An Immuno-Fuzzy Approach to Anomaly Detection

Jonatan Gómez Fabio González Dipankar Dasgupta

***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 definition 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 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].

Approaches inspired on artificial immune systems have been applied successfully to perform anomaly detection on computer network security [9], [10], [11]. However, there are some problems that have prevented this approach from being applied extensively:

- In order to guarantee good levels of detection, a large number of detectors needs to be generated. For some problems the number of detectors could be unmanageable [12]. This problem is aggravated by the binary representation that is being used, in general.
- 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 an the abnormal. This divides the space on 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 defining a degree of normalcy, that is, the set of normal elements is really a fuzzy set.

These issues were addressed with some success in [7]. The mentioned work proposed a technique inspired by the negative selection mechanism of the immune system that can detect foreign patterns in the abnormal (non-self) space. The pattern detectors (in the non-self space) were evolved using a genetic search, which could differentiate varying degrees of abnormality in network traffic. The evolved detectors had a 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.

The work in [8] presented an improvement of the algorithm proposed in the previous work [7]. Specifically, it used a different niching technique to generate the rule detectors. The initial algorithm used a sequential niching technique, whereas the new one used deterministic crowding, which proved to be more efficient on generating good anomaly detectors.

The solution to the crisp distinction between self (normal) and non-self (abnormal) proposed by the mentioned papers is based on dividing the non-self space in different levels. This allowed to estimate the amount of deviation from the normal for a given sample.

This discrete division of the non-self space on levels of deviation can be considered as a previous step to define a real fuzzy characterization of non-self. So, the idea presented in this paper is to extend the previous work [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 NIS. The negative-selection algorithm can be summarized as follows ([14]):

- Define self as a collection S of elements in a feature space U, a collection that needs to be monitored. For instance, if U corresponds to the space of states of a system represented by a list of features, S can represent the subset of states that are considered as normal for the system.
- 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

Jonatan Gómez is with Computer Science Division, Mathematical Sciences Department, University of Memphis, TN 38152 USA (e-mail:jgomez@memphis.edu) and also with Departamento de Ingeniería de Sistemas, Universidad Nacional de Colombia, Ciudad Universitaria, Bogotá, Colombia.

Fabio González is with Computer Science Division, Mathematical Sciences Department, University of Memphis, TN 38152 USA (e-mail:fgonzalz@memphis.edu) and also with Departamento de Ingeniería de Sistemas, Universidad Nacional de Colombia, Ciudad Universitaria, Bogotá, Colombia.

Dipankar Dasgupta is with Computer Science Division, Mathematical Sciences Department, University of Memphis, TN 38152 USA (e-mail:dasgupta@memphis.edu).

more efficient approach will try to minimize the number of generated detectors while maximizing the covering of the non-self space.
- 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 algorithm and it was able to solve anomaly detection problems [9], [15], fault detection problems [16], [17], to detect novelties in time series [1], [18], and even applied to function optimization [19].

In [7], a new version of the negative algorithm was proposed. The main differences with respect to the negative selection algorithm of Forrest et al. [13] are:

- The elements of self/non-self space are represented by n-dimensional real vectors.
- The detectors correspond to hyper-rectangles in $\mathbb{R}^n$ and have a high level representation as 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.

Figure 1 shows an example of the type of coverage generated by this algorithm. The basic structure of these detector rules is as follows:

R^1: If $Cond_1$ then **non_self**
$\vdots$ $\vdots$ $\vdots$
R^k: If $Cond_k$ then **non_self**

where,

- $Cond_i = x_1 \in [low_1^i, high_1^i]$ and $\ldots$and $x_n \in [low_n^i, high_n^i]$
- $(x_1, ..., x_n)$ is a feature vector
- $[low_i^j, high_i^j]$ specifies the lower and upper values for the feature x_i in the condition part of the rule R^j.

The condition part of each rule defines a hyper-rectangle in the feature space ($[0.0, 1.0]^n$). 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. Figure 1 illustrates an example of this kind of cover for $n = 2$.

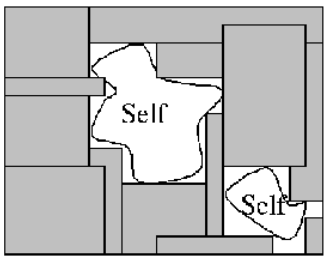

Fig. 1
APPROXIMATION OF THE NON-SELF SPACE BY RECTANGULAR INTERVAL RULES.

This work also proposed a mechanism that allows to estimate the level of deviation from the normal. The non-self space is further divided in different levels of deviation. In Figure 2, these levels of deviation are shown as concentric regions around the self zones. 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.

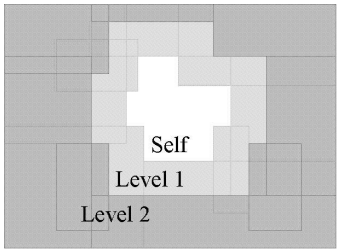

Fig. 2
TWO DIFFERENT SET OF DETECTOR RULES DEFINE TWO LEVELS OF DEVIATION IN THE NON-SELF SPACE.

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

III. PROPOSED APPROACH

The proposed work is a continuation of our effort in improving anomaly detection strategy. 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 after, the use of fuzzy rules improves the accuracy of the method and produces a measure of deviation from the normal that does not need of a discrete division of the non-self space.

A. Anomaly detection with fuzzy rules

The self/non-self space corresponds to $[0.0, 1.0]^n$; therefore, an element x in this space is represented by a vector $(x_1, \ldots, x_n)$ where $\mathrm{x_i} \in [0.0, 1.0]$. 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 being evaluated
T_i: fuzzy set
$\wedge$: fuzzy conjunction operator (in our case, $\min()$)

The fuzzy set T_i is defined by a combination of basic fuzzy sets (linguistic variables). Given a set of linguistic variables $S = \{S_1, \ldots, S_m\}$ 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:

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

The following is an example of a fuzzy detector rule in a self/non-self space with dimension $n = 3$ and using linguistic variables $S = \{S,\ M,\ L\}$:

If $x_1 \in S \wedge x_2 \in (S \cup M) \wedge x_3 \in (M \cup L)$ **then** 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 3 shows an example of such a division using five basic fuzzy sets representing the linguistic variables *Low*, *Medium-Low*, *Medium*, *Medium-High* and *High*.

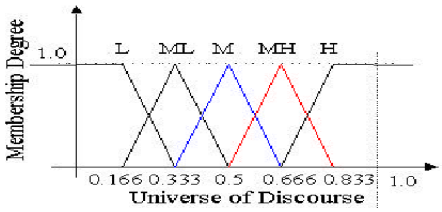

Fig. 3
PARTITION OF THE INTERVAL [0.0,1.0] IN BASIC FUZZY SETS.

Given a set of rules:

R^1: **If** $Cond_1$ **then** non_self
$\vdots$ $\vdots$ $\vdots$,
R^k: **If** $Cond_k$ **then** non_self

where $Cond_i$ corresponds to the condition part of the rule R^i, the abnormality degree of a sample x is defined by

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

where $Cond_i(x)$ represents the fuzzy true value produced by the evaluation of $Cond_i$ in x. $\mu_{\text{non\_self}}(x)$ 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 a genetic algorithm (GA) combined with a 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 [20] as niching technique since it was shown to perform better than sequential niching [21], 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. The algorithm is shown in Figure 4.

```
Initialize population with random individuals;
for j = 1 to numGenerations
  for k = 1 to population_size/2
    Select two individuals with uniform probability
     and without replacement;
    Apply crossover to generate a child;
    Mutate the child;
    if dist(child,parent1) < dist(child,parent2)
      and fitness(child) > fitness(parent1)
       Substitute parent1 with child;
    elseif dist(child,parent1) ≥ dist(child,parent2)
      and fitness(child) > fitness(parent2)
       Substitute parent2 with child;
    endif
  endfor
endfor
Extract the best individuals from the population;
```

Fig. 4
GENETIC ALGORITHM TO EVOLVE FUZZY RULE DETECTORS

1) Chromosome representation: Each individual (chromosome) in the genetic algorithm represents the condition part of a rule, since the consequent part is the same for all the rules (the sample belongs to non-self). As it was described before, a 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 variables) 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 5 shows the structure of the chromosome which is $n \times m$ bits long (n is the dimension of the space and m is the number of basic fuzzy sets).

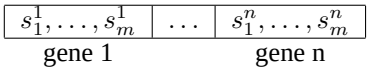

Fig. 5
STRUCTURE OF THE CHROMOSOME REPRESENTING THE CONDITION PART OF A RULE. EACH GENE REPRESENTS AN ATOMIC CONDITION $x_i \in T_i$ AND EACH BIT s_j^i IS 'ON' IF AND ONLY IF THE CORRESPONDING BASIC FUZZY SET S_j IS PART OF THE COMPOSITE FUZZY SET T_j.

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

- The fuzzy true value produced when the condition part of the rule, $Cond_i$, is evaluated for each element x from the self set: is a number of elements in the training set S, that belongs to the subspace represented by the rule:

$$selfCovering(R) = \frac{\sum_{x \in Self} Cond_i(x)}{|Self|}$$

- The fuzzy measure of the volume of the subspace repre-

sented by the rule:

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

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

The fitness is defined as:

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

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) Individual's Distance Calculation: A good measure of distance between individuals is important for deterministic crowding niching, since it allows the algorithm to replace individuals with closer individuals. This allows the algorithm to preserve the and form niches.

In this work we use the Hamming distance, because there is a strong relation between each single bit in the chromosome with a single fuzzy set of some particular attribute of the search space. For example, if the s_i^j bit (see Figure 5), 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, the more bits the parent and the child have in common, the more 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 with three different data sets as shown in table I. For determining the scalability of the proposed approach, each individual performs a random sampling of the training set. The size of the sampling was fixed to 400 data elements. 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**), which is an optimization algorithm based on random mutations of potential solutions population. The algorithms were run 1000 iterations with a population size of 200 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.

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 a ROC curve [22] for each of the algorithm tested. Also the reported DR is the detection rate obtained for each algorithm when the FA was fixed to 3%.

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

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. The equation is:

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

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], [23]. 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 $n = 4$. Five fuzzy sets, as shown in figure 3, were defined for each feature extracted.

2) Results and Analysis: The proposed approach performs better than the other two tested methods, see Figure 6. Moreover, the PHC reaches a better performance than the ERD algorithm. This behavior can be attributed to the fuzzyfication of the search space, because the fuzzy rule detectors provide a better characterization of the normal-abnormal boundaries.

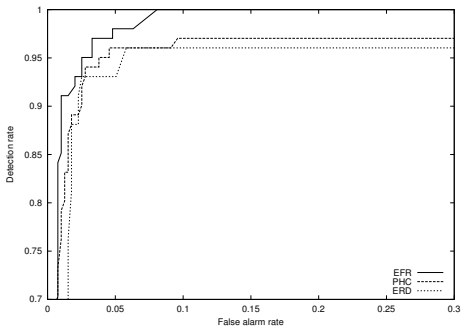

Fig. 6
ROC CURVES GENERATED BY THE THREE ALGORITHMS TESTED WITH THE MACKEY GLASS DATA SET

Table II compares the performance of the tested algorithms over the Mackey-Glass data set. When the FA rate is fixed to 3%, the proposed approach is able to detect a higher percentage of abnormal samples (row 1) than the other two approaches (rows 2 and 3). Also, the number of fuzzy rule detectors (rows 1 and 2) are considerably small compared with the number of crisp detectors (row 3). Therefore, the fuzzyfication of the

search space allows a simple characterization of the abnormal (non-self) space.

TABLE II
COMPARATIVE PERFORMANCE IN THE MACKEY-GLASS PROBLEM

Algorithm	DR%	# Detectors
EFR	**95.05**	**14**
PHC	94.06	32
ERD	93.07	78

In addition, the EFR algorithm is able to generate a more compact representation of the abnormal space than the PHC algorithm. Clearly, applying crossover along with a crowding strategy generates better fuzzy rule detectors than those produced by applying only mutation. Moreover, the EFR and PHC algorithms were tested using 15 fuzzy sets instead of 5, in order to determine their performance using a high fuzzy resolution (in this case, the size of the chromosome is three times the size of the original chromosome). Figure 7 compares the ROC curves for the EFR and PHC using 5 and 15 fuzzy sets per feature.

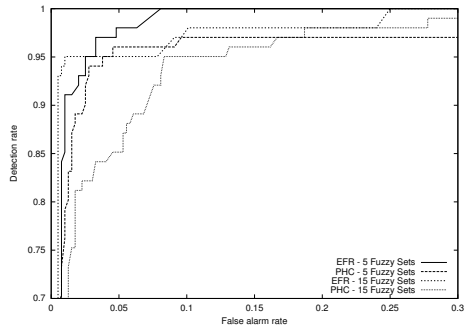

Fig. 7
ROC CURVES GENERATED BY EFR AND PHC WITH 5 AND 15 FUZZY SETS PER FEATURE

As expected, the performance of the proposed approach (EFR) using 15 fuzzy sets is better than the parallel hill climbing (PHC) using 15 fuzzy sets too. When the fuzzy resolution (number of fuzzy sets) is increased, the performance of the PHC decreases drastically while the performance of the EFR decreases smoothly. Then the performance of PHC is strongly affected by the dimensionality of the search space (size of the chromosome); this can be an evidence that mutation alone is not enough to find good solutions in this high dimensional search space. Also, the EFR performance remains comparable with the PHC performance using five fuzzy sets.

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 [24]. Experiments were conducted on 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).

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. An 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.

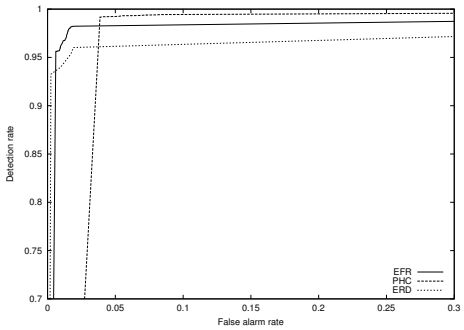

Fig. 8
ROC CURVES GENERATED BY THE THREE ALGORITHMS TESTED WITH THE KDD-CUP 99 DATA SET

2) Results and Analysis: The performance reached by the PHC and EFR algorithms are almost the same while are better than the performance reached by ERD, see Figure 8. Table III compares the performance of the tested algorithms and some results reported in the literature. The FA-DR reported in table III is the closest value to the optimal point (0,1). 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).

TABLE III
COMPARATIVE PERFORMANCE IN THE KDD CUP 99 PROBLEM

Algorithm	DR%	FA%	# Detectors
EFR	**98.22**	**1.9**	14
PHC	99.17	3.9	32
ERD	96.02	1.9	699
EFRID[25]	98.95	7.0	-
RIPPER-AA[26]	94.26	2.02	-

According to table III, the performance of EFR is comparable with the performance of approaches reported in the literature and in many cases performs better. For example, when EFR is compared with RIPPER-AA the detection rate is almost the

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

same (close to 2%) but EFR has a higher DR (4% more abnormal samples detected). Now, compared with the crisp approach (ERD) the performance is also superior (2.2% more abnormal samples detected). Clearly, the fuzzy characterization of the abnormal space reduces the number of false alarm while the detection rate is increased.

Besides the detection rate reached by the PHC algorithm is higher than the reached by EFR, the false alarm rate is also higher (3.9%) than in the EFR (1.9%). Also, the number of fuzzy rules detectors generated by PHC is big (32) compared with the generated by EFR (12).

C. Darpa 99

This data set, was also obtained from the MIT-Lincoln Lab [24]. 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.

2) Results and Analysis: The performance reached by the PHC and EFR algorithms are almost the same while are better than the performance reached by ERD, see Figure 9. 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.

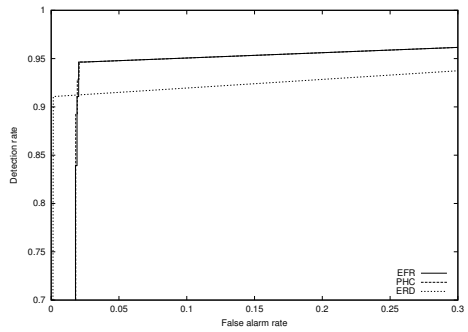

Fig. 9
ROC CURVES GENERATED BY THE THREE ALGORITHMS TESTED WITH THE DARPA 99 DATA SET

compares the performance of the tested algorithms over the Darpa 99 data set. The EFR and PHC algorithms (both of them based on fuzzy rules detectors), outperformed the ERD algorithm (based on a crisp characterization), see Table IV. Those methods increase the DR in at least 5%. The performance reached by EFR and by PHC are almost the same, but the number of fuzzy rules detectors generated by EFR is lower than the generated by PHC. In this way the proposed approach generates a simpler characterization of the abnormal space than PHC does.

TABLE IV
COMPARATIVE PERFORMANCE IN THE DARPA 99 PROBLEM

Algorithm	DR%	# Detectors
EFR	**94.63**	**7**
PHC	94.63	9
ERD	89.37	35

V. CONCLUSIONS AND FUTURE WORK

This paper presented a new technique that allows to generate a set of fuzzy rules that characterize the non-self space (abnormal) using as input only self (normal) samples. This work extended a 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. The following are the main advantages of the new approach:

- It provides a better definition of the boundary between normal and abnormal. The previous approach used a discrete division of the non-self space, whereas the new approach does not need such a a division since the fuzzy character of the rules provide a natural estimate of the amount of deviation from 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.

Our future work will explore the application of more advanced genetic algorithm representations such as structured GA [27] and perform a more extensive testing with other real data sets.

VI. 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).

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, 200.

[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. 1081–1088, 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*, 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), 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*, (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 *Special sessions on artificial immune systems in the 2002 congress on evolutionary computation, 2002 ieee world congress on computational intelligence*, (Honolulu, Hawaii), 2002.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] M. Labs, "Darpa intrusion detection evaluation." http://www.ll.mit.edu/IST/ideval/index.html, 1999.

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

[26] 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.

[27] D. Dasgupta and D. McGregor, "A more biologically motivated genetic algorithm: The model and some results," *Cybernetics and Systems: An International Journal*, vol. 25, no. 3, pp. 447–469, 1994.