Agents and Neural Networks for Intrusion Detection

Álvaro Herrero[1] and Emilio Corchado[1]

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

Abstract. Up to now, several Artificial Intelligence (AI) techniques and paradigms have been successfully applied to the field of Intrusion Detection in Computer Networks. Most of them were proposed to work in isolation. On the contrary, the new approach of hybrid artificial intelligent systems, which is based on the combination of AI techniques and paradigms, is probing to successfully address complex problems. In keeping with this idea, we propose a hybrid use of three widely probed paradigms of computational intelligence, namely Multi-Agent Systems, Case Based Reasoning and Neural Networks for Intrusion Detection. Some neural models based on different statistics (such as the distance, the variance, the kurtosis or the skewness) have been tested to detect anomalies in packet-based network traffic. The projection method of Curvilinear Component Analysis has been applied for the first time in this study to perform packet-based intrusion detection. The proposed framework has been probed through anomalous situations related to the Simple Network Management Protocol and normal traffic.

Keywords: Multiagent Systems, Case Based Reasoning, Artificial Neural Networks, Unsupervised Learning, Projection Methods, Computer Network Security, Intrusion Detection.

1 Introduction

Firewalls are the most widely used tools for securing networks, but Intrusion Detection Systems (IDS's) are becoming more and more popular [1]. IDS's monitor the activity of the network with the purpose of identifying intrusive events and can take actions to abort these risky events.

A wide range of techniques have been used to build IDS's. On the one hand, there have been some previous attempts to take advantage of agents and Multi-Agent Systems (MAS) [2] in the field of Intrusion Detection (ID) [3], [4], [5], including the mobile-agents approach [6], [7]. On the other hand, some different machine learning models – including Data Mining techniques and Artificial Neural Networks (ANN) – have been successfully applied for ID [8], [9], [10], [11].

Additionally, some other Artificial Intelligence techniques (such as Genetic Algorithms and Fuzzy Logic, Genetic Algorithms and K-Nearest Neighbor (K-NN) or K-NN and ANN among others) [12] [13] have been combined in order to face ID from a hybrid point of view. This paper employs a framework based on a dynamic

multiagent architecture employing deliberative agents capable of learning and evolving with the environment [14]. These agents may incorporate different identification or projection algorithms depending on their goals. In this case, a neural model based on the study of some statistical features (such as the variance, the interpoint distance or the skew and kurtosis indexes) will be embedded in such agents. One of the main novelties of this paper is the application of Curvilinear Component Analysis (CCA) for packet-based intrusion detection.

The overall architecture of this paper is the following: Section 2 outlines the ID multiagent system, section 3 describes the neural models applied in this research, section 4 presents some experimental results and finally section 5 goes over some conclusions and future work.

2 Agent-based IDS

An ID framework, called Visualization Connectionist Agent-Based IDS (MOVICAB-IDS) [14] and based on software agents [2] and neural models, is introduced. This MAS incorporates different types of agents; some of the agents have been designed as CBR-BDI agents [15], [16] including an ANN for ID tasks, while some others are reactive agents. CBR-BDI agents use Case Based Reasoning (CBR) systems [17] as a reasoning mechanism, which allows them to learn from initial knowledge, to interact autonomously with the environment, users and other agents within the system, and to have a large capacity for adaptation to the needs of its surroundings.

MOVICAB-IDS includes deliberative agents using a CBR architecture. These CBR-BDI agents work at a high level with the concepts of Believes, Desires and Intentions (BDI) [18]. CBR-BDI agents have learning and adaptation capabilities, what facilitates their work in dynamic environments.

The extended version of the Gaia methodology [19] was applied, and some roles and protocols where identified after the Architectural Design Stage [14]. The six agent classes identified in the Detailed Design Stage were: SNIFFER, PREPROCESSOR, ANALYZER, CONFIGURATIONMANAGER, COORDINATOR and VISUALIZER.

2.1 Agent Classes

The agent classes previously mentioned are described in the following paragraphs.

Sniffer Agent
This reactive agent is in charge of capturing traffic data. The continuous traffic flow is captured and split into segments in order to send it through the network for further process. As these agents are the most critical ones, there are cloned agents (one per network segment) ready to substitute the active ones when they fail.

Preprocessor Agent
After splitting traffic data, the generated segments are preprocessed to apply subsequent analysis. Once the data has been preprocessed, an analysis for this new piece of data is requested.

Analyzer Agent
This is a CBR-BDI agent embedding a neural model within the adaptation stage of its CBR system that helps to analyze the preprocessed traffic data. This agent is based on the application of different neural models allowing the projection of network data. In this paper, PCA [20], CCA [21], MLHL [22] and CHMLHL [23] (See Section 3) have been applied for comparison reasons. This agent generates a solution (getting an adequate projection of the preprocessed data) by retrieving a case and analyzing the new one using a neural network. Each case incorporates several features, such as segment length (in ms), total number of packets and neural model parameters among others. A further description of the CBR four steps for this agent can be found in [14].

ConfigurationManager Agent
The processes of data capture, split, preprocess and analysis depends on the values of several parameters, as for example: packets to capture, segment length, features to extract... This information is managed by the CONFIGURATIONMANAGER reactive agent, which is in charge of providing this information to some other agents.

Coordinator Agent
There can be several instances (from 1 to *m*) of the ANALYZER class of agent. In order to improve the efficiency and perform a real-time processing, the preprocessed data must be dynamically and optimally assigned to ANALYZER agents. This assignment is performed by the COORDINATOR agent.

Visualizer Agent
At the very end of the process, this interface agent presents the analyzed data to the network administrator by means of a functional and mobile visualization interface. To improve the accessibility of the system, the administrator may visualize the results on a mobile device, enabling informed decisions to be taken anywhere and at any time.

3 Neural Projection Models

Projection models are used as tools to identify and remove correlations between problem variables, which enable us to carry out dimensionality reduction, visualization or exploratory data analysis. In this study, some neural projection models, namely PCA, MLHL, CMLHL and CCA have been applied for ID.

Principal Component Analysis (PCA) [20] is a standard statistical technique for compressing data; it can be shown to give the best linear compression of the data in terms of least mean square error. There are several ANN which have been shown to perform PCA [24], [25], [26]. It describes the variation in a set of multivariate data in

terms of a set of uncorrelated variables each of which is a linear combination of the original variables. Its goal is to derive new variables, in decreasing order of importance, which are linear combinations of the original variables and are uncorrelated with each other.

Curvilinear Component Analysis (CCA) [21] is a nonlinear dimensionality reduction method. It was developed as an improvement on the Self Organizing Map (SOM) [27], trying to circumvent the limitations inherent in some linear models such as PCA. CCA is performed by a self-organised neural network calculating a vector quantization of the submanifold in the data set (input space) and a nonlinear projection of these quantising vectors toward an output space.

As regards its goal, the projection part of CCA is similar to other nonlinear mapping methods, as it minimizes a cost function based on interpoint distances in both input and output spaces. Quantization and nonlinear mapping are separately performed: firstly, the input vectors are forced to become prototypes of the distribution using a vector quantization (VQ) method, and then, the output layer builds a nonlinear mapping of the input vectors.

Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [23] extends the MLHL model [22] that is a neural implementation of Exploratory Projection Pursuit (EPP). The statistical method of EPP [28] linearly project a data set onto a set of basis vectors which best reveal the interesting structure in data. MLHL identifies interestingness by maximising the probability of the residuals under specific probability density functions which are non-Gaussian.

CMLHL extends the MLHL paradigm by adding lateral connections [23], which have been derived from the Rectified Gaussian Distribution [29]. The resultant model 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.

4 Experiments and Results

In this work, the above described neural models have been applied to a real traffic data set [11] containing “normal” traffic and some anomalous situations. These anomalous situations are related to the Simple Network Management Protocol (SNMP), known by its vulnerabilities [30]. Apart from “normal” traffic, the data set includes: SNMP ports sweeps (scanning of network hosts at different ports - a random port number: 3750, and SNMP default port numbers: 161 and 162), and a transfer of information stored in the Management Information Base (MIB), that is, the SNMP database.

This data set contains only five variables extracted from the packet headers: timestamp (the time when the packet was sent), protocol, source port (the port of the source host that sent the packet), destination port (the destination host port number to which the packet is sent) and size: (total packet size in Bytes). This data set was generated in a medium-sized network so the “normal” and anomalous traffic flows were known in advance. As SNMP is based on UDP, only 5866 UDP-based packets were included in the dataset. In this work, the performance of the previously

described projection models (PCA, CCA, MLHL and CMLHL) has been analysed and compared through this dataset (See Figs. 1 and 2.).

PCA was initially applied to the previously described dataset. The PCA projection is shown in Fig. 1.a. After analysing this projection, it is discovered that the normal traffic evolves in parallel straight lines. According to the parallelism to normal traffic, PCA is only able to identify the port sweeps (Groups 3, 4 and 5 in Fig. 1.b). On the contrary, it fails to detect the MIB information transfer (Groups 1 and 2 in Fig. 1.b) because the packets in this anomalous situation evolve in a direction parallel to the "normal" one.

Fig. 1.b shows the MLHL projection of the dataset. Once again, the normal traffic evolves in parallel straight lines. There are some other groups (Groups 1, 2, 3, 4 and 5 in Fig. 1.a) evolving in an anomalous way. In this case, all the anomalous situations contained in the dataset can be identified due to their non-parallel evolution to the normal direction. Additionally, in the case of the MIB transfer (Groups 1 and 2 in Fig. 1.b), the high concentration of packets must be considered as an anomalous feature.

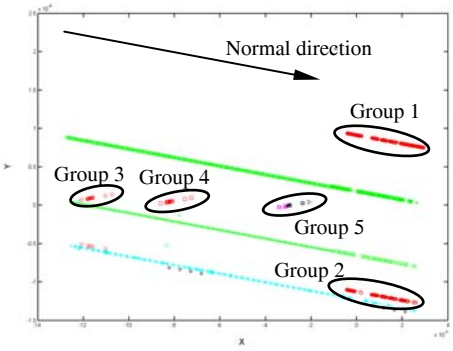

Fig. 1.a PCA projection.

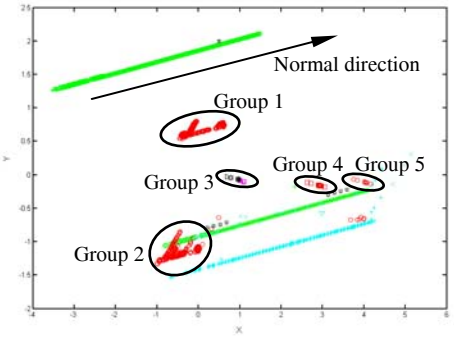

Fig. 1.b MLHL projection.

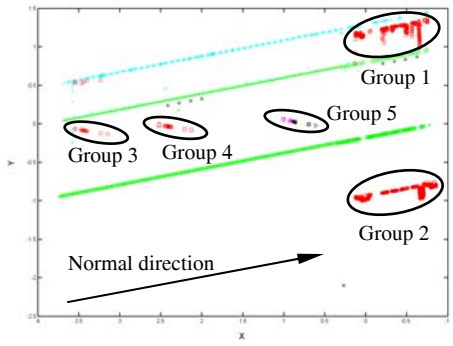

Fig. 1.c CMLHL projection.

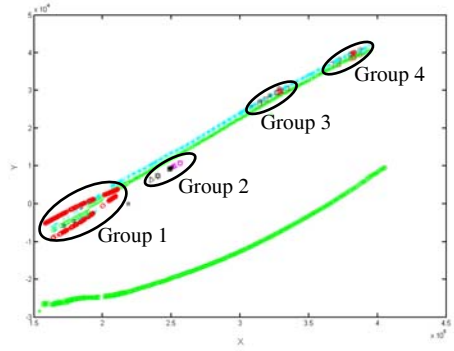

Fig. 1.d CCA (Euclidean dist.) projection.

It can be seen in Fig. 1.c how the CMLHL model is able to identify the two anomalous situations contained in the data set. As in the case of MLHL, the MIB information transfer (Groups 1 and 2 in Fig. 1.a) is identified due to its orthogonal

direction with respect to the normal traffic and to the high density of packets. The sweeps (Groups 3, 4 and 5 in Fig. 1.a) are identified due to their non-parallel direction to the normal one.

Several experiments were conducted to apply CCA to the analysed data set; tuning the different options and parameters, such as type of initialization, epochs and distance criterion. The best (from a projection point of view) CCA result, based on the Standardized Euclidean Distance, is depicted on Fig. 2. There's a marked contrast between the behavioral pattern shown by the normal traffic in previous projections and the evolution of normal traffic in the CCA projection. In the latter, some of the packets belonging to normal traffic do not evolve in parallel straight lines. That is the case of groups 1 and 2 in Fig. 2. The anomalous traffic shows an abnormal evolution once again (Groups 3 and 4 in Fig. 2), so it is not as clear as in previous projections to distinguish the anomalous traffic from the "normal" one.

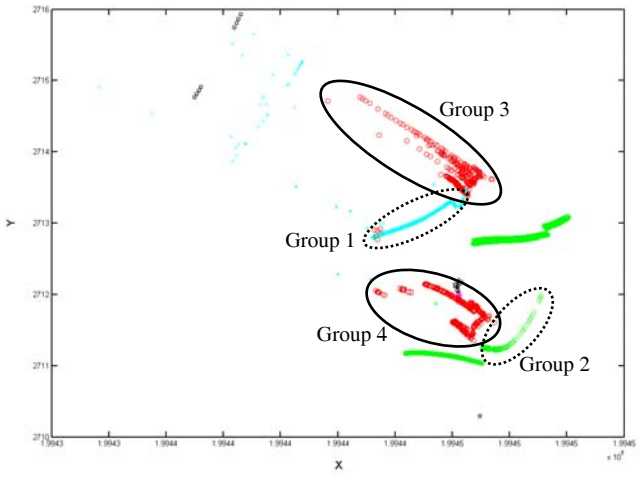

Fig. 2 CCA projection (employing Standardized Euclidean distance).

For comparison purposes, a different CCA projection for the same dataset is shown on Fig. 1.d. This projection is based on simple Euclidean Distance. The anomalous situations can not be identified in this case as the evolution of the normal and anomalous traffic is similar. Different distance criteria such as Cityblock, Humming and some others were tested as well. None of them surpass the Standardized Euclidean Distance, whose projection is shown on Fig. 2.

5 Conclusions and Future Work

The use of embedded ANN in the deliberative agents of a dynamic MAS let us take advantage of some of the properties of ANN (such as generalization) and agents (reactivity, proactivity and sociability) making the ID task possible. It is worth mentioning that, as in other application fields, the tuning of the different neural models is of extreme importance. Although the neural model can get a useful

projection, a wrong tuning of the model can lead to a useless outcome, as is the case of Fig 1.d.

We can conclude as well that CMLHL outperforms MLHL, PCA and CCA. This probes the intrinsic robustness of CMLHL, which is able to properly respond to a complex data set that includes time as a variable.

Further work will focus on the application of high-performance computing clusters. Increased system power will be used to enable the IDS to process and display the traffic data in real time.

Acknowledgments. This research has been partially supported by the project BU006A08 of the JCyL.

References

1. Chuvakin, A.: Monitoring IDS. Information Security Journal: A Global Perspective 12(6), 12 - 16 (2004)
2. Wooldridge, M., Jennings, N., R.: Agent theories, architectures, and languages: A survey. Intelligent Agents (1995)
3. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. Computer Networks: The Int. Journal of Computer and Telecommunications Networking 34(4), 547-570 (2000)
4. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. IEEE Potentials 22(4), 28-31 (2003)
5. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsettii, R.: CIDS: An agent-based intrusion detection system. Computers & Security 24(5), 387-398 (2005)
6. Wang, H.Q., Wang, Z.Q., Zhao, Q., Wang, G.F., Zheng, R.J., Liu, D.X.: Mobile Agents for Network Intrusion Resistance. In: APWeb 2006. LNCS, vol. 3842, pp. 965-970. Springer, Heidelberg (2006)
7. Deeter, K., Singh, K., Wilson, S., Filipozzi, L., Vuong, S.: APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System. In: Mobility Aware Technologies and Applications. LNCS, vol. 3284, pp. 244-253. Springer, Heidelberg (2004)
8. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised? In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 50-57. Springer, Heidelberg (2005)
9. Liao, Y.H., Vemuri, V.R.: Use of K-Nearest Neighbor Classifier for Intrusion Detection. Computers & Security 21(5), 439-448 (2002)
10. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonenen Net for Anomaly Detection in Network Security. IEEE Transactions on Systems Man and Cybernetics, Part B 35(2), 302-312 (2005)
11. Corchado, E., Herrero, A., Sáiz, J.M.: Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 905-910. Springer, Heidelberg (2005)
12. Middlemiss, M., Dick, G.: Feature Selection of Intrusion Detection Data Using a Hybrid Genetic Algorithm/KNN Approach. In: Design and Application of Hybrid Intelligent Systems. IOS Press. 519-527 (2003)
13. Kholfi, S., Habib, M., Aljahdali, S.: Best Hybrid Classifiers for Intrusion Detection. Journal of Computational Methods in Science and Engineering 6(2), 299 - 307 (2006)

14. Herrero, Á., Corchado, E., Pellicer, M., Abraham, A.: Hybrid Multi Agent-Neural Network Intrusion Detection with Mobile Visualization. In: Innovations in Hybrid Intelligent Systems. Advances in Soft Computing, vol. 44, pp. 320-328. Springer, Heidelberg (2007)
15. Corchado, J.M., Laza, R.: Constructing Deliberative Agents with Case-Based Reasoning Technology. International Journal of Intelligent Systems 18(12), 1227-1241 (2003)
16. Pellicer, M.A., Corchado, J.M.: Development of CBR-BDI Agents. International Journal of Computer Science and Applications 2(1), 25 - 32 (2005)
17. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39-59 (1994)
18. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A. (1987)
19. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: the Gaia Methodology. ACM Transactions on Software Engineering and Methodology 12(3), 317-370 (2003)
20. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine 2(6), 559-572 (1901)
21. Demartines, P., Herault, J.: Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets. IEEE Transactions on Neural Networks 8(1), 148-154 (1997)
22. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203-225 (2004)
23. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. Int. Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447-1466 (2003)
24. Oja, E.: A Simplified Neuron Model as a Principal Component Analyzer. Journal of Mathematical Biology 15(3), 267-273 (1982)
25. Sanger, D.: Contribution Analysis: a Technique for Assigning Responsibilities to Hidden Units in Connectionist Networks. Connection Science 1(2), 115-138 (1989)
26. Fyfe, C.: A Neural Network for PCA and Beyond. Neural Processing Letters 6(1-2), 33-41 (1997)
27. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE 78(9), 1464-1480 (1990)
28. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers 23(9), 881-890 (1974)
29. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems 10, 350-356 (1998)
30. Cisco Secure Consulting. Vulnerability Statistics Report. (2000)