Auto-Associative Neural Techniques for Intrusion Detection Systems

Álvaro Herrero, Emilio Corchado
Department of Civil Engineering
University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
Email: {ahcosio, escorchado}@ubu.es

Paolo Gastaldo, Francesco Picasso, Rodolfo Zunino
Dept. of Biophysical and Electronic Engineering (DIBE)
Genoa University
Via Opera Pia 11a, 16145 Genoa, Italy
Email: {paolo.gastaldo, fpi, rodolfo.zunino}@unige.it

***Abstract*— Intrusion Detection Systems (IDS's) ensure the security of computer networks by monitoring traffic and generating alerts, or taking actions, when suspicious activities are detected. This paper proposes a network-based IDS supporting an intuitive visualization of the time evolution of network traffic. The system is designed to assist the network manager in detecting anomalies, and exploits Auto-Associative Back-Propagation (AABP) neural networks to turn raw data extracted from traffic sources into an intuitive 2-D representation. The neural component operates as a sort of smart compression operator and supports a compact representation of multi-dimensional data. The empirical verification of the mapping method involved the detection of anomalies in traffic ascribed to the Simple Network Management Protocol (SNMP), and confirmed the validity of the proposed approach.**

I. INTRODUCTION

Intrusion Detection Systems (IDS's) monitor the traffic in computer networks and generate alerts, or trigger defensive actions, when suspect activities are detected. As a consequence of the degenerating scenario of open traffic, IDS's have become common elements in modern infrastructures to enforce network policies, yet some scientific issues remain open in IDS's development and run-time operation. IDS technologies typically embed two approaches [1]: misuse intrusion detection (MID) and anomaly intrusion detection (AID).

MID systems recognize known attack patterns, and typically rely on a knowledge base of rules to discriminate normal from malicious traffic. MID is today's state of the art in network security but suffers from basic drawbacks: first, the set of rules is susceptible to inconsistencies; secondly, continuous updating is required to incorporate unseen attack patterns.

AID systems model 'normal' traffic patterns and generate alerts when 'abnormal' events are detected. These techniques do not embed sets of rules and can support time-zero detection of novel attack strategies. On the other hand, the anomaly-based approach requires consistent modeling of normal traffic. Accuracy in detection proves indeed the major limitation of AID systems, which can exhibit a relatively high rate of false positives [1]. The typical approach to circumvent that issue is to drive the IDS's development empirically, and connectionist models can be profitably used for that purpose. Supervised methods [1-7] tackle intrusion detection as a binary classification problem (i.e., normal vs./ abnormal traffic) and attain quite accurate results. However, the need for data labeling in the set-up phase and the continuous evolution of attack types often lead to a very expensive training process.

Unsupervised methods for anomaly detection [1, 8, 9] first extract features from traffic data and then apply unlabelled learning methods. In general, the ultimate goal is to identify the significant portions of the feature space that support the distribution of normal traffic, whereas outliers will mark abnormal traffic activities. However, unsupervised techniques have also been profitably applied to IDS's supporting visual analysis of the network traffic [10]. Unsurprisingly, supervised methods outperform unsupervised approaches at identifying known patterns [1]; by contrast, the latter ones prove more robust when coping with unknown attacks in a dynamic scenario, and therefore have been chosen as the scientific baseline for the present research.

In particular, Auto-Associative Back-Propagation (AABP) neural networks [11] support the design of IDS's giving a 2-D visualization of the time evolution of the network traffic. The system assists a network manager in detecting anomalies and performs two tasks: 1) the analysis of the network traffic and 2) a synthetic visualization of the traffic analysis on a 2-D display, which provides a convenient interface. The AABP neural network operates as 'smart compression' operator, and supports the crucial task of mapping raw data extracted from traffic sources into an intuitive visual representation.

During training, the neural network is supplied with a set of *n*-dimensional vectors, which are associated with (unlabeled) network traffic and hold feature values extracted from the packets. The neural network learns to compress the data irrespectively of the actual nature (malicious or not) of the traffic situation. At run-time, the IDS feeds the neural component with the same vector representation, and derives an effective, two-dimensional, representation of data, in which abnormal and potentially malicious situations are quite evident.

The experimental domain adopted to verify the method's effectiveness involves traffic ascribed to the Simple Network Management Protocol (SNMP), which represents one of the top 5 most vulnerable services [12]. Empirical tests involved a dataset previously used in literature for unsupervised analysis [10], and proved the effectiveness of the proposed approach.

II. THE VISUAL INSPECTION OF TRAFFIC IN MODERN IDS'S

The network-based IDS described in this paper is organized as follows (Fig. 1):

- packets traveling through the network are intercepted by some *network capture* device;
- monitored traffic is represented by a set of numerical features spanning a multidimensional vector space;
- the actual neural component operates on these feature vectors and yields a two-dimensional representation of the network traffic;
- the outcomes of the neural module are presented to the network manager in a *traffic display* device.

The compression neural component processes an *n*-dimensional vector that has been previously assembled by a "packet processing" module, which extracts numerical features associated with each network packet. Hence, the proposed IDS operates at the packet level and not at the connection level as other models do [13, 14]. Indeed, the design of the feature set is a crucial issue that has been thoroughly addressed in the literature [15]. It has been proved that timestamp, source and address port, and protocol uniquely identify a connection [16]. When dealing with Transmission Control Protocol (TCP) traffic, additional features may be required (e.g. to track connection state [17]); instead, User Datagram Protocol (UDP) traffic can be effectively characterized by a reduced feature set [10].

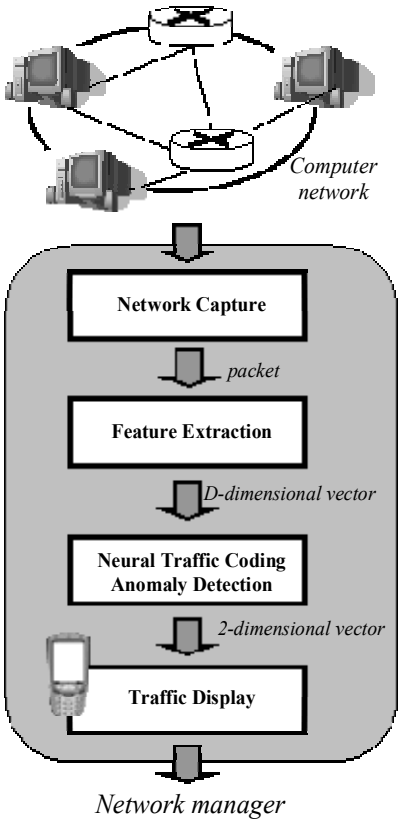

Fig. 1. Structural schema of the neural-based anomaly detection and IDS functioning.

The neural component clearly is the actual core of the overall IDS. That module is designed to yield an effective, two-dimensional representation of network traffic, thus providing a powerful tool for further visual inspection. As a consequence, the effectiveness of the overall approach based on the two-dimensional representation of traffic strictly relates to the successful support to the network supervisor at detecting traffic anomalies.

A connectionist approach appears consistent with the anomaly-detection problem setting mainly because it allows a system to empirically learn the input-output relationship between raw traffic and subsequent interpretation. The crucial advantage is that the outlier-detection method does not require any *a-priori* analytical formulation of the underlying phenomenon.

In principle, any unsupervised method applies to the involved representation process, and indeed Self-Organizing Maps [7, 8] and Vector Quantization-based methods [9] have had a considerable success in supporting IDS technology. As compared with those models, Auto-Associative Back-Propagation (AABP) neural networks represent an intriguing alternative for unsupervised learning, especially when considering its non-linear formulation [11].

From a structural viewpoint, the approach proposed in this paper for Anomaly Intrusion Detection exploits the AABP capability to implement universal nonlinear approximation. The following section describes the AABP neural network and the design strategy for the neural-based AID module.

III. AUTO-ASSOCIATIVE BACK-PROPAGATION NETWORKS FOR DIMENSIONALITY REDUCTION

A. Back-Propagation Networks

A neural network-based device can be viewed as a mapping box configured by a set of parameters ('weights'), which should be adjusted so as to reproduce a given input-output relationship as accurately as possible. The crucial advantage of neural approaches is that the weight values can be learned empirically; hence the mapping tool does not need any a-priori analytical formulation of the observed phenomenon.

MultiLayer Perceptrons (MLPs) [18] support the mapping by arranging several nonlinear units ('neurons') into a layered structure. Each neuron transforms its own weighted inputs by a sigmoidal function $\sigma(r)$, whose nonlinearity is crucial because theory proves that such networks can support arbitrary mappings [19, 20]. A traditional MLP includes three layers (input, 'hidden', output), and associates an input vector, $\boldsymbol{x} \in \Re^D$, with an output vector, $\boldsymbol{y} \in \Re^Q$, computed as:

$$y_q(\mathbf{x}) = w'_{q,0} + \sum_{u=1}^{N_h}\left[w'_{u,q} \cdot \sigma\left(w_{u,0} + \sum_{k=1}^{D} w_{u,k} x_k \right)\right]; q = 1,...,Q \qquad (1)$$

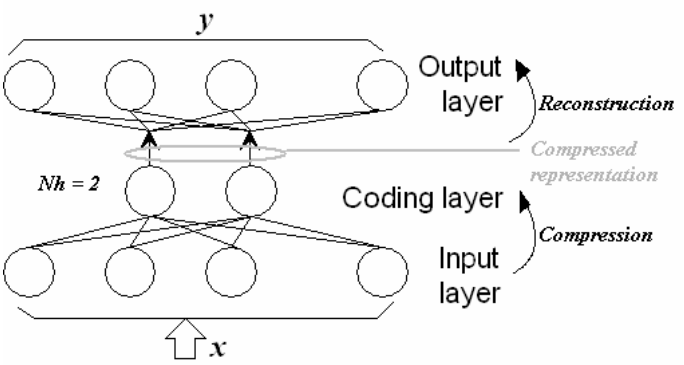

Fig.2. A three-layer Auto-Associative Back-Propagation network supports a lossy compression of input data.

where N_h is the depth of the sigmoid series expansion, and the coefficients W=$\{\boldsymbol{w}, \boldsymbol{w'}\}$ are the weights for the interconnections between the two upper layers (Fig. 2).

Those weights, W, are adjusted empirically so that the network best reproduces the desired ($\boldsymbol{x}$, $\boldsymbol{y}$) mapping over a given training set. The classical cost function measuring the mapping distortion is the mean square error, E_W, between the desired response ('target'), for a given input vector and the actual network output. Thus, the network-training process is formulated as an optimization problem expressed as

$$\min_{W} E_W = \min_{W} \frac{1}{n} \sum_{s=1}^{n} \left\| \mathbf{t}^{(s)} - \mathbf{y}\left(\mathbf{x}^{(s)}\right) \right\|^2 \qquad (2)$$

where $\boldsymbol{t}^{(s)}$ is the desired output for the s-th training vector, $\boldsymbol{x}^{(s)}$, and n is the number of training pairs ($\boldsymbol{x}^{(s)}$, $\boldsymbol{t}^{(s)}$). The Back-Propagation (BP) algorithm [18] is a powerful tool for training (2), and its effect on the practical impact of MLPs has been so vast that MLPs are often called 'Back-Propagation' networks. BP tackles the learning problem (2) by a stochastic gradient-descent strategy over the weight space.

B. Auto-Associative Back-Propagation Networks

Auto-Associative BP networks constitute an unsupervised variant of the general model, in which the desired outputs coincide with the network inputs: $\mathbf{t} \equiv \boldsymbol{x}$ (Fig. 2). Forcing the network to replicate the training sample distribution mainly aims at a reduction in dimensionality, since the hidden layer is typically smaller that the input/output ones.

At run-time, an AABP network is used to associate the coding values computed by the hidden neurons with each input vector; these 'mapping outputs' actually support the (lossy) transformation from the input space into a lower-dimensional representation. Theory proved [11] that a three-layer AABP network supports a mapping that is affine (if not equivalent) to Principal Component Analysis (PCA) [21, 22].

Quite in view of this equivalence, it clearly appears that such a compression mapping might eventually suffer from the same drawbacks affecting PCA-like representations, the most prominent of which is a considerable sensitivity to the presence

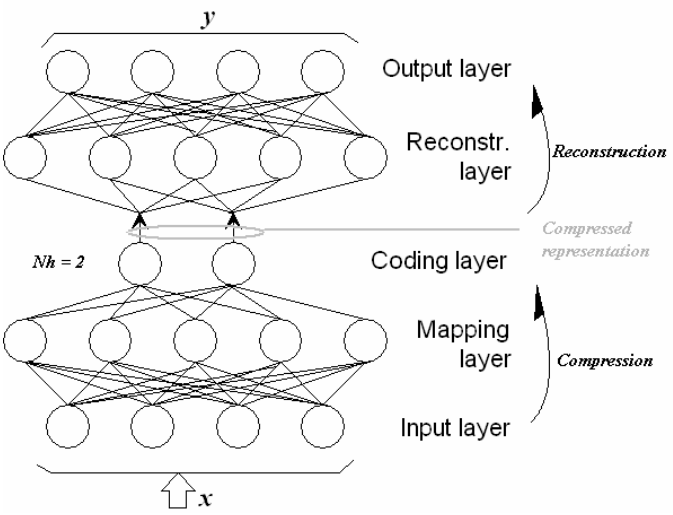

Fig. 3. A NonLinear Auto-Associative Back-Propagation network includes five layers to reduce data dimensionality.

of outliers in the training set. It is indeed known that the vector eigenvectors induced by linear mappings may suffer from poor consistency when the PCA-based training includes abnormal data points [23].

That consideration, together with the ability of universal approximation theoretically ascribed to the Back-Propagation model [19], lead to a more sophisticated model of AABP networks, which anyway adhere to the basic principle of unsupervised training. The output layer still remaps the input vector values, and a hidden compression layer still supports a dimensionality reduction. The basic difference is in the compression/reconstruction sections, as both include an additional layer of neurons, thereby leading to a five-layer Auto-Associative network (Fig. 3).

The mapping supported by such an architecture was called NonLinear Principal Component Analysis (NLPCA) [11].

The run-time use of the resulting networks, after training completion, is totally equivalent to the use of three-layer AABP networks: the output values of the coding layer ('mapping outputs') provide the low-dimensional representation of each input vector.

The increase in representation power (and complexity) conveyed by the NLPCA augmentation is remarkable. The lower half of the network, also called the 'compression section', actually embeds a complete three-layer BP network, and therefore benefits from the universal capabilities predicted by theory [18]. The critical issue, of course, is that no one could ever know in advance the N_h target values that should be imposed for a conventional training process. The trick involved with the NLPCA approach is that those target values can be implicitly imposed by forcing the network to reconstruct the original sample in the upper section. Thus the 'reconstruction' section is symmetrical with respect to the compression section, in order to yield equivalent, universal (inverse) mapping capabilities.

The main advantage is that the ultimate compressed representation is no longer linked to any linear underlying model, but stems instead from an universal internal representation, that is learned empirically. On the other hand, the complexity of the augmented model is apparent, and the weight-tuning process might turn out to be quite difficult due to the much larger number of free parameters. This possibly gives rise to presence of local minima, especially in the presence of limited training sets, and optimized learning algorithms are often applied to tame training complexity [24].

In summary, NLPCA techniques seem to fit those domains for which 1) a nonlinear representation is required to best encompass the observed empirical phenomenon, and at the same time, 2) a considerable number of empirical samples is available.

IV. Using AABP Networks for Anomaly Detection

A. Simple Network Management Protocol

The present IDS is targeted to detect traffic anomalies within the Simple Network Management Protocol (SNMP), which is a part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. SNMP is an application layer protocol that supports the exchange of management information between network devices. This protocol enables network administrators to manage network performance and is used to control network elements such as routers, bridges and switches. This property makes SNMP data quite sensitive and liable to potential attacks. Indeed, an attack based on the SNMP protocol may severely compromise system security, as reported by CISCO [12].

The method presented in this paper addresses two different types of attacks that rely on the SNMP protocol:

- MIB information transfer: the Management Information Base (MIB) is a collection of information concerning a managed device, including sensitive data such as network and router information (e.g. IP, Mac address and VLan configuration). As specified by the Internet Activities Board (IAB), the SNMP protocol is used to access MIB objects; thus, protecting a network from malicious MIB information transfer is crucial.
- SNMP port sweep: a port scan (or sweep) is an attempt to count the services running on a machine by probing each port for a response. In this paper, the SNMP port scanning is tackled.

B. Feature extraction

The eventual network-based IDS for the detection of SNMP anomalous traffic is structured as shown in Fig. 1. The "packet processing" component (see Sec. II) generates feature vectors by working out information contained in the packet header. In the present research, network packets are characterized by using the set of features that already proved to be effective for detection of anomalous SNMP traffic [10]. The set of four features that are extracted from packets contribute to build up the neural-network input vector, $\mathbf{x} \in \Re^4$; these features can be listed as follows:

- *Protocol ID*: an integer number that identifies the protocol of the packet;
- *Source port*: the port number of the device that sent the packet;
- *Destination port*: the port number of the target host, i.e. the host to which the packet is sent;
- *Size*: the packet size (in Bytes).

As such, at the output of the "packet processing" module the network traffic is mapped in a four-dimensional feature space.

According to the set-up discussed in Sec. III, the AID module exploits an AABP neural network to generate a two-dimensional representation of the network traffic by starting from the four-dimensional space defined by the feature set. Thus, first an offline training phase uses empirical data to set the configuration of weight quantities for the neural network. Then, the eventual neural system is used to process the feature vectors generated at run-time and to feed the visual display.

Fig. 4 gives an outline of the run-time operation of the neural-based IDS.

V. Experimental Results

The proposed network-based IDS has been tested by using the data set utilized in [10]. The data set contained network packets captured from UDP (i.e. User Datagram Protocol) traffic, as SNMP uses UDP as the transport protocol for passing data between managers and agents. Hence, the data set included only packets that use UDP as transport layer and IP as network layer. A total of 5866 samples (i.e. network packets)

The IDS run-time operation algorithm

```
0.  (Initialization)
    Inputs:  neural network weights, W
    Time slot for visual update rate
1.  For each time slot:
    a. Scan network traffic and acquire
       packets
    b. Extract numerical features
    c. Associate with each packet datum a
       four-dimensional feature vector, x
    d. Feed the AABP network with vector x
    e. Register the two-dimensional mapping
       vector, v, spanned by the middle-layer
       neurons
    f. Feed the visual interface to the
       network manager with vector v.
```

Fig. 4. The run-time operation algorithm of the neural-network based IDS.

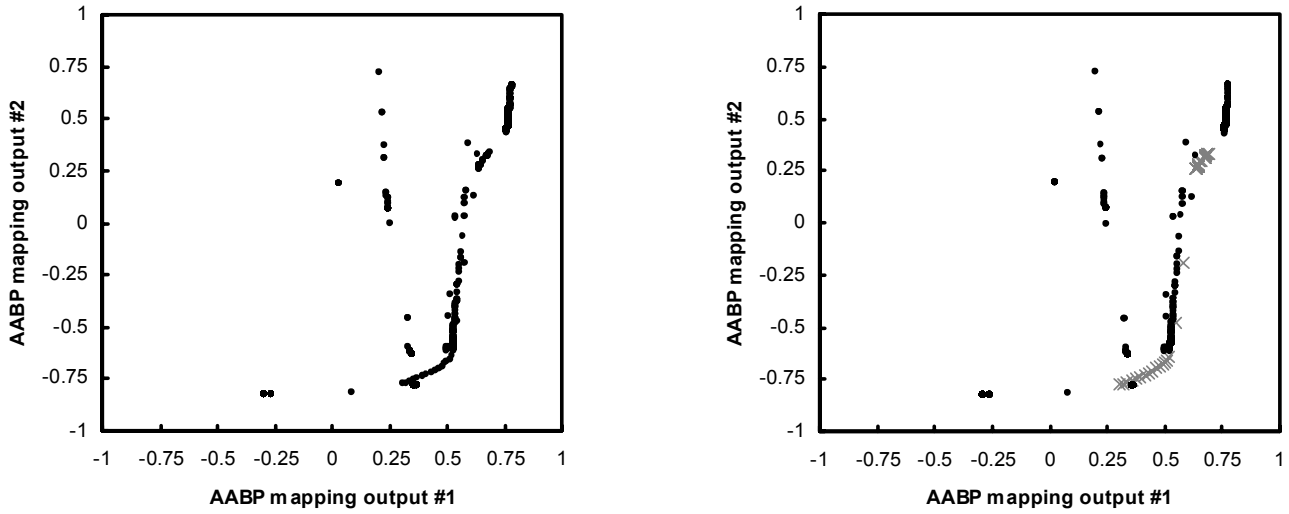

Fig. 5. AABP compression supports the accurate detection of traffic anomalies. Normal traffic evolves along parallel lines while abnormal traffic diverges from the normal direction: a) Unsupervised representation of traffic progress; b) Traffic representation where malicious traffic is highlighted.

were involved in the experiment.

As anticipated in Sec IV, network packets were characterized by a feature set (i.e., *Protocol ID*, *Source Port*, *Destination Port* and *Size*) spanning a four-dimensional space. Thus, the eventual AABP-based IDS was trained to map such a four-dimensional space into a two-dimensional space for an intuitive visualization of the traffic progress.

In the experiments presented here, the configuration of the AABP network included a number of 30 nodes in the hidden layers (coding and reconstruction), while of course the number of neurons in the middle layer was N_h=2. Although theoretical studies did not succeed in providing any established design criterion to set the number of a network's hidden nodes, the literature provides both analytical [25] and practical criteria [26] for dimensioning a network size, in order to ensure prediction accuracy while minimizing the risk of overfitting training data. The present research adopted a well-known practical approach [26] mainly because of its simplicity and proved effectiveness.

In summary, the architecture of the overall AABP network was set as follows: four nodes in the input layer, 30 nodes in the compression layer, 2 nodes in the coding layer, 30 nodes in the decompression layer, and four nodes in the output layer.

Fig. 5 presents the results obtained by the AABP mapping; the graph in Fig. 5-a) plots the network packets by using as coordinates the two outputs of the coding layer, i.e. the compressed representation of the inputs, which therefore is the actual visual rendering of the original four-dimensional representation of the network traffic. One easily notes that the data pattern exhibits two anomalies, which are quite apparent because most of the data are organized according to almost parallel patterns, whereas two different, smaller groups of data evolve in a different direction.

To allow a correct interpretation of these results, Fig. 5-b) augments the displayed information by associating different markers with the actual packet nature: the data points marked by gray crosses relate to anomalous traffic. This graph points out that the two groups of abnormal data, which the previous (unsupervised) analysis highlighted due to the odd direction of progression, did in fact involve packets that should be classified as anomalous traffic. This confirmed that the bi-dimensional mapping process obtained by using the AABP network, adjusted empirically during the training process, succeeded in identifying anomalous traffic.

Fig. 6 shows indeed that with the present testbed the AABP-based approach outperforms conventional PCA. The figure illustrates the result obtained by mapping into the two-dimensional space provided by PCA the same dataset used in the experiment described above.

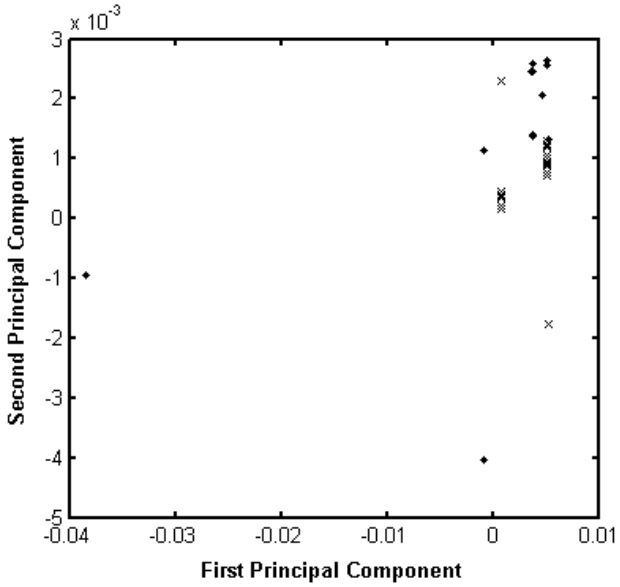

Fig. 6. PCA compression does not support the detection of traffic anomalies.

The graph clearly points out that PCA does not attain a satisfactory representation of the network traffic, as all the data is depicted in the same way. PCA is not able to differentiate the anomalous traffic from the normal one. The used dataset is a representative and complete sample as it includes both normal and anomalous traffic. The anomalous situations contained in the dataset are characteristic of SNMP-related intrusive actions.

VI. Conclusions

This paper presents a network-based IDS supporting a powerful 2-D visualization of network traffic. The IDS has been designed to support the network manager in detecting traffic anomalies by embedding a synthetic visualization of the traffic analysis on a 2-D display.

The proposed method exploits a connectionist approach to tackle the crucial issue of the effective representation of network traffic on a two-dimensional domain. The major result of the present research lies in showing that AABP neural networks can represent a valuable tool for addressing such a task. Indeed, two important aspects make the AABP-based approach interesting: 1) the set up of AID model follows an unsupervised paradigm, and 2) the AABP network can implement universal nonlinear approximation.

Acknowledgment

This research has been partially supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

[1] P. Laskov, P. Dussel, C. Schafer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in *Proc. Int. Conf. on Image Analysis and Processing*, *ICIAP 2005*, pp. 50-57, 2005.

[2] R.A. Maxion and K.M.C. Tan, "Benchmarking anomaly-based detection systems," in *Proc. 1st Conf. on Dependable Systems and Networks,* New York, pp. 623-630, 2000.

[3] S. Mukkamala, G.I. Janoski, and A.H. Sung, "Intrusion detection using support vector machines," in *Proc. High Performance Computing Symposium, HPC 2002*, pp. 178-183, 2002.

[4] Y. Qiao, X.W. Xin, Y. Bin, and S. Ge, "Anomaly intrusion detection method based on HMM," *Elect. Letters*, vol. 38, no. 13, pp. 663-664, June 2002.

[5] Y. Liao and V.R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Comput. Security*, vol. 21, no. 5, pp. 439-448, October 2002.

[6] L.P. Cordella, A. Limongiello, and C. Sansone, "Network intrusion detection by a multi-stage classification system," in *Proc. Multiple Classifier Systems: 5th Int. Workshop*, Cagliari, Italy, pp. 324-333, 2004.

[7] S.T. Sarasamma, A.Z. Qiuming, and J. Huff, "Hierarchical Kohonen net for anomaly detection in network security," *IEEE Trans. on SMC – part B: cybernetics*, vol. 35, no. 2, April 2005.

[8] S. Zanero, "Analyzing TCP traffic patterns using self organizing maps," in *Proc. Int. Conf. on Image Analysis and Processing*, *ICIAP 2005*, pp. 83-90, 2005.

[9] J. Zheng, and M. Hu, "An anomaly intrusion detection system based on vector quantization," *ICIE Trans. on Inf. & Syst.*, vol. E89-D, no. 1, January 2006.

[10] E. Corchado, A. Herrero, and J.M. Sáiz, "Detecting compounded anomalous SNMP situations using unsupervised pattern recognition," in *Proc. Int. Conf. on Artificial Neural Networks*, ICANN 2005, Springer - LNCS, vol. 3697. pp. 905-910, 2005.

[11] M.A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2 February 1991.

[12] Cisco Secure Consulting: Vulnerability statistics report. 2000

[13] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, K. Das, "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation," in *Proc. 3rd International Workshop on Recent Advances in Intrusion Detection*, Springer - LNCS, vol. 1907. pp. 162 -182, 2000.

[14] M. Sabhnani and G. Serpen, "Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context," in *Proc. 2003 Int Conference on Machine Learning, Models, Technologies and Applications*, pp. 623–630, June 2003.

[15] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proc. 2001 IEEE Symp. on Security and Privacy*, pp. 130-143, May 2001.

[16] W. Lee, S. J. Stolfo, and K. W. Mok. "Mining in a data-flow environment: Experience in network intrusion detection," in *Proc. 5th International Conference on Knowledge Discovery and Data Mining (KDD-99)*, ACM, pp. 114–124, 1999.

[17] Lee, W., S. J. Stolfo, and K. W. Mok, "Adaptive intrusion detection: A data mining approach," *Artificial Intelligence Review*, vol. 14, no. 6, pp. 533–567, 2000.

[18] D. E. Rumelhart and J. L. McClelland, *Parallel distributed processing*. Cambridge, MA: MIT Press, 1986.

[19] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

[20] BB. G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math. Control, Signals, Systems*, vol. 2, 303–314, 1989.

[21] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559-572, 1901.

[22] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Education Psychology*, vol. 24, pp. 417-444, 1933.

[23] B. Gabrys, B. Baruque, and E. Corchado, "Outlier resistant PCA ensembles," *in Proc. 10th Int. Conf. on Knowledge-Based & Intelligent Information & Engineering Systems*, Springer – LNCS, vol. 4253, pp. 432-440, 2006.

[24] D. Anguita, GC. Parodi, and R. Zunino, "An efficient implementation of BP on RISC-based workstations," *Neurocomputing*, vol. 6, pp.57-65, 1994.

[25] E. B. Baum and H. David, "What size net gives valid generalization?," *Neural Computation*, vol. 1, no. 1, pp. 151-60, 1989.

[26] B. Widrow and M.A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline and Backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415-1442, 1990.