# Computational-Intelligence Models for Visualization-based Intrusion Detection Systems

Paolo Gastaldo[1], Francesco Picasso[1], Rodolfo Zunino[1], Emilio Corchado[2] and Alvaro Herrero[2]

[1]Dept. of Biophysical and Electronic Engineering (DIBE), Genoa University
Via Opera Pia 11a, 16145 Genoa, Italy
{paolo.gastaldo, francesco.picasso, rodolfo.zunino} @unige.it

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

**Abstract.** Intrusion Detection Systems (IDS's) are essential components in a network communication infrastructure, as they enforce security by monitoring traffic and detecting malicious activities. In this research, Computational Intelligence models support an IDS technology to obtain a synthetic, effective visualization of the traffic analysis. Auto-Associative Back-Propagation (AABP) neural networks map feature vectors extracted from traffic sources into a compact representation on a 2-D display. During training, the neural network learns to compress the data in an unsupervised fashion; at run time, the trained neural component synthesizes an effective, 2-D representation of the traffic situation. Empirical tests involving Simple Network Management Protocol (SNMP) traffic proved the validity of the approach.

**Keywords**: network security, intrusion detection system, Auto-Associative Back-Propagation Networks, Vector Quantization

## 1 Introduction

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. IDS's nowadays are standard components in networked infrastructures, as they effectively support administrators in detecting attacks and policy violations. Two basic approaches exist toward that end [1]: misuse intrusion detection (MID) and anomaly intrusion detection (AID). The former typically rely on a knowledge base of rules to discriminate normal from malicious traffic, and are designed to recognize known attack patterns. MID technology is today's state of the art but suffers from structural drawbacks: the set of rules is liable to inconsistencies, hence continuous updating is required to incorporate unseen attack patterns. On the other hand, AID systems tend to model 'normal' traffic patterns and generate alerts when they detect events that deviate from normal profiles. AID can support time-zero detection of

novel attack strategies but, to achieve satisfactory performances, the anomaly-based approach requires consistent modeling of normal traffic. Thus detection accuracy is a critical issue of AID systems that may result in a relatively high rate of false positives, and computational intelligence can tackle such a drawback [1].

A connectionist approach fits the anomaly-detection framework especially because it allows an IDS to develop empirically. In supervised methods [1-3], intrusion detection is tackled as a binary classification problem (i.e., normal vs. abnormal traffic). These methods 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 very expensive training. Unsupervised methods for anomaly detection [1, 4-6] extract features from traffic data and apply unlabelled learning methods. The goal is to identify, in the feature space, the significant regions that support the distribution of normal traffic; outliers mark abnormal activities. Unsurprisingly, supervised methods outperform unsupervised approaches at identifying known attacks [1]; by contrast, the latter ones prove more robust when coping with unknown patterns in a dynamic scenario, and therefore have been chosen as the scientific baseline for the present work.

The research presented in this paper adopts a slightly different approach, in which an IDS operates as an aiding tool to the network manager, and unsupervised methods serve to drive a compact visualization of the traffic evolution. The system assists the network manager in detecting anomalies by performing two tasks: 1) the analysis of network traffic, and 2) a synthetic visualization of the traffic progress on a 2-D display, which provides a convenient interface. Auto-Associative Back-Propagation (AABP) neural networks [7] are entrusted with the mapping of raw traffic data into the intuitive, visual format. During the training process, the neural network is supplied with a set of unlabeled feature vectors extracted from packets, and learns to compress these data irrespectively of the nature (normal/malicious) of the contingent traffic situation. At run time, the IDS feeds the trained network with the current feature vector and obtains a two-dimensional representation of data in which abnormal, potentially malicious situations become apparent.

The experimental campaign to verify the method's effectiveness addressed Simple Network Management Protocol (SNMP) traffic, mainly because SNMP is one of the most vulnerable services [8]. The tests involved a dataset previously used for unsupervised analysis [5] and proved the reliability of the proposed approach.

## 2 Visual Inspection of Traffic in Modern IDS's

The network-based IDS embeds three different modules. The *packet-processing* module maps the monitored traffic packets into a set of numerical features, $\phi$, spanning a multidimensional vector space, $\mathbf{f}$. The *AID* module compresses feature vectors into a two-dimensional rendering, $\mathbf{o}$, of the network traffic. Finally, the *visualization* module presents the analysis outcomes to the network manager in a traffic display device.

The AID module is the crucial core of the overall IDS. That component is fed with $n$-dimensional feature vectors, $\mathbf{f}_i$. These vectors are assembled by the packet-processing module, which associates numerical features with network packets. The AID output is a compact two-dimensional representation, $\mathbf{o}$, of network traffic, which

retains important information about the traffic progress and provides a powerful tool for further visual inspection. The overall visualization-based approach strictly relies on a useful support to the network manager, in facilitating the detection of traffic anomalies.

The computational-intelligence approach proposed in this paper is appealing because the input-output relationship between feature vectors and display representation can be learned empirically and does not need an *a-priori* analytical formulation of the observed domain. 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 [4] and Vector Quantization-based methods [6] have had a considerable success in supporting IDS technology. Auto-Associative Back-Propagation (AABP) neural networks represent an intriguing unsupervised alternative to those models, especially in its non-linear formulation [7]. In the framework proposed here, AABP neural networks operate as 'smart compression' tools and support the crucial task of mapping raw traffic data into a 2-D space.

## 3 AABP Neural Networks for Dimensionality Reduction

### 3.1 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 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) [9] support the mapping task by a set of nonlinear units ('neurons') arranged into a layered structure. Each neuron transforms its own (weighted) inputs by a sigmoidal function $\sigma(r)$; such a nonlinearity is crucial because sigmoidal networks can support arbitrary mappings [10]. A typical MLP includes three layers (input, 'hidden', output), and associates an input vector, $\mathbf{x}\in\Re^{D}$, with an output vector, $\mathbf{y}\in\Re^{Q}$, computed as:

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

where $N_h$ is the depth of the sigmoid series expansion, and the coefficients $W=\{\mathbf{w}, \mathbf{w}'\}$ are the weights for the interconnections between the two upper layers. Those weights, $W$, are adjusted empirically so that the network best reproduces the desired $(\mathbf{x}, \mathbf{y})$ 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

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

where $\mathbf{t}^{(s)}$ is the desired output for the *s*-th training vector, $\mathbf{x}^{(s)}$, and *n* is the number of training pairs ($\mathbf{x}^{(s)}$, $\mathbf{t}^{(s)}$). The Back-Propagation (BP) algorithm [9] is a powerful tool for training (2), hence MLPs are often called 'Back-Propagation' networks. BP tackles the learning problem (2) by a stochastic gradient descent over the weight space.

### 3.2 Auto-Associative Back-Propagation Networks

Auto-Associative BP networks constitute an unsupervised variant of the general MLP model, in which the desired outputs coincide with the network inputs: $\mathbf{t} \equiv \mathbf{x}$. Forcing the network to replicate the training sample distribution aims at a reduction in dimensionality, as the hidden layer is typically smaller that the input/output ones. At runtime, an AABP network associates each input vector with the 'coding' values computed by the hidden neurons; these mapping outputs support the (lossy) transformation from the input space into a lower-dimensional representation.

A three-layer AABP network implements a mapping that is affine to Principal Component Analysis (PCA) [7]. Quite in view of this equivalence, the resulting mapping might suffer from the same drawbacks affecting PCA-like representations, such as a remarkable sensitivity to outliers in the training set.

The NonLinear Principal Component Analysis (NLPCA) architecture (Fig. 1) involved a sophisticated model of AABP and was proposed to tackle that issue [7]. Alike conventional three-layer AABP, the output layer imposes the input values as targets and a hidden layer still supports a dimensionality reduction. The crucial difference from classical AABP lies in the compression/reconstruction sections, each including an additional layer of neurons and thus leading to a five-layer network.

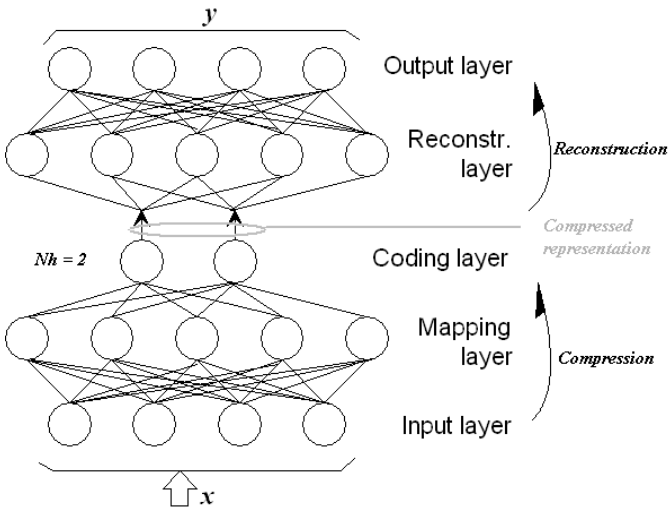



**Fig. 1.** A NonLinear Auto-Associative Back-Propagation network includes five layers.

The NLPCA architecture retains the universal approximation ability of Back-Propagation networks [10], and still adheres to the principle of unsupervised training. The run-time use of the resulting network, after training completion, is equivalent to the use of a three-layer AABP structure, as the mapping outputs of the middle 'coding' layer provide the low-dimensional representation of each input vector.

On the other hand, the increase in representation power conveyed by the NLPCA augmentation is remarkable. The lower half of the network, denoted as the 'compression section', actually embeds a complete three-layer BP network, and therefore benefits from the universal capabilities predicted by theory [10]. The problem, of course, is that no one knows in advance the $N_h$ target values that should be imposed to the lower section of the network for learning the compression task. The trick in the NLPCA approach is that those target values are imposed implicitly 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 compressed representation does not relate to any linear model (as was the case for PCA), but stems instead from a mostly general internal representation that is learned empirically. On the other hand, the complexity of the augmented model is apparent, and the weight-tuning process might require sophisticated training techniques due to the large number of free parameters. 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.

## 4 AABP-based IDS for Anomaly Detection in SNMP Traffic

### 4.1 SNMP Protocol

The present IDS technology considered 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 for the exchange of management information between devices. This protocol enables network administrators to drive network performance and is used to control network elements such as routers, bridges and switches. Hence, SNMP data are quite sensitive and liable to potential attacks [8]. Two different types of threats were considered, namely 1) SNMP port scanning (attempts to locate open ports for SNMP service on a machine), and 2) MIB information transfer. The Management Information Base (MIB) is a collection of information about a managed device, including sensitive data such as network and router configurations. SNMP accesses MIB objects, hence protecting a network from malicious MIB information transfer is crucial.

### 4.2 Feature Extraction

The eventual network-based IDS for the detection of SNMP anomalous traffic is structured as shown in Fig. 2. The "packet processing" component generates feature vectors $\mathbf{f}_i$ by working out information contained in the packet header. Then, the AID module exploits AABP neural network to generate a two-dimensional representation, $\mathbf{o}$, of the network traffic by starting from the $n$-dimensional space defined by the feature set $\phi$. Thus, first an offline training phase uses empirical data to set the configuration of weight quantities for the AABP. Then, the eventual neural system is used to process the feature vectors generated at run-time and to feed the visual display.

The design of the feature set $\phi$ is indeed a crucial issue that has been thoroughly addressed in the literature [11]. It has been proved that timestamp, source and address port, destination and address port, and protocol uniquely identify a connection [11]. When dealing with Transmission Control Protocol (TCP) traffic, additional features may be required (e.g. to track connection state [11]); instead, User Datagram Protocol (UDP) traffic can be effectively characterized by a reduced feature set [5].

In the present research, networks packet are characterized by using the set of features that already proved to be effective for detection of anomalous SNMP traffic [5]. The set of four features that are extracted from packets contribute to build up the neural-network input vector, $\mathbf{f} \in \Re^4$; 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.

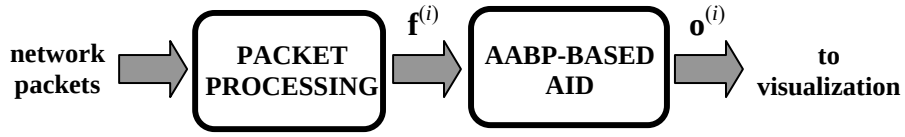



**Fig. 2.** Structural schema of the neural-based anomaly detection and IDS functioning.

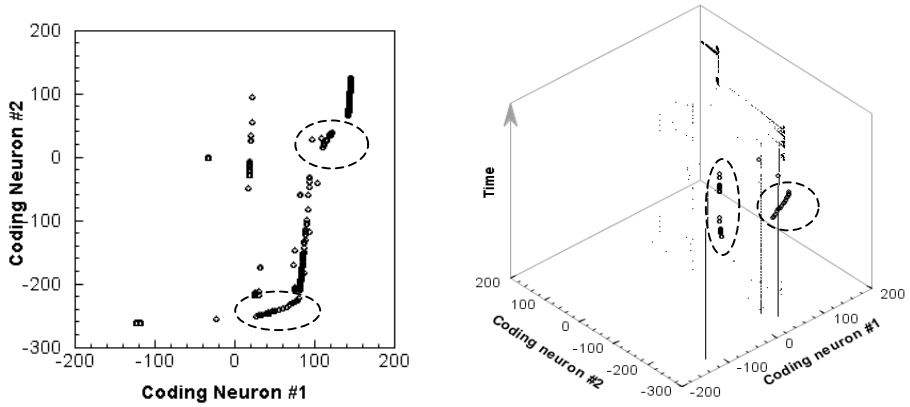
## 5 Experimental Results

The proposed AABP-based IDS has been tested on the data set used in [5]. Since SNMP relies on User Datagram Protocol (UDP) as a transport protocol, the data sample contained network packets captured from UDP traffic as transport layer and IP as network layer. A total of 5866 patterns were involved in the experiment; anomalous traffic of the type discussed in Sec. 4 covered a share of 1% of the whole dataset. Network packets were characterized by a four-dimensional feature set (i.e., Protocol

ID, Source Port, Destination Port and Size). Thus, the AID module was trained to map such a four-dimensional space into a two-dimensional space for an intuitive visualization of the traffic progress.

MLP theory does not provide any established design criterion to dimension the hidden layers, hence the present research adopted a practical approach [12], mainly thanks to its simplicity and proved effectiveness. Thus, in the experiments, the AABP configuration included 30 nodes in the hidden layers (coding and reconstruction); the number of coding neurons in the middle layer clearly was $N_h = 2$. Thus the overall AABP networks exhibited a (4 – 30 – 2 – 30 – 4) layered topology.

Figure 3 a) presents the results obtained by the middle-layer AABP mapping. The coordinate axes give the outputs of the two neurons in the coding layer, while each marker denotes a network packet. The compressed representation of the input vectors rendered the original network traffic visually; the graph highlighted two quite apparent anomalies, since most of the data were represented along almost vertical patterns whereas two smaller groups of data followed a different direction. The unsupervised representation clearly did not give any indication about the nature of those traffic patterns, but correctly pointed out them as anomalous situations.




**a)** **b)**

**Fig. 3.** AABP compression performance on SNMP traffic.

To verify the mapping of the obtained results, Figure 3b) augments the above 2-D visual representation by including time information, and by associating different markers with the actual packet nature: normal dots indicate normal traffic, whereas circles denote anomalous traffic. Such information was introduced a posteriori for post-analysis, and never entered either the AABP training or the IDS run-time operation. The graph confirmed that the two groups of abnormal data highlighted in the unsupervised analysis did in fact comprise packets that were to be classified as malicious traffic, and the timing progression of the visual display clearly demonstrates the effectiveness in outlier detection.

## 6 Conclusions

The paper presented 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 exploited 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 rep-resent a valuable tool for addressing such 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.

## References


1. Laskov, P., Dussel, P., Schafer, C., and Rieck, K.: Learning intrusion detection: supervised or unsupervised?. Proc. ICIAP 2005, Cagliari, Italy, 50-57
2. Liao, Y., and Rao Vemuri, V.: Use of k-nearest neighbor classifier for intrusion detection. Comput. Security 21(5) (2002) 439-448
3. Sarasamma, S.T., Qiuming, A.Z. , and Huff, J.: Hierarchical Kohonen net for anomaly detection in network security. IEEE Trans. on SMC – part B 35(2) (2005) 302-312
4. Zanero, S.: Analyzing TCP traffic patterns using self organizing maps. Proc. ICIAP 2005, Cagliari, Italy, 83-90
5. Corchado, E., Herrero, A., and Saiz, J.M.: Detecting compounded anomalous SNMP situations using unsupervised pattern recognition. Proc. ICANN 2005, Warsaw, Poland, 905-910.
6. Zheng, J., and Hu, M.: An anomaly intrusion detection system based on vector quantization. ICIE Trans. on Inf. & Syst. E89-D(1) (2006) 201-210.
7. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal 37(2) (1991) 233-243
8. Cisco Secure Consulting: Vulnerability statistics report. 2000
9. D. E. Rumelhart and J. L. McClelland, Parallel distributed processing. Cambridge, MA: MIT Press, (1986).
10. K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks. 2(5), 359-66 (1989).
11. Lee, W., Stolfo, S.J., and Mok, K.W.: Adaptive intrusion detection: a data mining approach. Artificial Intelligence Review 14(6), 533–567
12. B. Widrow and M.A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Back Propagation," Proc. IEEE, vol. 78, no. 9, pp. 1415-42, 1990.