Neuro-Immune and Self-Organizing Map Approaches to Anomaly Detection: A Comparison

Fabio González[†] **and Dipankar Dasgupta**
Division of Computer Science
The University of Memphis
[†]and Universidad Nacional de Colombia
{fgonzalz, ddasgupt}@memphis.edu

Abstract

The purpose of this work is to investigate a hybrid approach (neuro-immune technique) for anomaly detection on time series data. In many anomaly detection applications, only positive (normal) samples are available for training purpose. However, conventional classification algorithms need both positive and negative samples. The proposed approach uses normal samples to generate abnormal samples that are subsequently used as training data for a neural network. The approach is compared against an anomaly detection technique that uses self-organizing maps to cluster the normal data sets (samples).[1]

1 Introduction

The anomaly detection problem can be stated as a two-class classification problem: given an element of the space, classify it as *normal* or *abnormal*. Different terminologies are used in different applications, such as "novelty [3] or surprise [13] detection", "fault detection" [20], and "outlier detection" . Accordingly, many approaches have been proposed which include statistical [4], machine learning [15], data mining [16] and immunological inspired techniques [2, 8, 11].

In many anomaly detection applications, however, negative (abnormal) samples are not available at the training stage. For instance, in a computer security application, it is difficult, if not impossible, to have information about all possible attacks. In the machine learning approaches, the lack of samples from the abnormal class causes difficulty in the application of supervised techniques (e.g. classification). Therefore, the obvious machine learning solution is to use an unsupervised algorithm (e.g. clustering).

In our previous work [9], we presented an approach inspired by the immune system that allows the application of conventional classification algorithms to perform anomaly detection tasks. This approach uses a negative selection algorithm (NSA) [6] coupled with a classification algorithm to produce an anomaly detection function. The paper [9] examines the possibility of combine NSA with a neural network classifier in order to detect anomalies in a time series. The purpose of the present work is to perform further experimentation and compare the results to those produced by an unsupervised technique that clusters the normal samples.

The clustering technique used for this purpose is self-organizing maps (SOM) [14]. It is applied to the normal samples to produce clusters that constitute a compact description of the normal space. This compact representation is subsequently used to classify new samples as normal or abnormal [7, 17, 12].

2 Neuro-Immune Technique for Anomaly Detection

The NSA was initially proposed by Forrest and her group [6] based on the principles of self/non-self discrimination in the immune system. It uses as input, a set of strings that represents the normal data (self set) in order to generate detectors in the non-self space. The negative detectors are chosen by matching them to the self strings: if a detector matches it is discarded, otherwise, it is kept. Some efficient implementations of the algorithm (for binary strings) that run in linear time with the size of self have been proposed [5, 6, 11].

[1]Published in Proceedings of the 1st International Conference on Artificial Immune Systems, pages 203-211, Canterbury, UK, Sept. 9-11, 2002.

However, the time complexity of these algorithms is exponential on the size of the matching window (the number of bits to use in the comparison of two binary strings).

We proposed [9] a new version of the NSA that represents the self/non-self space as n-dimensional real vectors. One of the advantages of this approach is that it is easier to extract meaningful knowledge from the generated detectors as the representation is closer to that of the problem space. The detectors generated by the NSA are used as artificial abnormal samples that serve as input to a classification algorithm that learns an anomaly detection function.

Similar to the binary-valued NSA [6], the real-valued NSA [9] tries to cover the non-self space with minimum number of detectors. This is accomplished by an iterative process that updates the position of the detectors driven by two objectives: to move detectors away from self points and to keep the detectors separated in order to maximize the covering of non-self space (non-overlapping). This algorithm is shown in Figure 1.

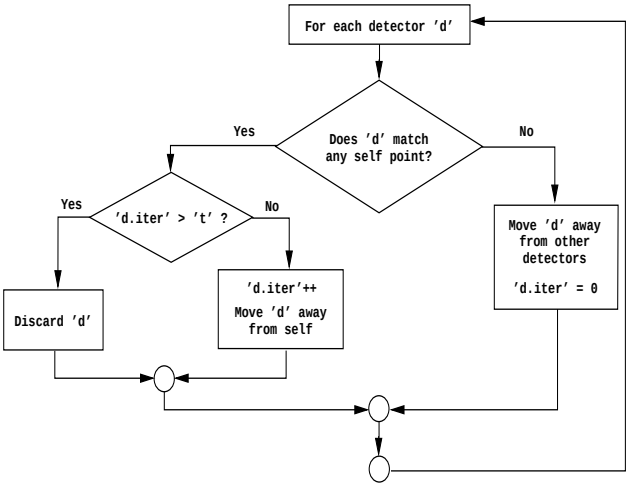


Figure 1: Illustrates an iteration of the real-valued negative selection algorithm with a flow diagram.

We used a hybrid approach by combining NSA and a neural network–multi-layer perceptron (MLP) with a hidden layer trained using back-propagation [10]. Figure 2 illustrates the basic idea of the approach. During the training stage, the input corresponds to the normal samples (feature vectors extracted from normal time series), while the NSA [9] is used to generate abnormal samples. Subsequently, the normal and abnormal samples are used to train a neural network classifier. The trained neural network corresponds to the anomaly detection function that is used during the testing phase to classify new samples as normal or abnormal.

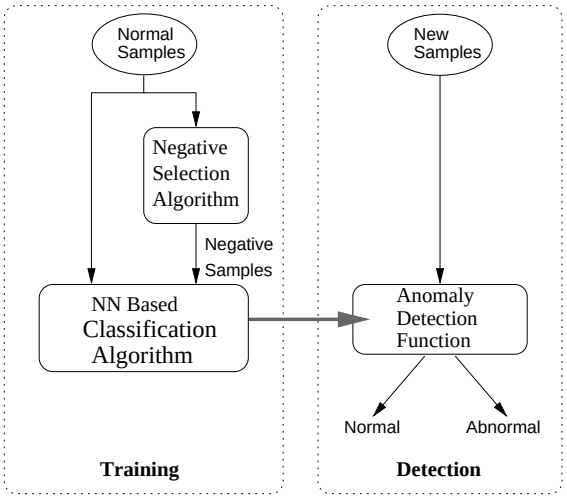


Figure 2: A process to generate an anomaly characterization function from normal samples.

3 Anomaly Detection Using Self-Organizing Maps

A self-organizing map (SOM) is a type of neural network that uses competitive learning [14, 10]. A SOM is able to capture the important features contained on the input space and provides a structural representation that preserves a topological structure. The output neurons of a SOM are organized in a one- or two-dimensional lattice. The weight vectors of these neurons represent prototypes of the input data that can be interpreted as the centroids of clusters of similar samples.

In our experiments, we used SOM to cluster the normal samples. After the network is trained, the generated clusters are used to determine if a new sample is normal or abnormal. The basic idea is: if a new sample is 'close' enough to a normal cluster it is considered normal, otherwise it is classified as abnormal.

In general, we have a distance function $dist(s, K)$ that measures how close the sample s is to the cluster, K. To determine the abnormality of a new sample, the following function is used:

$$dist(s, Normal) = \min\{dist(s, K_i) \mid K_i \in C\}$$

$$\chi_{abnormal}(s) = \begin{cases} 1 & \text{if } dist(s, Normal) \geq t \\ 0 & \text{otherwise} \end{cases},$$

where, C is the set of clusters (found by the SOM algorithm) that represents the normal sub-space. If we think the function $dist(s, Normal)$ is a kind of membership function[2] of the abnormal subspace, the function $\chi_{abnormal}(s)$ corresponds to the crisp version of it. In this case, the value t represents a threshold that defines the boundary between the normal and abnormal classes.

In order to determine a good distance measure $dist(s, K)$, we tested three options (in all the cases w_K represents the centroid of the cluster K, neuron weights):

- **Euclidean distance**. This is the natural (or naive) choice since the SOM algorithm uses it to determine if a sample belongs to a given cluster:

$$dist(s, K) = \|s - w_K\|$$

- **Normalized distance**. The idea is to take into account the size of the cluster. Some clusters can be very sparse and others can have all the elements concentrated around the centroid. A measure of the size is the standard deviation. So, the standard deviation of the distance to the centroid of all the elements in a cluster (σ_K) is calculated and it is used to normalize the distance:

$$dist(s, K) = \frac{\|s - w_K\|}{\sigma_K}$$

- D_∞ **Minkowsky distance**. The Euclidean distance gives the same importance to all the features. So, it is possible that a sample with a non-negligible deviation in one feature will be considered as having the same overall deviation as a pattern with small deviation on many features. The D_∞ distance only takes into account the maximum of the differences for all the features:

$$dist(s, K) = \max\{|s_i - w_{K_i}| \text{ for } i = 1, \ldots, n\}$$

4 Time Series Data Set

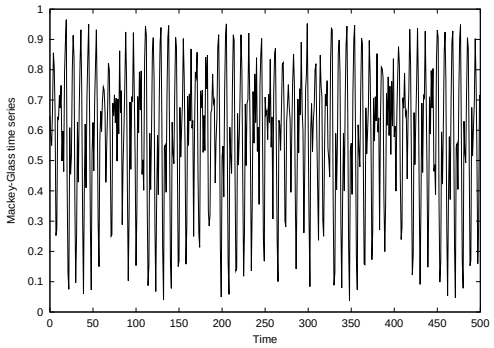
We used the Mackey-Glass equation to generate time series data. It is a non-linear, delay-differential equation whose dynamics exhibit chaotic behavior for some parameter values. The equation is:

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1 + x^c(t-\tau)} - bx(t)$$

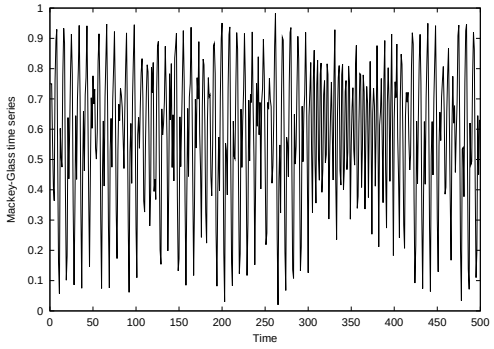
[2]Strictly speaking, this is not a membership function since it is not bounded. However, we can apply, for instance, a sigmoid function to make it bounded.

The parameters chosen were $a = 0.2$, $b = 0.1$, and $c = 10$. This set of parameters are the general choice in the literature [3, 1]. The parameter τ controls the complexity of the series dynamics. For the first experiment $\tau = 30$ was used to generate the normal samples.

The equation is solved numerically using fourth-order Runge-Kutta method (included in Matlab) with an integration step of 0.02, a sampling rate of 12, and an initial value vector with all its elements equal to 1.1. The normal samples were produced from a time series with 500 elements generated using $\tau = 30$ and discarding the first 1000 samples to eliminate the initial value effect. The resulting time series is shown in Figure 3.a.



(a) normal time series



(b) time series with an anomaly

Figure 3: Mackey-Glass series: (a) normal, using $\tau = 30$, (b) with an anomaly, $\tau = 17$ from 300 to 400.

The features are extracted using a sliding overlapping window of size n. If the time series has the values: $x_1, x_2, ..., x_m$, the feature set generated from it will be the following:

$$
\begin{array}{llll}
(x_1, & x_2, & \ldots & x_n) \\
(x_2, & x_3, & \ldots & x_{n+1}) \\
\vdots & \vdots & \vdots & \vdots \\
(x_{m-n+1} & x_{m-n+2} & \ldots & x_m)
\end{array}
$$

So, from a time series with m elements and using a sliding window of size n, we can generate (m-n+1) samples.

In order to perform the testing, we need new normal and abnormal samples. For abnormal samples, we change the parameter of the series (τ). For the preliminary experiments, we used τ = 17 (as used in [3, 1]). Figure 3.b shows an example of a time series with an abnormal segment (time 300 to 400) where the parameter τ was changed from 30 to 17.

5 Experimental Results

5.1 Experiments using SOM technique

To perform SOM experiments, we used a tool that is available on Internet (GeneCluster [19], `http://www-genome.wi.mit.edu/cancer/software/software.html`). This tool is primarily used to cluster gene expression information, however, it can be applied to any kind of data. We found the visual representation of clusters is very useful for our purpose.

For this set of experiments, we used the normal Mackey-Glass data, as plotted in Figure 3, for training. A window size of 4 was used to generate the feature vectors. Accordingly, a total number of 497 patterns were generated. The clusters generated by the application using an output grid of 6×4 neurons are shown in Figure 4. Each box shows a cluster centroid (middle curve) as well as the variations for each feature: maximum value (upper curve) and minimum value (lower curve) in the cluster. The number of samples on each cluster is also presented.

We also tested output grids with 3×4, and 8×8 neurons. In all cases, the SOM algorithm was run for 100 iterations using Gaussian neighborhood. The initial and final learning rate were 0.1 and 0.005 respectively. The initial σ value was 5 and the final was 0.2.

During testing, we applied the technique described on section 3 using the data in Figure 3.b. Figure 5 shows the anomaly detection function (i.e. $dist(s, Normal)$) for three different distance measures using an SOM with an 8×8 output layer configuration.

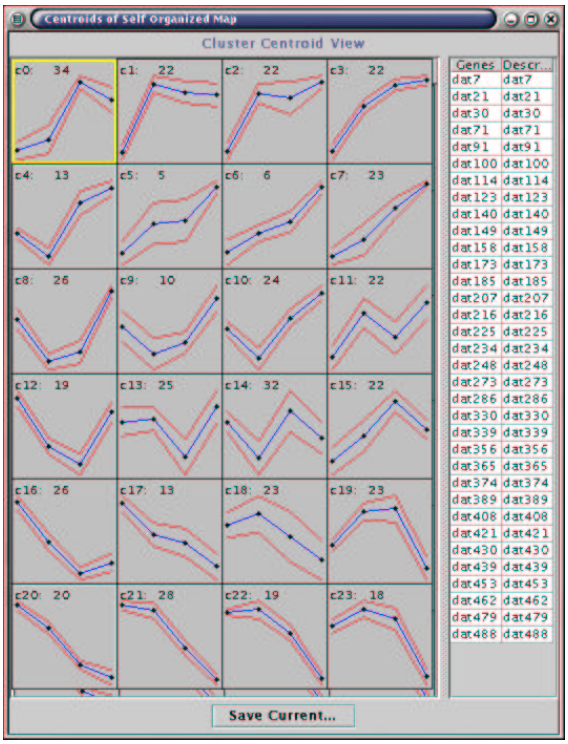
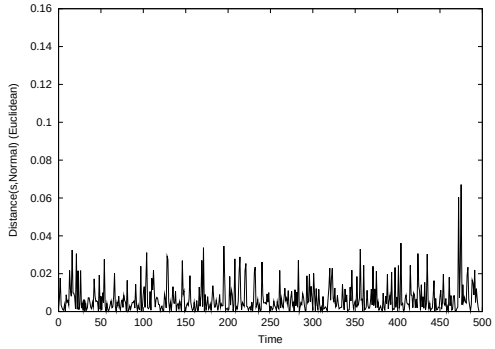


Figure 4: Clustering of the normal data produced by GeneCluster [19] (columns in right hand side are not relevant to our experiments).

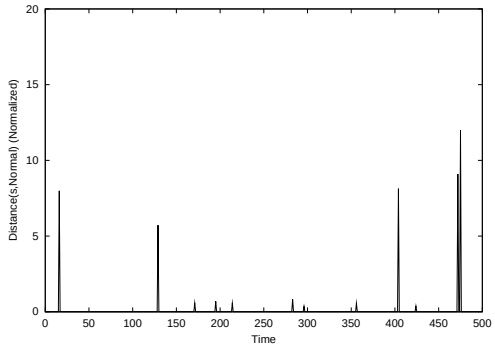
It is clear that the anomaly detection based on Euclidean distance (Figure 5.a) is not able to detect the anomalous patterns. The normalized distance does not improve either. The plots corresponding to D_∞ Minkowsky distance show an increase on the average value between the time 300 and 400 which corresponds to the anomalous section. This indicates that this distance measure is able to detect the anomalous patterns.

It is to be noted that the change on the number of output neurons reflected on the shape of the function, i.e. the more neurons on the output, the smoother the function. This is explained by the fact that more neurons imply more clusters which can approximate the normal set better.

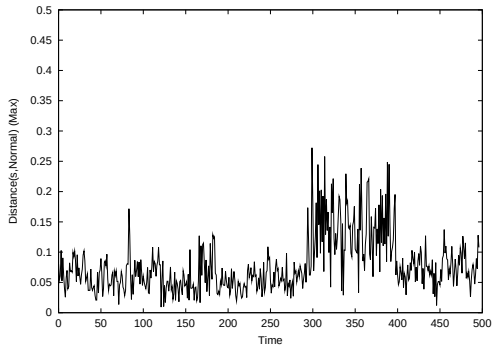
The Euclidean distance and the normalized distance assume that the clusters are spherical, that is, the distribution is the same for all directions. It seems that this is not the case, as it is evidenced in the poor performance of these distance measures. The D_∞ distance eliminates, to some degree, the interference between features and this seems to be an advantage for this specific problem. However, its main drawback is that it does not take into account the shape of the clus-



(a) Euclidean distance



(b) normalized distance



(c) D_α Minkowsky

Figure 5: Anomaly function ($dist(s, Normal)$) generated using the SOM-based technique and applied to the testing set. The net has an 8×8 output layer. Each graph represents a different distance measure: (a) Euclidean distance, (b)Normalized distance, and (c) D_∞ Minkowsky distance.

ter. Our hypothesis is that a distance measure such as Mahalanobis distance will perform much better, since it can represent ellipsoid clusters.

The anomaly function presents many peaks; in order to smooth it, a moving average technique was applied. The new output $\widehat{O_t}$ is calculated from the old output O_t using the following formula:

$$\widehat{O_t} = \frac{\sum_{i=1}^{s} O_{t-i}}{s}$$

where s is the smoothing factor and indicates the size of the averaging window. Figure 6 shows the results of the smoothing process for the anomaly function corresponding to D_∞ Minkowsky distance using $s = 10$. It is evident from the figure, how the smoothing process makes a clear boundary between the normal and the abnormal sections. As it was discussed previously, the contrast is bigger for the SOM with more output neurons (8×8). A quantitative comparison of these anomaly functions is performed in section 5.3.

5.2 Experiments using Neuro-Immune anomaly detection technique

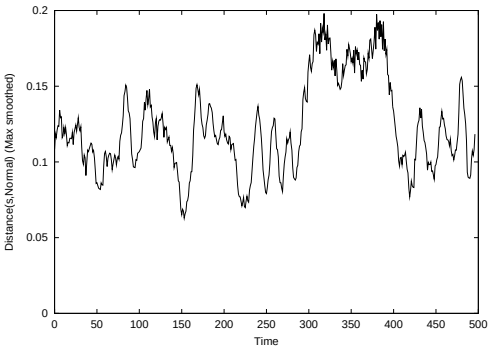
The data in Figure 3 was used to generate the training set using a window size of 4. This generated 497 normal samples that were used as input for the NSA which generated 400 abnormal samples. The normal samples were assigned an output value of 0.0 and the abnormal samples an output value of 1.0. For the classification phase, a multilayer neural network with 4 inputs, and one output neuron was used. We tested three different MLPs with 6, 12, and 16 hidden neurons respectively.

The training algorithm was back-propagation with momentum using the following parameters: learning rate 0.2, momentum 0.9, number of epochs 4000. Figure 7 shows the output of the a MLP with 16 hidden units when applied to the testing set.

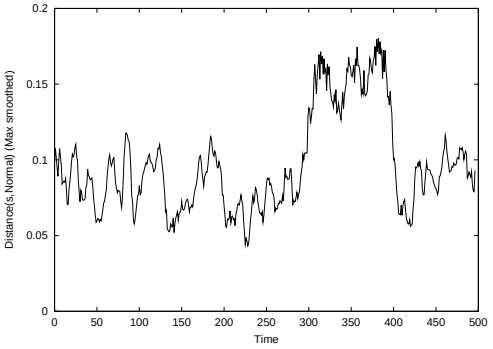
The results show that the trained MLPs are able to detect the anomalous segment present on the testing set. The output from the simplest MLP (six hidden neurons) shows more spikes. A possible explanation is that a larger number of hidden neurons allows to represent more details of the normal subspace. However, the smoothing process is able to eliminate most of them.

5.3 Comparison of the two techniques

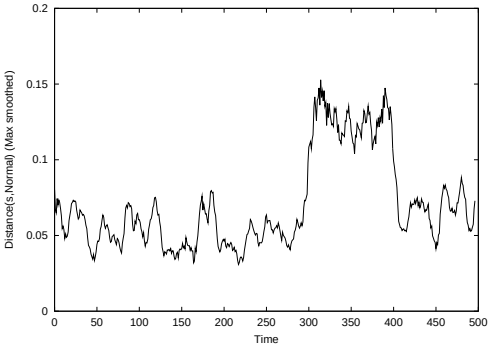
In order to compare the two techniques (SOM and neuro-immune) it is necessary to define a measure of



(a)

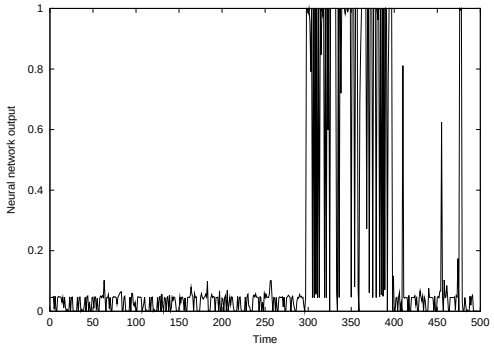


(b)

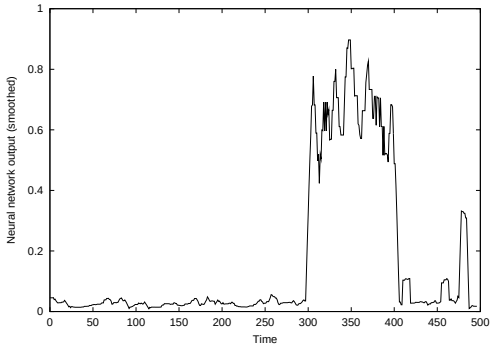


(c)

Figure 6: D_∞ Minkowsky distance anomaly function smoothed using a moving average with parameter $s = 10$. The different plots represent different topologies: (a) 3×4 neurons, (b) 6×4 neurons, and (c) 8×8 neurons.



(a) raw output (without using the smoothing function)



(b) smoothed output using $s = 10$

Figure 7: Neural network output for the testing set using 16 hidden neurons (neuro-immune technique).

accuracy for the classification. The idea is to calculate the number of true positives (TP, anomalous elements identified as anomalous), true negatives (TN, normal elements identified as normal), false positives (FP, normal elements identified as anomalous) and false negatives (FN, anomalous elements identified as normal). These values are used to calculate two measures of effectiveness:

$$\text{Detection rate} = \frac{TP}{TP + FN}$$

$$\text{False alarm rate} = \frac{FP}{TN + FP}$$

In general, we want a very high detection rate with a very low false alarm. However, there is a trade-off between these two measures. This trade-off can be shown using ROC (receiver operating characteristics) curves [18]. The sensitivity of the system is controlled

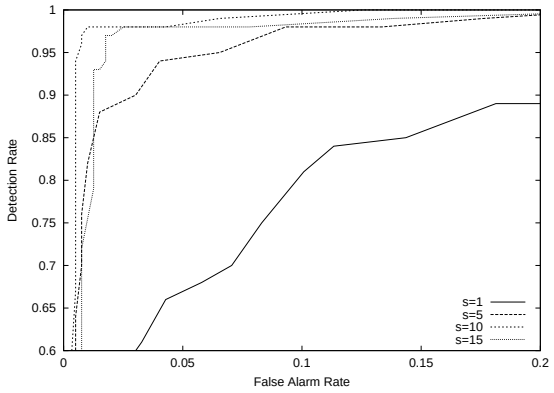
by a threshold that determines when a new sample is normal or abnormal. By varying this threshold, we can obtain different values for the detection and false alarm rates which are used to plot ROC curves.

Figure 8 shows ROC curves for SOM-based and neuro-immune anomaly detection techniques. In all cases, it is clear that the smoothing parameter (s) improves the classification accuracy. However, the SOM-based technique seems to be more sensitive to its value. This is explained by the fact that the anomaly detection function generated by this method is not as smooth as the one generated by the neuro-immune method.

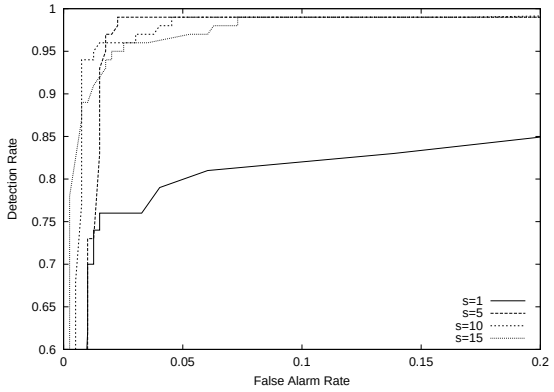
For the two methods the most complex networks generate better results. As it was explained previously, a most complex network allows a more detailed modeling of the normal subspace.

The best anomaly detection functions from the two methods are shown in Figure 9. There is no clear winner. The anomaly detection function generated by the SOM method is able to produce a very good detection rate with a low false alarm rate. But, if a small increase on the false alarm rate is allowed, the neuro-immune method is able to produce a better detection rate than the SOM method.

An important issue on anomaly detection is how to find a good threshold value that produces a detection rate with an acceptable false alarm rate. This could be very difficult if the anomaly detection function is very sensitive to this threshold. Figure 10 shows how the detection and false alarm rates change when the threshold is modified. For the neuro-immune method, the detection rate increases gradually as the threshold increases. The false alarm rate only increases at the end, producing a good range of threshold values where it is possible to have a high detection rate keeping the false alarm rate low. In the case of the SOM-based method, the detection rate changes suddenly with a small change on the threshold. The range of threshold values that can produce a good detection rate with a low false alarm rate is very small. This means, that the threshold has to be chosen very carefully and that a small variation can easily deteriorate the performance of the anomaly detection system.



(a) SOM-based method



(b) Neuro-Immune method

Figure 8: ROC curves for different values of the threshold parameter (s).

6 Conclusions

In this paper, we compared two different approaches for anomaly detection: one uses a neuro-immune technique and the other uses self-organizing maps (SOM). Their performances, from the point of view of classification accuracy, appears to be very similar. In both cases, the smoothing process (moving average) improved the classification performance significantly.

As it was expected, more complex neural networks had better performance; SOM networks were, in general, more complex than the feed-forward networks (MLP) used on the neuro-immune technique that exhibit similar performance. For instance, two networks that are compared (shown in figure 10) have $(1 + 4) \times 16 + 16 = 97$ weights (neuro-immune) and $4 \times 64 = 256$ weights (SOM) needed to be trained.

In general, the anomaly detection functions generated by the neuro-immune method were relatively smoother. This represents a clear advantage as they are less sensitive to changes on the threshold. How-

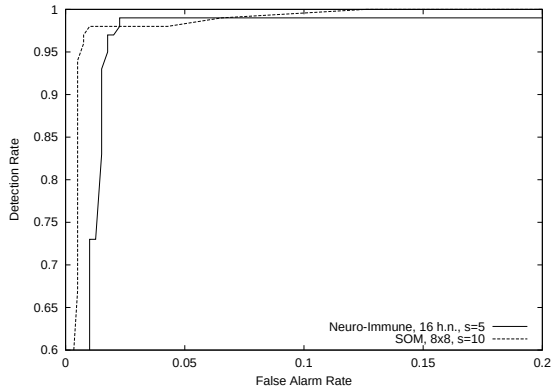
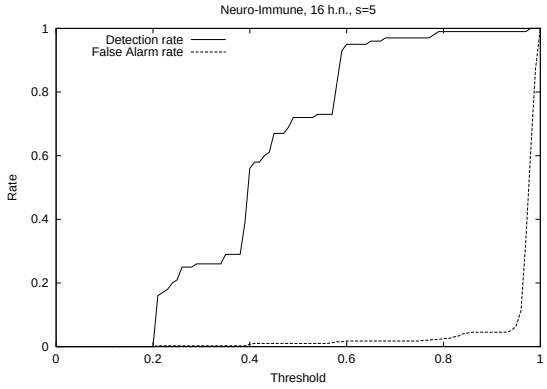


Figure 9: Best anomaly detection functions of each method.

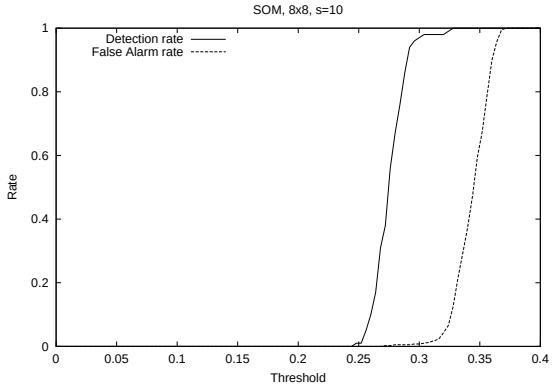
ever, there is room for improvement for the SOM method too. For instance, a distance measure that takes into account the shape of the cluster (like Mahalanobis distance) will probably improve the performance of the SOM method. So, it is necessary to test new distance measures and perform additional experiments using wide variety of data sets in order to make a fair comparison.

7 Acknowledgments

This work was funded by the Defense Advanced Research Projects Agency (no. F30602-00-2-0514) and National Science Foundation (grant no. IIS-0104251).



(a) Neuro-Immune method



(b) SOM-based method

Figure 10: Evolution of detection and false alarm rates when the threshold is modified.

References

[1] T. Caudell and D. Newman. An adaptive resonance architecture to define normality and detect novelties in time series and databases. In *IEEE World Congress on Neural Networks*, pages 166–176, Portland, Oregon, 1993.

[2] D. Dagupta and F. González. An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Transactions on Evolutionary Computation*, 6(3):1081–1088, June 2002.

[3] D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology. In *Proceedings of the International Conference on Intelligent Systems*, pages 82–87, June 1996.

[4] D. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13(2):222–232, February 1987.

[5] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: Algorithms. In *Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy*, pages 110–119, Oakland, CA, 1996. IEEE Computer Society Press.

[6] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Proc. IEEE Symp. on Research in Security and Privacy*, pages 202–212, May 1994.

[7] K. L. Fox, R. R. Henning, J. H. Reed, and R. P. Simonian. A neural network approach towards intrusion detection. In *Proc. 13th NIST-NCSC National Computer Security Conference*, pages 125–134, 1990.

[8] F. González and D. Dasgupta. An imunogenetic technique to detect anomalies in network traffic. In *Gecco 2002: proceedings of the genetic and evolutionary computation conference*, pages 1081–1088, New York, 9-13 July 2002. Morgan Kaufmann Publishers.

[9] F. González, D. Dasgupta, and R. Kozma. Combining Negative Selection and Classification Techniques for Anomaly Detection. In *Proceedings of the Congress on Evolutionary Computation*, pages 705–710, Honolulu, HW, May 2002.

[10] S. Haykin. *Neural Networks : A Comprehensive Foundation*. Macmillan, New York, 1994.

[11] S. A. Hofmeyr and S. Forrest. Architecture for an artificial immune system. *Evolutionary Computation*, 8(4):443–473, 2000.

[12] W. H. Hsu, L. S. Auvil, W. M. Pottenger, D. Tcheng, and M. Welge. Self-organizing systems for knowledge discovery in databases. In *In Proceedings of the International Joint Conference on Neural Networks (IJCNN-99)*, Washington, DC, July 1999.

[13] E. Keogh, S. Lonardi, and B. Y. Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, Alberta, Canada, 2002.

[14] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[15] T. Lane. *Machine Learning Techniques For The Computer Security*. PhD thesis, Purdue University, West Lafayette, IN, 2000.

[16] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, pages 26–29, San Antonio, TX, January 1998.

[17] L. Portnoy, E. Eskin, and S. J. Stolfo. Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, Philadelphia, PA, November 2001.

[18] F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings Of 15th International Conference On Machine Learning*, pages 445–453, San Francisco, Ca, 1998. Morgan Kaufmann.

[19] P Tamayo, D Slonim, J Mesirov, Q Zhu, S Kitareewan, E Dmitrovsky, ES Lander, and TR Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A*, 96(6):2907–12, March 1999.

[20] T. Y. Yoshikiyo. Fault detection by mining association rules from house-keeping data. In *Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2001)*, Montreal, Canada, June 2001.