Clustering and Neural Visualization for Flow-Based Intrusion Detection

Raúl Sánchez, Álvaro Herrero and Emilio Corchado

Abstract To secure a system, potential threats must be identified and therefore, attack features are understood and predicted. Present work aims at being one step towards the proposal of an Intrusion Detection System (IDS) that faces zero-day attacks. To do that, MObile VIsualisation Connectionist Agent-Based IDS (MOVICAB-IDS), previously proposed as a hybrid-intelligent visualization-based IDS, is being upgraded by adding clustering methods. To check the validity of the proposed clustering extension, it faces a realistic flow-based dataset in present paper. The analyzed data come from a honeypot directly connected to the Internet (thus ensuring attack-exposure) and is analyzed 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 · Network flow · Neural projection · Clustering

R. Sánchez · Á. Herrero (✉)
Department of Civil Engineering, University of Burgos,
Avenida de Cantabria s/n, 09006 Burgos, Spain
e-mail: ahcosio@ubu.es

R. Sánchez
e-mail: rsarevalo@ubu.es

E. Corchado
Departamento de Informática y Automática, Universidad de Salamanca,
Plaza de la Merced, s/n, 37008 Salamanca, Spain
e-mail: escorchado@usal.es

Á. Herrero et al. (eds.), *International Joint Conference*, Advances in Intelligent Systems and Computing 369, DOI 10.1007/978-3-319-19713-5_29

1 Introduction

Intrusion Detection (ID) is a field that focuses on the identification of attacks, in both networks and computers. The huge amount of previous work focused on network-based ID can be categorized by different criteria. One of them is the nature of the analysed data and according to that, there are two categories based on the source of data to be analyzed: packets or flows. Some network Intrusion Detection Systems (IDSs) analyze packets travelling along the network and then extracts the information from the different fields in the packet (headers and payload, mainly). On the other hand, some IDSs deals with flows, being defined as "a set of IP packets passing on an observation point in the network during a certain time interval and having a set of common properties" [1].

In flow-based IDSs, rather than looking at all packets in a network, they look at aggregated information of related packets of network traffic in the form of a flow, so the amount of data to be analyzed is summarized and then reduced. With the rise of network speed and number and types of attacks, existing IDSs, face challenges of capturing every packet. Hence a flow-based IDS has an overall lower amount of data to be process, therefore it is the logical choice for high speed networks [2, 3].

MObile VIsualisation Connectionist Agent-Based IDS (MOVICAB-IDS) was proposed [4] as a novel IDS comprising a Hybrid Artificial Intelligent System (HAIS). Its main goal was to apply an unsupervised neural projection model to extract traffic dataset projections and to display them through a mobile visualisation interface. One of its main drawbacks was its dependence on human processing; MOVICAB-IDS could not automatically raise an alarm. Additionally, human users could fail to detect an intrusion even when visualised as an anomalous one, when visually processing big amounts of data [5]. This IDS is being extended by the application of clustering techniques in conjunction with neural visualization, to overcome its limitations.

Based on successful results obtained by upgrading MOVICAB-IDS with clustering techniques to detect different attacks on packet-based data [6, 7], present work focuses on flow-based data. Hence, present work proposes the combination of MOVICAB-IDS and different clustering techniques to analyze a database of flow-based attack situations, generated by the University of Twente [8]. The experimental study in present paper tries to know whether clustering could be more informative applied over the projected data rather than the original flow data captured from the network.

Clustering and neural visualization have been previously applied to the identification of different anomalous situations (network scans, MIB transfer and community string searches) related to the SNMP network protocol [6].

Clustering has been previously applied to intrusion detection: [9] proposes an alert aggregation method, clustering similar alerts into a hyper alert based on category and feature similarity. From a similar perspective, [10] proposes a two-stage

clustering algorithm to analyze the spatial and temporal relation of the network intrusion behaviors' alert sequence. [11] describes a classification of network traces through an improved nearest neighbor method, while [12] applies data mining algorithms for the same purpose and the results of preformatted data are visually displayed. Finally [13] discusses on how the clustering algorithm is applied to intrusion detection and analyzes intrusion detection algorithm based on clustering problems. Differentiating from previous work, the approach proposed in present paper, applies clustering to previously projected data (processed by neural models).

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

2 Proposed Solution

To better detect intrusions, an upgrade of MOVICAB-IDS, combining projection and clustering results is being proposed. In keeping with this idea, present study focuses on flow-based data for the first time.

MOVICAB-IDS [4] is based on the application of different AI paradigms to process the continuous data flow of network traffic. In order to do so, MOVICAB-IDS split massive traffic data into limited datasets and visualises them, thereby providing security personnel with an intuitive snapshot to monitor the events taking place in the observed computer network. The following paradigms are combined within MOVICAB-IDS.

2.1 Cooperative Maximum Likelihood Hebbian Learning

One neural implementation of Exploratory Projection Pursuit (EPP) [14] is Maximum Likelihood Hebbian Learning (MLHL) [15]. It identifies interestingness by maximising the probability of the residuals under specific probability density functions which are non-Gaussian.

An extended version of this model is the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [16] model. CMLHL is based on MLHL [15] adding lateral connections [16], which have been derived from the Rectified Gaussian Distribution [17]. 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 [16] 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 [16] and A a symmetric matrix used to modify the response to the data [16]. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.2 Clustering

Cluster analysis [18, 19] consist in the organization of a collection of data items or patterns (usually represented as a vector of measurements, or a point in a multi-dimensional 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 [20, 21]. There are different approaches to clustering data [18], but given the high number and the strong diversity of the existent clustering methods, a representative technique for partitional as well as hierarchical clustering are applied in present study.

In general terms, there are two main types of clustering techniques: hierarchical and partitional approaches. Hierarchical methods produce a nested series of partitions (illustrated on a dendrogram which is a tree diagram) 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.

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.

3 Experimental Study

As previously stated, the proposed approach is applied to analyze flow-based data. To check the performance of clustering and projection techniques, two different alternatives are considered: clustering on projected (dimensionality-reduced) data, and clustering on original (twelve-dimensional or nine-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.

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. Four different distance measures are applied in present study for *K*-means algorithm, namely: sqEuclidean, Cityblock, Cosine, and Correlation. For agglomerative clustering and based on the way the proximity matrix is updated in the second phase, a variety of linking methods can be designed. Present study has applied the following linking methods: Single, Complete, Ward, Median, Average, Centroid, and Weighted.

3.1 Datasets

The analyzed dataset contains flow-based information from traffic collected by the University of Twente [8], in September 2008. The honeypot was directly connected to the Internet and ran several typical network services, such as ftp, ssh, etc.

This data set, consisting of 14.2 M flows, has been collected by using a honeypot ensuring traffic to be realistic. A honeypot can be defined as an "environment where vulnerabilities have been deliberately introduced to observe attacks and intrusions" [22]. Present work focuses on two segments obtained from the above mentioned database, which has been split in different overlapping segments, as MOVICAB-IDS usually do with network traffic. Every segment contains all the flows whose timestamp is between the segment initial and final time limit. Segment length is stated as 782 s to cover the whole database, whose length is 539,520 s (from the beginning of the first flow to the end of the last one), that is amounts to 6 days and 709 segments. As defined for MOVICAB-IDS, there is a slight time overlap of 10 s between each pair of consecutive segments.

Two out of the 709 generated segments have been chosen for present study:

- Segment 59: two types of data can be found: ssh_conn and irc_sideeffect.
- Segment 545: two types of data can be found: ssh_conn and http_conn.

The following fourteen features were extracted from the database to define every single flow:

- **id**: the ID of the flow.
- **src_ip**: anonymized source IP address (encoded as 32-bit number).
- **dst_ip**: anonymized destination IP address (encoded as 32-bit number).
- **packets**: number of packets in the flow.
- **octets**: number of bytes in the flow.
- **start_time**: UNIX start time (number of seconds).
- **start_msec**: start time (milliseconds part).
- **end_time**: UNIX end time (number of seconds).
- **end_msec**: end time (milliseconds part).
- **src_port**: source port number.
- **dst_port**: destination port number.
- **tcp_flags**: TCP flags obtained by ORing the TCP flags field of all packets of the flow.
- **prot**: IP protocol number.
- **type**: alert type.

Two of the above listed features are not provided to the models: the first one (added to identify each single packet) and the last one (alert type).

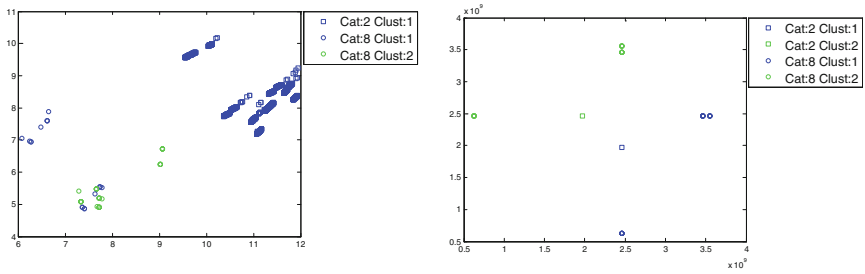
This set of features has been processed in order to summarize the four features related to time, being joined in only one feature, named as flow length. By doing so, new datasets have been generated, whose nine features are also analyzed. All the data sets (detailed time information vs. flow length) have been studied but, as the results are pretty similar, only the results for the second data set are shown in present paper. The four different datasets can be described as follows:

- **Segment 59**: contains 1,215 flows where 56 are from type irc_sideeffect and the rest 1159 are from type ssh_conn. Every segment comprises 12 dimensions.
- **Segment 545**: contains 731 flows where 12 are from type http_conn and the rest 719 are from type ssh_conn. Every segment comprises 12 dimensions.
- **Segment 59 with segment length**: contains 1,215 flows where 56 are from type irc_sideeffect and the rest 1159 are from type ssh_conn. The four dimensions related to time has been joined in one which is segment length calculated from start time to end time and given in milliseconds. Hence, every segment comprises 9 dimensions.
- **Segment 545 with segment length**: contains 731 flows where 12 are from type http_conn and the rest 719 are from type ssh_conn. The four dimensions related to time has been joined in one which is segment length calculated from start time to end time and given in milliseconds. Hence, every segment comprises 9 dimensions.

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 (different colors and symbols). The projections comprise a legend that states the color and symbol used to depict each packet, according to the original category of the data. The following subsections comprise the results obtained by the projection and clustering technique for two of the datasets. Although the four above mentioned datasets (Segment 59, Segment 545, Segment 59 with segment length, and Segment 545 with segment length), only results for the two last ones are shown in this section because the other two are similar.

Segment 59 with segment length.



1.a K-means on projected data: k=2, sqEuclidean distance.

1.b K-means on original data: k=2, sqEuclidean distance.

Fig. 1 Some clustering results under the frame of MOVICAB-IDS through k-means for Segment 59 with segment length

For the irc_sideeffect flows on projected data (Cat. 8 in Fig. 1.a), the clustering splits all the data from this type on two different clusters (Clust. 1 and Clust. 2); the ssh_conn flows (Cat. 2 in Fig. 1.a) are grouped correctly in just one cluster (Clust. 1), however some of the flows from Cat. 8 are clustered together with them. Apart from these two projections, some more experiments have been conducted, whose details: performance, False Positive Rate (FPR) and False Negative Rate (FNR), values of k parameter, etc., can be seen in Table 1.

It can be seen from Table 1 that there is a non-zero False Negative Rate, but this value is worse on original data because the clusters are more mixed although the number of clusters (k parameter) is the same. The results on projected data in the majority of the cases probed, are better than on original (none projected) data.

Some of the run experiments for the agglomerative method with no clustering error are shown in Table 2. It can be seen that, in the case of projected data, the minimum number of clusters with no error is 3, while in the case of original data it

Table 1 K-means results for Segment 59 with segment length

Data	k	Distance criteria	False positive	False negative	Replicates/iterations	Sum of distances
Projected	2	sqEuclidean	0 %	1.4803 %	5/2	2498.16
Original	2	sqEuclidean	47.6974 %	2.3026 %	5/2	8.03752E+19
Projected	4	sqEuclidean	27.7961 %	0 %	5/7	786.499
Original	4	sqEuclidean	47.6974 %	0.7401 %	5/3	3.18651E+19
Projected	6	sqEuclidean	0 %	0 %	5/7	296.276
Original	6	sqEuclidean	0 %	0 %	5/3	7.1228E+16
Projected	2	Cityblock	59.2928 %	0 %	5/8	2065.86
Original	2	Cityblock	47.6974 %	1.8092 %	5/2	8.19316E+10
Projected	4	Cityblock	42.1053 %	0 %	5/10	955.305
Original	4	Cityblock	70.2303 %	0 %	5/2	6.98674E+10
Projected	6	Cityblock	13.8980 %	0 %	5/5	643.167
Original	6	Cityblock	29.6053 %	0 %	5/3	6.98583E+10
Projected	2	Cosine	70.2303 %	0.7401 %	5/2	1.16382
Original	2	Cosine	47.6974 %	2.3026 %	5/2	1.62372
Projected	4	Cosine	59.5395 %	0 %	5/5	0.119684
Original	4	Cosine	47.6974 %	0 %	5/4	0.0647986
Projected	6	Cosine	0 %	0 %	5/6	0.059721
Original	6	Cosine	0 %	0 %	5/5	0.000641182
Projected	2	Correlation	25.1645 %	0.7401 %	5/2	1.02283
Original	2	Correlation	47.6974 %	2.3026 %	5/2	1.95889
Projected	4	Correlation	47.8619 %	0 %	5/7	0.370916
Original	4	Correlation	47.6974 %	0 %	5/4	0.0823528
Projected	6	Correlation	47.8619 %	0 %	5/8	0.0810012
Original	6	Correlation	0 %	0 %	5/5	0.000810621

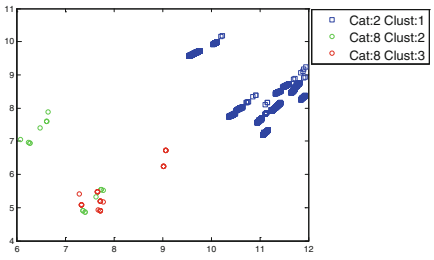
Table 2 Experimental setting of the agglomerative method for Segment 59 segment length

Data	Distance	Linkage	Cutoff	Cluster
Projected	Euclidean	Single	3	3
Projected	sEuclidean	Complete	6	3
Projected	Cityblock	Average	7	3
Projected	Minkowski p = 3	Weighted	4	3
Projected	Chebychev	Complete	5	3
Projected	Mahalanobis	Average	5	3
Projected	Cosine	Single	0.002	5
Projected	Correlation	Complete	0.005	6
Original	Euclidean	Complete	10×10^8	4
Original	Cityblock	Single	13×10^8	5
Original	Minkowski p = 3	Complete	15×10^8	4
Original	Chebychev	Complete	15×10^8	4
Original	Cosine	Average	0.007	6
Original	Correlation	Weighted	0.001	6

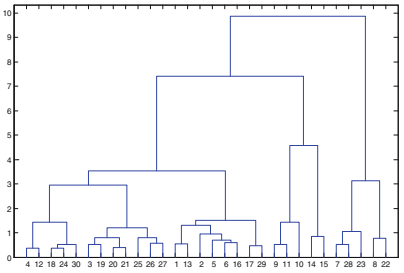
is 4, with appropriate distance method. In the case of original data, the sEuclidean distance and Mahalanobis distance can not be applied because the maximum recursion level has been reached in the first case, and the covariance matrix can not be computed in the second case.

Results for one of the experiments from Table 2 are depicted in Fig. 2, including flow visualization and the associated dendrogram on projected data. The chosen experiment parameters are: sEuclidean distance, complete linkage, cutoff: 6 and 3 groups with no clustering error.

Segment 545 with segment length.
Figure 3 shows the results obtained by k-means on this data. The data has been labeled as follows: ssh_conn flows (Cat. 2) and http_conn (Cat. 6).

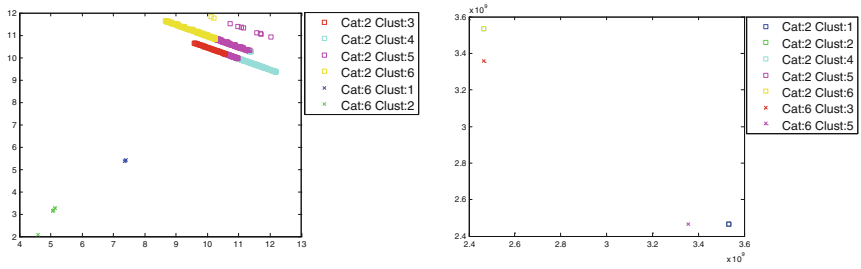


2.a Agglomerative clustering on projected data: sEuclidean, linkage: complete, cutoff: 6.



2.b Corresponding dendrogram.

Fig. 2 Best results of agglomerative clustering under the frame of MOVICAB-IDS for Segment 59 with segment length



3.a K-means on projected data: k=6, Cityblock distance.

3.b K-means on original data: k=6, Cityblock distance.

Fig. 3 Some clustering result under the frame of MOVICAB-IDS through k-means for Segment 545 with segment length

Although very low, original data has a non-zero False Positive Rate value, while projected data has no error. For the ssh_conn flows on projected data (Cat. 2 in Fig. 3.a), the clustering technique groups data with no errors, even though the number of clusters (k parameter) is bigger than the categories, hence some clusters groups only data from the same category. Apart from these two projections, some more experiments have been run, whose details (performance, false positive and false negative rates, values of k parameter, etc.) can be seen in Table 3.

The run experiments for the agglomerative method with no error are shown in Table 4.

It can be seen that, in the case of projected data, the minimum number of clusters with no error is 3, while in the case of original data it is 4, with appropriate distance method. In the case of original data, the sEuclidean distance and Mahalanobis distance can not be applied because the maximum recursion level has been reached in the first case, and the covariance matrix can not be computed in the second case.

Results of one of the best experiments from Table 4 are depicted in Fig. 4, including traffic visualization and the associated dendrogram on projected data. The chosen experiment parameters are: sEuclidean distance, complete linkage, cutoff: 8 and 3 groups with no clustering error.

4 Conclusions

A clustering extension of MOVICAB-IDS has been proposed and applied to a real-life flow-based database obtained from a honeypot at the University of Twente.

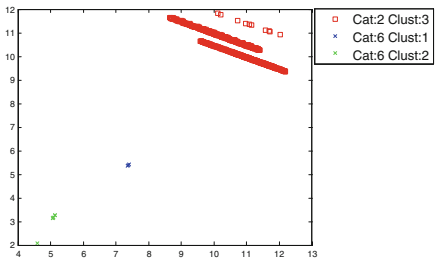
Detailed conclusions about experiments on the different datasets and with several different clustering techniques and criteria can be found in Sect. 3. Experimental results show that some of the applied clustering methods obtain a good clustering performance on the analyzed data, according to false positive and false negative rates. The obtained results vary from the different analyzed datasets and

Table 3 K-means experiments with different conditions for Segment 545 with segment length

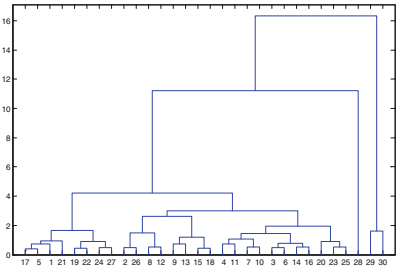
Data	k	Distance criteria	False positive	False negative	Replicates/iterations	Sum of distances
Projected	2	sqEuclidean	42.4863 %	0 %	5/12	1710.13
Original	2	sqEuclidean	49.1803 %	0.8197 %	5/2	3.65729E+17
Projected	4	sqEuclidean	15.9836 %	0 %	5/21	854.837
Original	4	sqEuclidean	0 %	0 %	5/3	3.88519E+10
Projected	6	sqEuclidean	0 %	0 %	5/20	121.144
Original	6	sqEuclidean	0 %	0 %	5/12	1.13974E+10
Projected	2	Cityblock	46.7213 %	0 %	5/8	814.403
Original	2	Cityblock	49.1803 %	0.8197 %	5/2	2.11713E+09
Projected	4	Cityblock	11.7486 %	0 %	5/21	593.5
Original	4	Cityblock	22.2678 %	0 %	5/14	1.05984E+09
Projected	6	Cityblock	0 %	0 %	5/16	402.065
Original	6	Cityblock	10.5191 %	0 %	5/12	1.0584E+09
Projected	2	Cosine	51.3661 %	0 %	5/9	1.37952
Original	2	Cosine	49.1803 %	0.8197 %	5/2	0.00345234
Projected	4	Cosine	25.1366 %	0 %	5/36	0.259027
Original	4	Cosine	0 %	0 %	5/3	1.04725E-09
Projected	6	Cosine	0 %	0 %	5/26	0.105203
Original	6	Cosine	0 %	0 %	5/12	3.07186E-10
Projected	2	Correlation	55.6011 %	0 %	5/8	53.3814
Original	2	Correlation	49.1803 %	0.8197 %	5/2	0.00437153
Projected	4	Correlation	24.8634 %	0 %	5/12	19.5017
Original	4	Correlation	0 %	0 %	5/3	1.14901E-09
Projected	6	Correlation	17.4863 %	0 %	5/27	13.6746
Original	6	Correlation	0 %	0 %	5/10	3.36364E-10

Table 4 Experimental setting of the agglomerative method for Segment 545 with segment length

Data	Distance	Linkage	Cutoff	Cluster
Projected	Euclidean	Single	8	3
Projected	sEuclidean	Complete	8	3
Projected	Cityblock	Average	12	3
Projected	Minkowski p = 3	Weighted	8	3
Projected	Chebychev	Single	5	3
Projected	Mahalanobis	Single	6	3
Projected	Cosine	Complete	0.08	3
Projected	Correlation	Average	0.05	8
Original	Euclidean	Single	1×10^8	4
Original	Cityblock	Complete	1×10^8	4
Original	Minkowski p = 3	Average	1×10^8	4
Original	Chebychev	Weighted	1×10^8	4
Original	Cosine	Single	0.0001	4
Original	Correlation	Complete	0.0001	4



4.a Agglomerative clustering on projected data: sEuclidean, linkage: single, cutoff: 6.



4.b Corresponding dendrogram.

Fig. 4 Best results of agglomerative clustering under the frame of MOVICAB-IDS for Segment 545 with segment length

the behavior of the applied clustering techniques. These results are consistent with those previously obtained for other SNMP anomalous situations [6, 7].

There is no distance criterion which shows the best results, hence its selection will depend on the analyzed data. Comparing projected data results with the ones from original data, it can be said that projected data has better results (fewer number of groups with no errors).

Finally, it can be concluded that the applied methods are able to detect anomalous situations. It has been proven that clustering methods could help in intrusion detection over flow-based network data. On the other hand, using clustering, automatic response could be added to MOVICAB-IDS, to quickly abort intrusive actions while happening.

References

1. Quittek, J., Zseby, T., Claise, B., Zander, S.: Requirements for IP flow information export (IPFIX)
2. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B.: An overview of IP flow-based intrusion detection. IEEE Commun. Surv. Tutor. **12**, 343–356 (2010)
3. Sperotto, A., Pras, A.: Flow-based intrusion detection. In: IFIP/IEEE International Symposium on Integrated Network Management (IM), 2011, pp. 958–963 (2011)
4. Corchado, E., Herrero, Á.: Neural visualization of network traffic data for intrusion detection. Appl. Soft Comput. **11**, 2042–2056 (2011)
5. Yorn-Tov, E., Inbar, G.F.: Selection of relevant features for classification of movements from single movement-related potentials using a genetic algorithm. In: 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2001, vol. 2, pp. 1364–1366 (2001)
6. Sánchez, R., Herrero, Á., Corchado, E.: Clustering extension of MOVICAB-IDS to identify SNMP community searches. Logic J. IGPL **23**, 121–140 (2015)
7. Sánchez, R., Herrero, Á., Corchado, E.: Visualization and clustering for SNMP intrusion detection. Cybern. Syst. Int. J. **44**, 505–532 (2013)
8. Sperotto, A., Sadre, R., Vliet, F.v., Pras, A.: A Labeled Data Set For Flow-based Intrusion Detection, pp. 39–50. IP Operations and Management, Berlin (2009)
9. Zheng, Q.H., Xuan, Y.G., Hu, W.H.: An IDS alert aggregation method based on clustering. In: Zhang, H., Shen, G., Jin, D. (eds.): Advanced Research on Information Science, Automation and Material System, Pts 1-6, vol. 219–220, pp. 156–159. Trans Tech Publications Ltd, Stafa-Zurich (2011)
10. Qiao, L.B., Zhang, B.F., Lai, Z.Q., Su, J.S.: 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)
11. Jiang, S., Song, X., Wang, H., Han, J.-J., Li, Q.-H.: A clustering-based method for unsupervised intrusion detections. Pattern Recogn. Lett. **27**, 802–810 (2006)
12. Cui, K.Y.: IEEE: Research on Clustering Technique in Network Intrusion Detection. IEEE Computer Society, Los Alamitos (2012)
13. Ge, L., Zhang, C.Q.: The application of clustering algorithm in intrusion detection system. In: Jin, D., Lin, S. (eds.) Advances in Future Computer and Control Systems, vol. 159, pp. 77–82. Springer, Berlin (2012)
14. Friedman, J.H., Tukey, J.W.: A projection pursuit algorithm for exploratory data-analysis. IEEE Trans. Comput. **23**, 881–890 (1974)
15. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and minimum likelihood hebbian learning for exploratory projection pursuit. Data Min. Knowl. Disc. **8**, 203–225 (2004)
16. Corchado, E., Fyfe, C.: Connectionist techniques for the identification and suppression of interfering underlying factors. Int. J. Pattern Recognit. Artif.Intell. **17**, 1447–1466 (2003)
17. Seung, H.S., Socci, N.D., Lee, D.: The rectified Gaussian distribution. Adv. Neural Inf. Process. Syst. **10**, 350–356 (1998)
18. Jain, A.K., Murty, M.N, Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31** (1999)
19. Xu, R., Wunsch, D.C.: Clustering. Wiley, New York (2009)
20. Andreopoulos, B., An, A., Wang, X., Schroeder, M.: A roadmap of clustering algorithms: finding a match for a biomedical application. Brief Bioinform **10**, 297–314 (2009)
21. Zhuang, W.W., Ye, Y.F., Chen, Y., Li, T.: Ensemble clustering for Internet security applications. IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev. **42**, 1784–1796 (2012)
22. Pouget, F., Dacier, M.: Honeypot-based forensics. In: Proceedings of the AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004), 23–27 May 2004, Brisbane, Australia (2004)