IEEE Transactions on Computers* **23**, 881–890, 1974.
[27] E. Corchado, J. M. Corchado, L. Saiz and A. Lara. Constructing a global and integral model of business management using a CBR system. In Y. Luo (ed.) *CDVE 2004*, vol. 3190. pp. 141–147. Springer, 2004.
[28] C. Fyfe and E. Corchado. Maximum Likelihood Hebbian Rules. *10th European Symposium on Artificial Neural Networks (ESANN 2002)* pp. 143–148, 2002.
[29] E. Corchado, Y. Han and C. Fyfe. Structuring global responses of local filters using lateral connections. *Journal of Experimental & Theoretical Artificial Intelligence* **15**, 473–487, 2003.
[30] H. S. Seung, N. D. Socci and D. Lee. The rectified gaussian distribution. *Advances in Neural Information Processing Systems* **10**, 350–356, 1998.
[31] A. K. Jain, M. N. M., P. J. Flynn. Data clustering: a review. *ACM Computing Surveys* **31**, 1999.
[32] R. Xu and D. C. Wunsch. *Clustering*. Wiley, 2009.
[33] B. Andreopoulos, A. An, X. Wang and M. Schroeder. A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics* **10**, 297–314, 2009.
[34] W. W. Zhuang, Y. F. Ye, Y. Chen and T. Li. Ensemble clustering for internet security applications. *IEEE Transactions on Systems Man Cybernetics Part C-Applications and Review*. **42**, 1784–1796, 2012.

[35] Q. Tu, J. F. Lu, B. Yuan, J. B. Tang and J. Y. Yang. Density-based hierarchical clustering for streaming data. *Pattern Recognition Letters* **33**, 641–645, 2012.
[36] V. L. Brailovsky. A probabilistic approach to clustring. *Pattern Recognition Letters* **12**, 193–198, 1991.
[37] A. Argyrou. Clustering hierarchical data using self-organizing map: a graph-theoretical approach. In J. C. Principe and R. Miikkulainen (eds.) *Advances in Self-Organizing Maps, Proceedings*, vol. 5629. pp 19–27. Springer, 2009.
[38] S. Staniford, J. A. Hoagland and J. M. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security* **10**, 105–136, 2002.
[39] P. Ren, Y. Gao, Z. C. Li, Y. Chen and B. Watson. IDGraphs: intrusion detection and analysis using stream compositing. *IEEE Computer Graphics and Applications* **26**, 28–39, 2006.
[40] K. Abdullah, C. Lee, G. Conti and J. A. Copeland. Visualizing network data for intrusion detection. *Sixth Annual IEEE Information Assurance Workshop - Systems, Man and Cybernetics* pp. 100–108, 2005.
[41] M. Malowidzki. GetBulk worth fixing. *The Simple Times* **10**, 3–6, 2002.
[42] R. Sprenkels and J. P. Martin-Flatin. Bulk Transfers of MIB data. *Technical Report SSC/1999/009*. Communication Systems Division. Swiss Federal Institute of Technology Lausanne, 1999.
[43] R. Sánchez, Á. Herrero and E. Corchado. Visualization and Clustering for SNMP intrusion detection. *Cybernetics and Systems* **44**, 505–532, 2013.
[44] J. Mao and A.K.J. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks* **7**, 16–29, 1996.
[45] R. A. Baeza-Yates. Introduction to data structures and algorithms related to information retrieval. In W. B. Frakes and R. Baeza. Yates, (Eds). Information Retrieval: Data Structures and Algorithms, pp. 13–27. Prentice-Hall, Inc., 1992.

Received 12 November 2014