An Immuno-Fuzzy Approach to Anomaly Detection

Jonatan Gómez* Fabio González* Dipankar Dasgupta

Computer Science Division, The University of Memphis, TN 38152 USA
*and Departamento de Ingeniería de Sistemas, Universidad Nacional de Colombia
{jgomez,fgonzalz,dasgupta}@memphis.edu

***Abstract*— This paper presents a new technique for generating a set of fuzzy rules that can characterize the non-self space (abnormal) using only self (normal) samples. Because, fuzzy logic can provide a better characterization of the boundary between normal and abnormal, it can increase the accuracy in solving the anomaly detection problem. Experiments with synthetic and real data sets are performed in order to show the applicability of the proposed approach and also to compare with other works reported in the literature.**

I. Introduction

The detection of unusual behavior patterns is an important problem in computer security as most security breaches exhibit anomalous system behavior. However, anomalous patterns can also be generated when normal behavior changes.

The problem of anomaly detection is also studied in other contexts. Different terminologies are used in different applications, such as "novelty [1] or surprise [2] detection", "fault detection" [3], and "outlier detection". Accordingly, many approaches have been proposed which include statistical [4], machine learning [5], data mining [6] and immunological inspired techniques [7], [8], [9].

Artificial immune systems have been applied successfully in anomaly based computer network intrusion detection [9], [10], [11]. However, there are some problems that have prevented this approach from being applied extensively:

- A large number of detectors is needed in order to guarantee good levels of detection, specially with binary representation. For some problems the number of detectors could be unmanageable [12].
- The low level representation of the detectors prevents, in many cases, extraction of meaningful domain knowledge. This makes it difficult to implement modules that explain, using high level terms, the reasons to report an anomaly.
- A sharp distinction between the normal and the abnormal. This divides the space into two subsets *self* (normal) and *non-self* (abnormal) . An element of the space is considered abnormal if there exists an antibody that matches it. Clearly, the normalcy is not a crisp concept. A natural way to characterize the normal is by defining a degree of normalcy, that is, the set of normal elements can be represented as a fuzzy set.

These issues were addressed with some success in [7] by evolving the detectors (in the non-self space) with a genetic search and dividing the non-self space into different levels. The evolved detectors had hyper-rectangular shape that could be interpreted as rules. The paper demonstrated the usefulness of such a technique to detect a wide variety of intrusive activities on networked computers.

An improvement of this technique was presented in [8]. Specifically, it used a different niching technique to generate the rule detectors. The initial algorithm used a sequential niching technique, whereas the new one [8] used deterministic crowding, which proved to be more efficient in generating good anomaly detectors.

The discrete division of the non-self space on levels of deviation proposed in [7], [8] allows a non-crisp distinction between self (normal) and non-self (abnormal). Then, it can be considered as a first step to define a real fuzzy characterization of non-self. So, this paper is an extension of these previous works [7], [8] by using fuzzy logic. Specifically, fuzzy rules will be used, instead of crisp rules, to cover the non-self space (i.e. fuzzy detectors).

II. Previous work

Forrest et al. [13] developed a negative selection algorithm (**NSA**) based on the principles of self/non-self discrimination in the natural immune system (**NIS**). The NSA is as follows ([14]):

- Given a feature space U, list of features that represents the system state, define self (normal states) as a collection S of elements in U.
- Generate a set R of *detectors*, each of which fails to match any string in S. An approach that mimics what happens in the NIS would generate random detectors and discard those that match any element in the self set. However, a more efficient approach will try to minimize the number of generated detectors while maximizing the covering of the non-self space (abnormal).
- Monitor S for changes by continually matching the detectors in R against S. If any detector ever matches, then a change is known to have occurred, as the detectors are designed not to match any of the original strings in S.

There are different variations of the NSA with different application including: anomaly detection [9], [15], fault detection [16], [17], detect novelties in time series [1], [18], and function optimization [19].

A real valued negative algorithm was proposed by Dasgupta and Gonzalez [7] (**RNS**) and improved with a deterministic crowding in [8]. The main differences between the original NSA [13] and RNS are:

- The elements of self/non-self space are represented by n-dimensional real valued vectors.
- The detectors correspond to hyper-rectangles in $\mathbb{R}^n$ and represent high-level rules.
- The detectors are evolved using a genetic algorithm that maximizes the covering of the non-self space while minimizing the matching of self points. A niching technique is used in order to evolve multiple detectors that cover cooperatively the non-self space.

The basic structure of a rule detector is as follows:

R: If $x_1 \in [lo_1, hi_1] \wedge ... \wedge x_n \in [lo_n, hi_n]$ then **non_self**

where,

- $(x_1, ..., x_n)$ is a feature vector
- $[lo_i, hi_i]$ specifies the lower and upper values for the feature x_i in the condition part of the rule R.

The condition part of each rule defines a hyper-rectangle in the feature space ($[0, 1]^n$). Then, a set of these rules tries to cover the non-self space with hyper-rectangles. For the case $n = 2$, the condition part of a rule represents a rectangle.

This work also proposed a mechanism that allows to estimate the level of deviation from the normal. The non-self space is further divided into different levels of deviation. The genetic algorithm is run as many times as deviation levels are needed. The difference between each run is determined by a variability parameter which specify the degree of variation from the normal set.

Other works also have proposed the use of hyper-rectangles to characterize data in high-dimensional spaces. Simpson [20], [21] proposed a fuzzy min-max neural network architecture for classification and clustering of spatial data. In this technique, the hyper-rectangles represent fuzzy clusters. A deterministic procedure to place and size the hyper-rectangles was used; however, its performance was very sensitive to the algorithm parameters and the order of presentation of the data samples. Fogel and Simpson [22] used evolutionary programming to optimize the position of hyper-clusters to cluster data. This work was extended [23] to support hyper-rectangles not necessarily aligned with the coordinated axis; however, this work was restricted to a 2-dimensional space. The main difference between these approaches and the technique described at the beginning of this section is that the generated hyper-rectangles cover the input data (positive space), whereas in the previous technique the hyper-rectangles cover the negative space.

III. Proposed approach

Our idea is to extend the approach proposed in [7], [8] to use fuzzy rules instead of crisp rules. That is, given a set of self samples, generate fuzzy detector rules in the non-self space that can determine if a new sample is normal or abnormal. As it will be shown later, the use of fuzzy rules improves the accuracy of the method and produces a measure of deviation from the normal that does not need a discrete division of the non-self space. Notice that, unlike the previous approach, the fuzzy rules are not represented by hyper-rectangles.

A. Anomaly detection with fuzzy rules

In the self/non-self space $[0, 1]^n$, an element x in this space is represented by a vector $(x_1, \ldots, x_n)$ where $x_i \in [0, 1]$. A fuzzy detection rule has the following structure:

$$\textbf{If } x_1 \in T_1 \wedge \ldots x_n \in T_n \textbf{ then } \text{non\_self},$$

where

$(x_1, \ldots x_n)$: element of the self/non-self space
T_i: fuzzy set
$\wedge$: fuzzy conjunction operator (in our case, $\min()$)

The fuzzy set T_i is defined as a combination of basic fuzzy sets (linguistic values). Given a set of linguistic values $S = \{S_1, \ldots, S_m\}$ and a subset $\widehat{T}_i \subseteq S$ associated to each fuzzy set T_i,

$$T_i = \bigcup_{S_j \in \widehat{T}_i} S_j,$$

where $\bigcup$ corresponds to a fuzzy disjunction operator. We used the addition operator defined as follows:

$$\mu_{A \cup B}(x) = \min\{\mu_A(x) + \mu_B(x), 1\}.$$

An example of fuzzy detector rules in the self/non-self space with dimension $n = 3$ and linguistic values $S = \{L, M, H\}$:

$$\textbf{If } x_1 \in L \wedge x_2 \in (L \cup M) \wedge x_3 \in (M \cup H) \textbf{ then } \text{non\_self}$$

In our experiments, the basic fuzzy sets correspond to a fuzzy division of the real interval $[0.0, 1.0]$ using triangular and trapezoidal fuzzy membership functions. Figure 1 shows an example of such a division using five basic fuzzy sets.

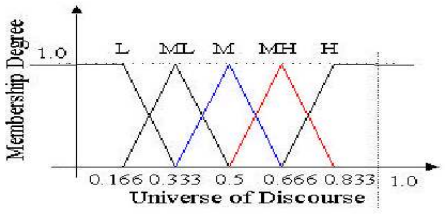

Fig. 1. Partition of the interval [0.0,1.0] in basic fuzzy sets.

Given a set of rules R_1, R_2, .., R_m, the degree of abnormality of a sample x is defined by

$$\mu_{\text{non\_self}}(x) = \max_{i=1,\ldots m} \{eval_{R_i}(x)\},$$

where $eval_{R_i}(x)$ represents the fuzzy true value produced by the evaluation of the condition of fuzzy rule R_i, and $\mu_{\text{non\_self}}(x)$ represents the degree of membership of x to the non-self set; thus, a value close to zero means that x is normal and a value close to 1 indicates that it is abnormal.

B. Evolving fuzzy detector rules

In our previous work [7], we used genetic algorithm (**GA**) combined with niching technique to evolve a set of detector rules that cover cooperatively the non-self space. In the present work, we use the same algorithm, but using deterministic crowding (**DC**)[24] as niching technique since it was shown to perform better than sequential niching [25], as it was demonstrated in [8]. The input to the GA is a set of n-dimensional feature vectors $Self = \{x^1, ..., x^m\}$, which represents samples of normal behavior, the population size and the number of generations. Details of this algorithm can be found in the original work of Gonzalez et al. [8].

1) Chromosome representation: Each individual (chromosome) in the genetic algorithm represents the condition part of a rule, since the consequent part is same for all rules (the sample belongs to non-self). The condition is a conjunction of atomic conditions. Each atomic condition, $x_i \in T_i$, corresponds to a gene in the chromosome that is represented by a sequence $(s_1^i, \ldots, s_m^i)$ of bits, where $m = |S|$ (the size of the set of linguistic values), and $s_j^i = 1$ if and only if $S_j \subseteq T_i$. That is, the bit s_j^i is 'on' if and only if the corresponding basic fuzzy set S_j is part of the composite fuzzy set T_j. Figure 2 shows the structure of a chromosome which is $n \times m$ bits long (n is the dimension of the space and m is the number of basic fuzzy sets).

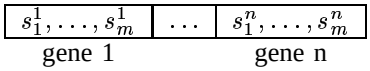

Fig. 2. Structure of the chromosome representing the condition part of a rule.

2) Fitness Evaluation: The fitness of a rule R is calculated taking into account the following two factors:

- The number of elements in the training set S that belong to the subspace represented by the rule is calculated as:

$$covering(R) = \frac{\sum_{x \in Self} eval_R(x)}{|Self|}$$

- The fuzzy measure of the volume of the subspace represented by the rule is calculated as:

$$volume(R) = \prod_{i=1}^{n} measure(T_i),$$

  where $measure(T_i)$ corresponds to the area under the membership function of the fuzzy set T_i.

The fitness is defined as:

$$fitness(R) = C \cdot (1 - covering(R)) + (1 - C) \cdot volume(R),$$

where C, $0 \leq C \leq 1$, is a coefficient that determines the amount of penalization that a rule suffers if it covers normal samples. The closer the coefficient to 1 the higher the penalization. In our experimentation, we used values between 0.8 and 0.9.

3) Distance Measure: A good measure of distance between individuals allows the DC to replace individuals with closer individuals and to preserve niches. In this work, we use the Hamming distance, because there is a strong relation between each bit in the chromosome with a single fuzzy set of some particular attribute in the search space. For example, if the s_i^j bit (see Figure 2), in both parent and child fuzzy rule detectors is set to one, both individuals include the atomic sentence $x_i \in s_j$, i.e., they use the jth fuzzy set to cover some part of the ith attribute. Then, higher the number of bits the parent and the child have in common, higher common area they will cover.

IV. Experimentation

In order to determine the performance of the proposed approach (Evolving Fuzzy Rules Detectors - **EFR**), experiments were conducted on three different data sets, see table I. Each individual performs a random sampling of the training set for evaluating its fitness. The sampling size was fixed to 400. Also, two different algorithms were tested in order to compare the performance of the proposed approach: Evolving Rule Detectors (**ERD**), a non fuzzy method as explained in section II, and Parallel Hill Climbing of Fuzzy Rules Detectors (**PHC**).

PHC is an optimization algorithm based on random mutations of potential solutions population. In each iteration, every individual is changed randomly in one bit. If the produced individual is better than the original, the last is replaced by the new one. In other case the original is maintained in the population. It is clear that individuals in the population of a PHC do not exchange information (as in GA).

The algorithms were run for 1000 iterations with a population size of 100 individuals. The mutation probability for the ERD algorithm was fixed to 0.1 and the ERD was run four times, each time with a different level of deviation (0.1, 0.2, 0.3, and 0.4). The crisp detectors (hyper rectangles) generated by each run are combined to define the final set of detectors produced by the ERD. Five fuzzy sets, as shown in figure 1, were used for each feature extracted in the proposed approach EFR and in the PHC algorithm.

TABLE I
DATA SETS USED FOR EXPERIMENTATION

Data Set	Training	Testing	
		Normal	Abnormal
Mackey-Glass	497	396	101
Darpa 99	4000	5136	56
KDD-Cup 99	76222	19056	396745

There are two elements that define the cost function of an anomaly detection system: the false alarm rate (**FA**), the system produces an alarm in normal conditions, and the detection rate (**DR**), the system detects an attack. A good intrusion detection system is one that has low FA and high DR. In order to compare the performance of the proposed approach we generated an average ROC curve [26] for each of

the algorithm tested. All the reported results (ROC curve and DR values) are the average of ten different runs. The reported DR is the average produced by each algorithm when the FA is (2%).

A. Mackey-Glass Time series

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:

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

1) Experimental settings: The Mackey-Glass parameters used in the experimentation were $a = 0.2$, $b = 0.1$, and $c = 10$. This set of parameters are the general choice in the literature [1], [27]. 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 features are extracted using a sliding overlapping window of size $n = 4$.

2) Results and Analysis: It is clear that proposed approaches based on a fuzzy characterization (the proposed approach and PHC) outperform the one with crisp characterization (ERD), see figure 3. The fuzzy algorithms generated a higher DR than ERD for every given FA rate. Moreover, it is possible to have a DR higher than 75% with 0% FA rate in the fuzzy approaches while only a DR of 6% is reached by the ERD algorithm in the same conditions. On the other hand, the proposed approach (EFR) compares well with PHC. Besides, PHC performs a litter better when the FA is varied between 0 to 7%, but the performance of both methods is similar. Also, EFR performance is a little better when the FA rate is varied between 7% to 10% and after 23%. It is possible that the low dimensionality of the data set is allowing the PHC to reach a good solution (each individual is 20 bits length only).

When the FA rate is fixed to 2%, the performance of the proposed approach and the PHC algorithm are better than the performance of the crisp one (ERD), see Table II. The detection rate is increased in at least 4.5%. Moreover, the standard deviation reached by the fuzzy approaches is very low compared with the crisp one. The difference is almost one order of magnitude. Clearly, the fuzzy approaches produce more consistent results than the crisp one. Also, the number of fuzzy rule detectors (rows 1 and 2) are considerably small compared with the number of crisp detectors (row 3), almost half of number of crisp detectors. Therefore, the fuzzyfication of the search space allows a simple characterization of the abnormal (non-self) space. On the other hand, the difference in DR between EFR and PHC is lower than 0.4% in average. It can be explained for the higher number of fuzzy detectors that PHC generates compared with the number of fuzzy detectors generated by EFR (more than 2 detectors in average).

B. KDD Cup 99

This data set is a version of the 1998 DARPA intrusion detection evaluation data set prepared and managed by MIT Lincoln Labs [28]. Experiments were conducted with the ten percent that is available at the University of Irvine Machine Learning repository[1]. Forty-two attributes, that usually characterize network traffic behavior, compose each record of the 10% data set (twenty-two of them numerical). Also, the number of records in the 10% is huge (492021).

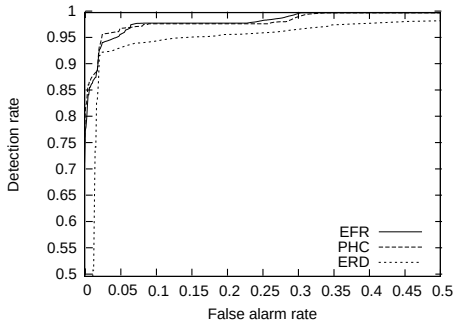

Fig. 3. ROC curves generated by the three algorithms tested with the Mackey Glass data set

TABLE II
COMPARATIVE PERFORMANCE IN THE MACKEY-GLASS PROBLEM

Algorithm	DR%	# Detectors
EFR	**92.27±0.007**	**25.3**
PHC	92.67±0.002	27.6
ERD	87.61±0.071	51.32

1) Experimental settings: We generated a reduced version of the 10% data set including only the numerical attributes, i.e., the categorical attributes were removed from the data set. Therefore, the reduced 10% data set is composed by thirty-three attributes. The attributes were normalized between 0 and 1 using the maximum and minimum values found. 80% of the normal samples were picked randomly and used as training data set, while the remaining 20% was used along with the abnormal samples as a testing set. Five fuzzy sets were defined for the 33 attributes. For reducing the time complexity of the ERD algorithm, 1% of the normal data set (randomly generated), was used as a training data set.

2) Results and Analysis: For this data set the proposed approach outperforms the other two tested algorithms, PHC and ERD, see Figure 4. The DR reached by EFR is 5% more than the performance reached by PHC and 37% more than ERD when the FA is fixed at 2%, see table III. Also, the PHC of fuzzy rule detectors outperforms the crisp one (the DR is increased in at least 32%). Amazingly, the number of detectors using fuzzyfication is very small compared to the number of detectors using the crisp characterization. It can be due to the high dimensionality of the data set (33 attributes). Moreover, the high dimensionality of the data set can be the key factor in reducing the efficiency of the PHC algorithm. Table III compares the performance of the tested algorithms with some results reported in the literature. The performance

[1]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

of EFR is comparable with the performance of approaches reported in the literature and in many cases it performs better, see table III. For example, when EFR is compared with RIPPER-AA the detection rate is higher (4% more abnormal samples detected) with almost the same FA rate. Clearly, the fuzzy characterization of the abnormal space reduces the number of false alarm while the detection rate is increased. Compared with EFRID, the DR rate is almost the same but the FA rate is lower (5% less of false alarms are generated by EFR compared to EFRID). Finally, the number of fuzzy rules detectors generated by EFR is smaller compared with the number of detectors generated by PHC (47.4) and ERD (331.35). Then fuzzy logic allows a better and more compact characterization of the abnormal and normal boundaries.

C. Darpa 99

This data set, is also obtained from the MIT-Lincoln Lab [28]. It represents both normal and abnormal information collected in a test network, where simulated attacks were performed. The data set is composed of network traffic data (tcpdump, inside and outside network traffic), audit data (bsm), and file systems data. We used the outside tcpdump network data for a specific computer (e.g., hostname: marx), and then we applied the tool *tcpstat* to get traffic statistics. The first week's data was used for training (attack free), and the second week's data for testing (this includes some attacks). We only considered the network attacks in our experiments.

1) Experimental Settings: Three parameters were selected (bytes per second, packets per second and ICMP packets per second), to detect some specific type of attacks. These parameters were sampled each minute (using *tcpstat*) and normalized. Because each parameter can be seen as a time series function, the features were extracted using a sliding overlapping window of size $n = 3$. Therefore, two sets of 9-dimensional feature vectors were generated: one as training data set and the other as testing data set. Ten fuzzy sets were defined for each feature extracted.

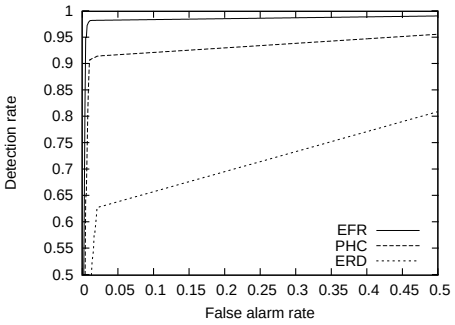

Fig. 4. ROC curves generated by the three algorithms tested with the KDD-Cup 99 data set

2) Results and Analysis: The performance reached by the PHC and EFR algorithms are almost the same and also better than the performance reached by ERD, see Figure 5. These results confirm the hypothesis that a good fuzzyfication of the search space allows fuzzy rule based algorithms to reach a higher performance level than the algorithm based on a crisp characterization of the search space. Table IV compares the performance of the tested algorithms over the Darpa 99 data set. The EFR and PHC algorithms outperformed the ERD algorithm, see Table IV. Also, the number of fuzzy rules is small (near to 9) compared with the number of rules detectors generated by EFR (almost 60). In this way, approaches based on fuzzy logic generate simplest characterization of the abnormal space than the one based on crisp (ERD).

TABLE III
COMPARATIVE PERFORMANCE IN THE KDD CUP 99 PROBLEM

Algorithm	DR%	FA%	# Detectors
EFR	**98.30±0.001**	**2.0**	15
PHC	93.09±0.209	2.0	47.4
ERD	60.94±0.347	2.0	331.35
EFRID[29]	98.95	7.0	-
RIPPER-AA[30]	94.26	2.02	-

TABLE IV
COMPARATIVE PERFORMANCE IN THE DARPA 99 PROBLEM

Algorithm	DR%	# Detectors
EFR	**98.33±0.004**	**8.87**
PHC	99.10±0.008	8.47
ERD	96.47±0.000	59.76

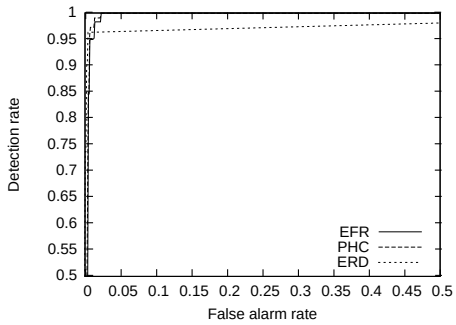

Fig. 5. ROC curves generated by the three algorithms tested with the Darpa 99 data set

V. CONCLUSIONS

This paper presented a new technique that allows to generate a set of fuzzy rules that characterize the non-self space (abnormal) by just taking self (normal) samples as input. This work extended our previous work that used crisp rules as detectors. The experiments performed showed that the proposed approach performs better than the previous one and comparable with other results reported in the literature.

- It provides better characterization of the boundary between normal and abnormal. The fuzzy characteristics of rules provide a natural estimate of the amount of deviation from the normal.

- It shows an improved accuracy on the anomaly detection problem. This can be attributed to the fuzzy representation of the rules which reduce the search space, allowing the evolutionary algorithm to find better solutions.
- It generates a more compact representation of the non-self space by reducing the number of detectors. This is also a consequence of the expressiveness of the fuzzy rules.

REFERENCES

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

[2] E. Keogh, S. Lonardi, and B. 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.

[3] T. Yoshikiyo, "Fault detection by mining association rules from housekeeping data," in *proceedings of international symposium on artificial intelligence, robotics and automation in space (i-sairas 2001)*, (Montreal, Canada), June 2001.

[4] D. Denning, "An intrusion-detection model," in *Ieee computer society symposium on research in security and privacy*, pp. 118–31, 1986.

[5] T. Lane, *Machine learning techniques for the computer security*. PhD thesis, Purdue University, 2000.

[6] W. Lee and S. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th USENIX security symposium*, (San Antonio, TX), 1998.

[7] D. Dagupta and F. González, "An immunity-based technique to characterize intrusions in computer networks," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 281–291, June 2002.

[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*, (New York), pp. 1081–1088, Morgan Kaufmann Publishers, 9-13 July 2002.

[9] S. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.

[10] D. Dasgupta, *Artificial immune systems and their applications*. New York: Springer-Verlag, 1999.

[11] J. Kephart, "A biologically inspired immune system for computers," in *Proceedings of Artificial Life*, (Cambridge, MA), pp. 130–139, July 1994.

[12] J. Kim and P. Bentley, "An evaluation of negative selection in an artificial immune system for network intrusion detection," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, (San Francisco, California, USA), pp. 1330–1337, Morgan Kaufmann, 2001.

[13] 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*, pp. 202–212, 1994.

[14] D. Dasgupta, "An overview of artificial immune systems and their applications.," in *Artificial immune systems and their applications* (D. Dasgupta, ed.), pp. pp 3–23, Springer-Verlag, Inc., 1999.

[15] D. Dasgupta and S. Forrest, "An anomaly detection algorithm inspired by the immune system," in *Artificial immune systems and their applications,*, pp. 262–277, Springer-Verlag, Inc., 1999.

[16] D. Dasgupta and S. Forrest, "Tool breakage detection in milling operations using a negative-selection algorithm," Technical Report CS95-5, Department of Computer Science, University of New Mexico, 1995.

[17] A. Tyrrell, "Computer know thy self! : a biological way to look at fault tolerance," in *2nd euromicro/ieee workshop on dependable computing systems*, (Milan), pp. 129–135, 1999.

[18] F. González, D. Dasgupta, and R. Kozma, "Combining negative selection and classification techniques for anomaly detection," in *Proceedings of the congress on evolutionary computation CEC2002*, (Honolulu, HI), pp. 705–710, IEEE, May 2002.

[19] C. A. C. Coello and N. C. Cortes, "A parallel implementation of the artificial immune system to handle constraints in genetic algorithms: preliminary results," in *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, (Honolulu, Hawaii), pp. 819–824, 2002.

[20] P. Simpson, "Fuzzy min-max neural networks. I. Classification," *IEEE Transactions on Neural Network*, vol. 3, pp. 776–786, sep 1992.

[21] P. Simpson, "Fuzzy min-max neural networks. II. clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 32–, feb 1993.

[22] D. Fogel and P. Simpson, "Experiments with evolving fuzzy clusters," in *Proceedings of the Second Annual Conference on Evolutionary Programming* (D. Fogel and W. Atmar, eds.), (La Jolla, California), pp. 90–97, 1993.

[23] A. Ghozeil and D. B. Fogel, "Discovering patterns in spatial data using evolutionary programming," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, (Stanford University, CA, USA), pp. 521–527, MIT Press, 28–31 July 1996.

[24] S. W. Mahfoud, "Crowding and preselection revisited," in *Parallel problem solving from nature 2* (R. Männer and B. Manderick, eds.), (Amsterdam), pp. 27–36, North-Holland, 1992.

[25] D. Beasley, D. Bull, and R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary Computation*, vol. 1, no. 2, pp. 101–125, 1993.

[26] 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*, (San Francisco, Ca), pp. 445–453, Morgan Kaufmann, 1998.

[27] T. Caudell and D. Newman, "An adaptive resonance architecture to define normality and detect novelties in time series and databases," (Portland, Oregon), pp. 166–176, 1993.

[28] "Mit lincoln labs. 1999 darpa intrusion detection evaluation." In http://www.ll.mit.edu/IST/ideval/index.html, 1999.

[29] J. Gomez and D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection," in *Proceedings of the 2002 IEEE Workshop on Information Assurance*, June 2002.

[30] W. Fan, W. Lee, M. Miller, S. Stolfo, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," in *Proceedings of the first IEEE International conference on Data Mining*, 2001.