

Multiobjective Evolutionary Clustering Approach to Security Vulnerability Assessments

G. Corral, A. Garcia-Piquer, A. Orriols-Puig, A. Fornells, and E. Golobardes

Grup de Recerca en Sistemes Intel·ligents
La Salle - Universitat Ramon Llull
c/ Quatre Camins 2, 08022 Barcelona (Spain)
{guiomar, alvarog, aorriols, afornells, elisabet}@salle.url.edu

Abstract. Network vulnerability assessments collect large amounts of data to be further analyzed by security experts. Data mining and, particularly, unsupervised learning can help experts analyze these data and extract several conclusions. This paper presents a contribution to mine data in this security domain. We have implemented an evolutionary multiobjective approach to cluster data of security assessments. Clusters hold groups of tested devices with similar vulnerabilities to detect hidden patterns. Two different metrics have been selected as objectives to guide the discovery process. The results of this contribution are compared with other single-objective clustering approaches to confirm the value of the obtained clustering structures.

Keywords: Multiobjective Optimization, Evolutionary Algorithm, Unsupervised Learning, Clustering, Network Security, AI applications.

1 Introduction

Information Technology and the communication networks that support it have gradually changed into critical resources for organizations. The combination of computer and communication technologies offers many benefits, but introduces weaknesses. Consequently periodic audits and vulnerability assessments are needed. Vulnerability assessment is the process of identifying and quantifying vulnerabilities in systems or networks [16]. As time and cost may restrict its depth, the automation of the involved processes is essential, specially those related to the data analysis. In addition, a comprehensive network security analysis must coordinate diverse sources of information to support large scale visualization and intelligent response [7]. So security applications require some intelligence to detect malicious data, unauthorized traffic or vulnerabilities [8].

Artificial intelligence can be applied to vulnerability assessment results. The use of clustering for discovering hidden patterns through the identification of device groups with similar vulnerabilities has been demonstrated [3]. Different validity techniques to select the best clustering solution have been analyzed [4]. These contributions have been included in *Analia*, a computer-aided system to

automate network security tests [3]. *Analia* helps security analysts, but it has a drawback. Two independent processes are needed: select (1) the clustering approach and (2) the validity index. The best clustering solution depends on the selected validity index, as each index may pursue different goals. Moreover, the goals of clustering and the index may not be aligned. Analysts also ask for a process where configuration parameters not related to their domain are provided.

This paper presents an improvement of *Analia* based on including an evolutionary multiobjective (EMO) clustering algorithm [12] to group network devices with similar vulnerabilities after a vulnerability assessment. The optimization of the different validity indices will be used as the goal to cluster tested devices in groups with similar vulnerabilities. This new approach will allow security analysts to obtain the best clustering solution considering different criteria simultaneously. In addition, this selection will become a transparent process to analysts, due to the fact that this technique includes the optimization of the selected criteria in the clustering process itself. Then analysts will not need to care about the difference between validity indices and will be able to focus only on the obtained clustering results, which is their actual concern.

The remainder of this paper is organized as follows. Section 2 describes related work on machine learning in the security domain. Section 3 details our clustering multiobjective evolutionary approach. Section 4 describes *Analia* with single and multiple optimization clustering. Section 5 summarizes the experimentation and results. Conclusions and further work are given in Section 6.

2 Related work

The large volume of data generated by vulnerability assessments has unleashed the need of using enhanced techniques to recognize malicious behavior patterns or unauthorized changes in data networks [8]. These domains are usually defined by sets of unlabeled examples, and experts aim at extracting novel and useful information about the network behavior that helps them detect vulnerabilities, among others. In this context, clustering appears as an appealing approach that permits grouping network devices with similar security vulnerabilities, thence, identifying potential threats to the network.

Several clustering techniques have been applied to the network security domain thus far. For example, K-means [13] has been used to group similar alarm records [2] and to detect network intrusions [15]. SOM [14] has been employed to detect computer attacks [8], network intrusions [9], and anomalous traffic [17]. Despite the success of these applications, all these clustering techniques guide the discovery process with a single criterion. For example, K-means minimizes the total within-cluster variance and tends to find spherical clusters [13]. Nevertheless, we are interested in obtaining clusterings that satisfy different criteria. For this purpose, several authors have proposed to run different clustering techniques to obtain different structures, and then, involve the network expert into the process in order to manually select the best structure according to certain predetermined validation methods.

In this paper, we propose to automatize this process by guiding the clustering process with different objectives. To achieve this, we employ a multiobjective clustering approach [12]. Among the different techniques for multiobjective optimization such as simulated annealing or ant colony optimization, we base on evolutionary algorithms since they (1) employ a population based-search, evolving a set of optimal trade-offs among objectives, (2) use a flexible knowledge representation that can be easily adapted to the type of data of our domain, and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions. In addition, EMO clustering has been successfully applied to important real-world problems such as intrusion detection [1], formation of cluster-based sensing networks in wireless sensor networks [19], and creation of security profiles [11].

3 Evolutionary Multiobjective Clustering Approach

This section explains the design of the EMO approach employed to evolve data clusterings that optimize several objectives. Our approach is based on the MOCK system [12] which uses the PESA-II algorithm. PESA-II evolves a set of solutions, where each one defines a possible clustering configuration. In what follows, the knowledge representation, the process organization to obtain the Pareto set of solutions, and the method to recover the best solution among the ones in the Pareto set are briefly explained.

Representation. The system evolves a population of individuals of size N , where each individual represents a cluster structure for the problem. The individual is represented in a vector of n integers: x_1, x_2, \dots, x_n . Then, x_i indicates that instance i is connected to instance x_i , that is, that they belong to the same cluster.

Process organization. The result of the algorithm is a Pareto set of solutions, that is, a set of individuals for which it does not exist any other individual in the population that dominates them¹. The population is evolved as follows. We first initialize the population with an individual that represents the minimum spanning tree (MST) constructed from the undirected, fully connected labeled graph that represents the Euclidean distance between each pair of examples. In addition, we also create $N - 1$ individuals that progressively remove the links with highest distance from the original MST. Then, the population iteratively goes through a process of selection, crossover, and mutation as described in [12].

Selection of the best solution. After evolving a set of non-dominated solutions, we use the following criterion to recover the best solution. The system returns the solutions between 6 and 9 clusters because, according to the experts, the devices included in the dataset can be broadly categorized in those groups. Furthermore, these solutions are the best ones that optimize the couple connectivity/deviation.

¹ In multiobjective algorithms, a solution x dominates another solution y if all the objectives of x are better than the corresponding objectives of y .

4 Analia

This section explains the architecture of *Analia* and the inclusion of evolutionary multiobjective clustering to improve data analysis in this security domain.

4.1 Single-objective clustering in *Analia*

Analia is the data analysis module of *Consensus* [5]. Whereas *Consensus* gathers security data, *Analia* includes AI to help analysts after a vulnerability assessment. *Analia* finds resemblances within tested devices and clustering aids analysts in the extraction of conclusions. Afterwards, the best results are selected by applying cluster validity indices [4]. Then explanations of clustering results are included to give a more comprehensive response [3]. Figure 1 depicts the architecture of *Analia* and its interaction with *Consensus*.

Previous work has validated the incorporation of K -means, X -means and SOM in *Analia* [4]. A drawback of this unsupervised domain is that no previous knowledge of the possible existing classes is known. So several executions are run to select the best one by using cluster validation techniques. The most used indices found in the literature have been included in *Analia*: Dunn [10], Davies-Bouldin [6] and Silhouette [18]. Besides, two indices have been designed ad hoc for this security domain: *Intracohesion* and *Intercohesion* factors [4]. The process to obtain the best partition in *Analia* is summarized in the following steps:

1. Select the clustering approach and execute it on *Consensus* dataset
2. Select the executions to analyze
3. Calculate validation indices for each execution
4. Select the validation index as decision criterion and obtain the best execution

When having run a clustering algorithm several times varying parameters or different clustering algorithms, several clustering solutions are obtained. Then, any of the mentioned validity indices can be selected to obtain the best solution over a set of executions. High values for *Dunn* and *Silhouette*, whereas low values for *DB* are preferred. Regarding *Cohesion factors*, the best solution should consider the highest *Intracohesion factor* and the lowest *Intercohesion factor*.

Security analysts do not usually care about the selected index criteria, but about the best clustering solution. Thus an automated mechanism has been designed to combine the calculated validity factors based on a weighted voting scheme. It ranks the list of options based on the number of votes each option earns, considering that some votes carry more weight than others. We give more importance to *Cohesion factors* by assigning a higher weight to them. Then, analysts can easily get the best partition without any previous knowledge about validity indices, knowledge not usually related with their study area.

The main drawback of this option is focused on the different decisions not directly related to their domain that security analysts must consider. This two-step process of selecting a clustering technique and, afterwards, applying validation indices may slow down the whole process. Next section presents a contribution to improve this statement. If validation indices are considered as initial goals of the clustering approach, the obtained clustering solutions will optimize the selected indices, thus reducing the process into a single step.

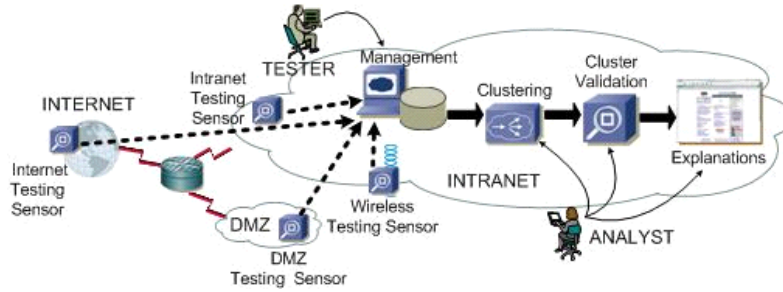


Fig. 1. Architecture of *Consensus* system and *Analia* data analysis module.

4.2 Multiobjective clustering in *Analia*

We have integrated the designed EMO clustering approach into *Analia*, optimizing two complementary objectives: the overall deviation and the connectivity. The former is related to cluster compactness and the latter is based on connectedness of clusters. These goals are aligned to *Cohesion* validation indices. The *Intracohesion factor* evaluates the cohesion between the elements of a cluster in terms of the vulnerabilities common to the members of that cluster [4], concept equivalent to cluster compactness. The *Intercohesion factor* evaluates the cohesion between clusters, considering the vulnerabilities common to elements of different clusters [4], concept equivalent to connectivity.

When including this EMO clustering in *Analia*, the process to obtain the best partition is summarized in a single step: execute the EMO clustering approach. All the other aforementioned steps are included in that single phase. The EMO clustering approach obtains a set of non-dominated solutions that optimize both deviation and connectivity in a single run. So the algorithm needs to be run only once. Note that, as the validation indices have been included in the search process, there is no need to calculate those indices afterwards. The best executions with the best number of clusters will be automatically obtained.

5 Experiments

The EMO clustering approach presented in this paper has run on the *Consensus* dataset, which contains information regarding port scanning, operating system fingerprinting and vulnerability testing of a data network. This dataset has been extracted from real security tests performed at La Salle (Universitat Ramon Llull) network. These assessments have been executed over 90 network devices, including public and internal servers, alumni laboratories and staff computers.

Alumni lab computers are the most restricted devices. The IT department installs a unique image on them, so any other software is forbidden. Thus lab devices should be grouped in the same cluster. If new software has been illegally installed or its configuration has been modified, it will be easy to identify as this rogue device should be separated in a new cluster. On the other hand, staff

computers are administered by their owners and hence different patterns will be found. Several servers with different Internet services and different operating systems have been audited, so their classification may also vary.

A solution for an EMO clustering run on *Consensus* dataset is shown in Figure 2. The best executions should find good tradeoffs between the two objectives and they are indicated by a circle centered around the solution. When analyzing the best clustering solutions, the number of clusters vary between 6 and 9. Some clusters are very clear and get repeated in all solutions. There is a cluster that always contains 14 PCs of lab1, 24 PCs of lab2 and 7 PCs of lab3, making a total of 45 devices. However this cluster should contain 46 devices, as lab3 was composed of 8 PCs. Then, it is very easy to discover that a device has been manipulated in that lab and the faked device is included in a cluster with a single element in all clustering solutions. Another big cluster is composed of all internal Linux servers. This cluster contains 27 devices that share the same operating system, although their open services are different. The rest of the devices are grouped in small clusters, depending on their operating system and the offered services. It is remarkable that 3 devices are separated always in 3 single clusters, showing their dissimilarity in comparison with the rest of the elements in the dataset. They correspond to the fake device of a lab, to the wireless access control server which is a Linux device but with specific peculiarities and, finally, to a Sun Solaris server.

Cohesion factors have been calculated to evaluate the correctness of the different solutions and the alignment between these indices and the EMO clustering objectives, overall deviation and connectivity. Results have shown that the best executions of EMO clustering also obtain the best values of *Cohesion factors*, achieving the best values of Intracohesion = 0.896 and Intercohesion = 0.38. The range of these indices is [0..1], preferring high values for *Intracohesion* and low values for *Intercohesion* indices.

Single-objective clustering algorithms have also been run on the same dataset to compare their solutions. *K*-means has run for a range of different numbers of clusters $k \in 3..10$ and different seeds. Considering *Cohesion* factors, the best solutions also obtain a number of clusters between 6 and 9. However, the calculated

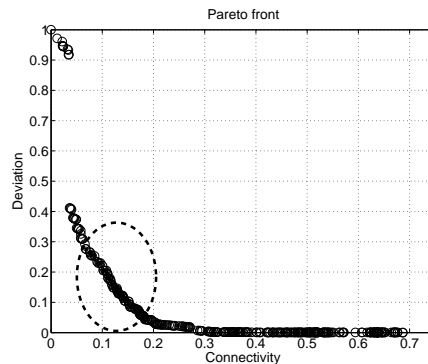


Fig. 2. EMO clustering output on *Consensus* with the deviation and the connectivity.

Table 1. Summary of the *Cohesion* factors for *K*-means, *X*-means and EMO clustering for different number of clusters.

Clusters	<i>K</i> -means		<i>X</i> -means		EMO Clustering	
	Intrach.	Interch.	Intrach.	Interch.	Intrach.	Interch.
6	0.662	0.597	0.559	0.546	0.858	0.451
7	0.626	0.526	0.628	0.538	0.879	0.419
8	0.672	0.591	0.621	0.531	0.895	0.380
9	0.662	0.538	0.563	0.517	0.866	0.371

Cohesion factors are lower than EMO results. The runs of *X*-means conclude in 7 for the best value of *K*. Again, the *Cohesion* factors are lower than EMO clustering results. Both partitioning algorithms show clustering structures without one-element clusters, or at least, only one unique cluster with one element.

A summary of the obtained *Cohesion* factors of the different clustering approaches are shown in Table 1. The best values for all possible configurations of number of clusters are obtained with the EMO approach. This is the approach that tries to optimize the two objectives more directly related to *Cohesion* factors. On the other hand, partition methods minimize only overall deviation and thus *Cohesion* factor results are not as good as the EMO clustering approach.

Network security experts have also analyzed the results obtained after clustering the dataset. Regarding to EMO clustering solution, the high number of clusters with a single element allows the location of outlier devices. But this approach returns a higher number of clusters for the same dataset, compared to the partitioning methods.

6 Conclusions

This paper has presented the incorporation of an evolutive multiobjective clustering algorithm based on PESA-II to analyze data from vulnerability assessments in a network security domain. This approach benefits from the use of multiple objectives. The achieved clustering solutions overcome the results obtained with different single-objective clustering algorithms, like *K*-means or *X*-means. Besides, the use of this EMO approach permits reducing the efforts spent by security analysts in the clustering phase. Security analysts do not need to have previous knowledge of cluster validation indices in order to obtain the best solution of a set of clustering executions. When using this EMO clustering approach, the pursued goals that guide the search for the best solutions are aligned with the validity indices. Therefore, the obtained clustering solutions comply with validity requirements. Then, a subsequent phase where validity indices are applied is not necessary. Once the clustering results are shown to security analysts, their task starts analyzing the characteristics of the obtained clusters. Clusters will group devices with similar operating systems, open ports, and vulnerabilities.

Further work will focus on the inclusion of different metrics as objectives of the EMO clustering approach. *Cohesion* factors and other validity indices will be incorporated as input goals to be optimized.

Acknowledgements

This work has been supported by the MCYT-FEDER projects TIN2006-15140-C03-03, TIN2008-06681-C06-05 and by the Generalitat de Catalunya (2005SGR-302). We want to thank La Salle - URL for the support to our research group.

References

1. K. Anchor, J. Zydallis, and G. Gunsch. Extending the computer defense immune system: Network intrusion detection with a multiobjective evolutionary programming approach. In *1st Conf. on Artificial Immune Systems*, pages 12–21, 2002.
2. E. Bloedorn, L. Talbot, and D. DeBarr. *Data Mining Applied to Intrusion Detection: MITRE Experiences*. Marcus A. Maloof, ed., Springer Verlag, 2005.
3. G. Corral, E. Armengol, A. Fornells, and E. Golobardes. Data security analysis using unsupervised learning and explanations. In *Innovations in Hybrid Intelligent Systems*, volume 44 of *Advances in Soft Computing*, pages 112–119. Springer, 2008.
4. G. Corral, A. Fornells, E. Golobardes, and J. Abella. Cohesion factors: improving the clustering capabilities of consensus. In *Intelligent Data Engineering and Automated Learning - IDEAL, LNCS*, volume 4224, pages 488–495. Springer, 2006.
5. G. Corral, A. Zaballo, X. Cadenas, and A. Grane. A distributed vulnerability detection system for an intranet. In *Proceedings of the 39th IEEE International Carnahan Conference on Security Technology (ICCST'05)*, pages 291–295, 2005.
6. D.L. Davies and D.W. Bouldin. A cluster separation measure. In *IEEE Transactions on Pattern Analysis and Machine Learning*, volume 4, pages 224–227, 1979.
7. J. Dawkins and J. Dale. A systematic approach to multi-stage network attack analysis. *2nd. IEEE Int. Information Assurance Workshop (IWIA'04)*, 2004.
8. L. DeLooze. Classification of computer attacks using a self-organizing map. In *Proc. of the 2004 IEEE Workshop on Information Assurance*, pages 365–369, 2004.
9. M.O. Depren, M. Topallar, E. Anarim, and K. Ciliz. Network-based anomaly intrusion detection system using soms. In *Proc. of the IEEE 12th Signal Processing and Communications Applications Conference*, pages 76–79, 2004.
10. J.C. Dunn. Well separated clusters and optimal fuzzy partitions. In *Journal of Cybernetics*, volume 4, pages 95–104, 1974.
11. M. Gupta, J. Rees, A. Chaturvedi, and J. Chi. Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. *Decision Support Systems*, 41(3):592–603, March 2006.
12. J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, Feb. 2007.
13. J.A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.
14. T. Kohonen. *Self-Organizing Maps*. Springer, 3rd. Edition, 2000.
15. K. Leung and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proc. 28th Australasian CS Conf.*, volume 38, 2005.
16. T.R. Peltier, J. Peltier, and J. Blackley. *Managing a Network Vulnerability Assessment*. Auerbach Publishers Inc., 2003.
17. M. Ramadas, S. Ostermann, and B. C. Tjaden. Detecting anomalous network traffic with self-organizing maps. In *RAID'03:Proc. 6th Symposium on Recent Advances in Intrusion Detection*, volume 2820, pages 36–54, 2003.
18. P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. In *J. of Comp. Applic. in Math*, volume 20, pages 53–65, 1987.
19. E. Yang, A. Erdogan, T. Arslan, and N. Barton. Multi-objective evolutionary optimizations of a space-based reconfigurable sensor network under hard constraints. In *Symp. on Bioinspired, Learning, and Int. Syst. for Security*, pages 72–75, 2007.