Proceedings of the 2002 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY June 2001

Evolving Fuzzy Classifiers for Intrusion Detection

Jonatan Gomez and Dipankar Dasgupta

Abstract – The normal and the abnormal behaviors in networked computers are hard to predict as the boundaries cannot be well defined. This prediction process may generate false alarms in many anomaly based intrusion detection systems. However, with fuzzy logic, the false alarm rate in determining intrusive activities can be reduced; a set of fuzzy rules (non-crisp fuzzy classifiers) can be used to define the normal and abnormal behavior in a computer network, and a fuzzy inference algorithm can be applied over such rules to determine when an intrusion is in progress. The main problem with this approach is to generate good fuzzy classifiers to detect intrusions. This paper proposes a technique to generate fuzzy classifiers using genetic algorithms that can detect anomalies and some specific intrusions. The main idea is to evolve two rules, one for the normal class and other for the abnormal class using a profile data set (a preprocessed DARPA data set is used [1]) with information related to the computer network during the normal behavior and during intrusive (abnormal) behavior. This paper exhibits some results and reports the performance of evolved fuzzy classifiers in intrusion detection.

Index terms – Intrusion detection, fuzzy classification, rule generation, and genetic algorithms

I. INTRODUCTION

The number of intrusions into computer systems is growing because new automated intrusion tools appearing every day, and these tools and different system vulnerability information are easily available on the web. These intrusions can come from inside (insider or legal users) or outside (outsider users) the system. Heady in [2] defined an intrusion as:

Any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource.

The problem of intrusion detection has been studied extensively in computer security ([3], [4], and [5]), and has received a lot of attention in machine learning and data mining ([6], [7], and [8]). The problem of intrusion detection can be stated as follows: detect when a computer system is being attacked or an intrusion is in progress. Basically, there are two models of intrusion detection [5]:

Anomaly Detection: Known and unknown intrusions are detected by analyzing changes in the normal pattern of utilization or behavior of the computer system. This approach does not use information about the system behavior when an intrusion is in progress.

Signature or Misuse Detection: Known intrusions are detected by looking at the computer system behavior some characteristic pattern of such intrusions. This approach uses some collected information about the system behavior under normal conditions and under some known intrusions to determine the current state of the system. In this case, the intrusion detection problem is a classification problem.

There are many approaches that use one or the other above-mentioned model to solve the intrusion detection problem:

1. Using data mining techniques over system audit data to extract consistent and useful patterns of program and user behavior, and to build classifiers that can recognize anomalies [8]. The basic data mining techniques used are the classical association rules and the frequent episodes.

2. Using temporal association rules (a data mining technique that uses time concepts), in terms of multiple time granularities [9]. The temporal association rules technique generates fuzzy and classical rules [10].

3. Using short sequences of system calls that running programs perform as discriminators between normal and abnormal operating characteristics [11]. The discriminator uses the Hamming distance as a distance function between short sequences of system calls. If the distance of a particular sequence to the normal sequences is higher than a threshold then the sequence is abnormal.

4. Distributing the detection task in multiple independent entities (autonomous agents) working collectively [6]. The functionality of each agent is not defined but it can be simple or complex according to the specific detection task that an agent is assigned. If some agent detects some anomalies or intrusions, the agent sends messages to other

Intelligent Security System Research Lab. Division of Computer Sciences. Mathematical Sciences Department, The University of Memphis. Memphis, TN 38152.

ISBN 555555555/$10.00

agents to define, in a distributed manner, the corresponding action to take.

5. Using genetic programming to build autonomous agents that detect intrusions. The learning model uses feedback; and the process evaluates evolved agents over the scenario of intrusions and normal behavior [7].

6. Emulating mechanisms of the natural immune systems to detect anomalies in a distributed manner [12]. It combines two anomaly detection methods: using profiles of user behavior and correlation of user behavior with network statistical behavior. The decision support component uses an ART neural network and a Fuzzy Controller.

In this paper, we show the applicability of genetic algorithms to evolve a simple set of fuzzy rules (fuzzy classifier) that can solve some well-studied intrusion detection problems. In this approach, genetic algorithms can find good and simple fuzzy rules to characterize intrusions (abnormal) and normal behavior of network systems. As the difference between the normal and the abnormal activities are not distinct, but rather fuzzy, fuzzy logic can reduce the false signal rate in determining intrusive activities.

The subsequent sections are organized as follows. Section 2 briefly describes the basic fuzzy logic and fuzzy classifiers concepts used in this paper, section 3 presents the proposed approach to solve some intrusion detection problems, section 4 describes experiments and analysis of results, and section 5 draws some conclusions.

II. FUZZY CLASSIFIERS FOR INTRUSION DETECTION

The intrusion detection problem (IDP) is viewed in the misuse or signature model as a two-class classification problem: the goal is to classify patterns of the system behavior in two categories (normal and abnormal), using patterns of known attacks, which belong to the abnormal class, and patterns of the normal behavior. With fuzzy rules, the solution of this classification problem is based on fuzzy logic concepts.

A. Fuzzy Logic

In fuzzy logic fuzzy sets define the linguistic notions and membership functions define the truth-value of such linguistic expressions. Table 1 shows the difference between classic sets and fuzzy sets.

FUZZY SETS	CLASSIC SETS
In fuzzy sets an object can partially be in a set.	In classic sets an object is entirely in a set or is not.
The membership degree takes values between 0 and 1.	The membership degree takes only two values 0 or 1.
1 means entirely in the set, 0 means entirely not in the set, other values mean partially in the set.	1 means entirely in the set, 0 means entirely not in the set. Other values are not allowed.

Table 1: Comparisson between fuzzy sets and classic sets

The membership degree to a fuzzy set of an object defines a function where the universe of discurse (set of values that the object can take) is the domain, and the interval [0,1] is the range. That function is called the membership function. Figure 1 shows the most used membership function, the triangular membership function:

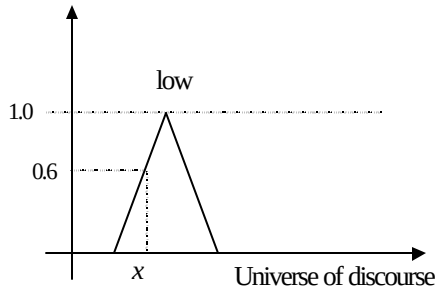

Figure 1: Triangular membership function for a fuzzy set

In figure 1, the object *x* has 0.6 degree of membership to the fuzzy set *low*, i.e., *x* does not entirely belong to the set *low*, but *x* belongs to the fuzzy set and does not belong to the set at the same time. A collection of fuzzy sets, called fuzzy space, defines the fuzzy linguistic values or fuzzy-classes that an object can belong to. A standard fuzzy space is shown in figure 2.

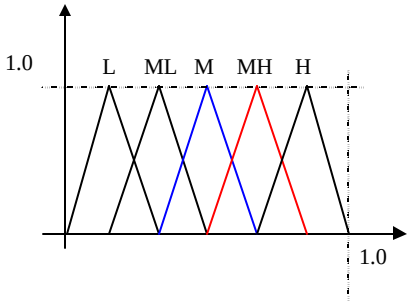

Figure 2: Fuzzy space of five fuzzy sets

With fuzzy spaces, fuzzy logic allows an object to belong to different classes at the same time. This concept is helpful when the difference between classes is no well defined. It is the case in the intrusion detection task, where the difference between the normal and abnormal class are not well defined.

With these linguistic concepts, atomic and complex fuzzy logic expressions can be built. An atomic fuzzy expression is an expression:

parameter is [not] fuzzyset

ISBN 555555555/$10.00

Where, *parameter* is an object, and *fuzzyset* is a fuzzy set that belongs to the defined fuzzy space for the parameter. The truth-value (TV) of an atomic expression is the degree of membership of the parameter to the fuzzy set. Because TVs are expressed by numbers between 0 and 1, (0 means entirely false, 1 means entirely true, and others values means partially true), the fuzzy expression evaluation process is reduced to arithmetic operations. Also, for each classical logic operator (and, or, negation), there is a common fuzzy logic arithmetic operator (shown in table 2):

LOGIC OPERATOR	FUZZY OPERATOR
p AND q	*min{p, q}*
P OR q	*max{p, q}*
NOT p	*1.0-p*

Table 2: Fuzzy logic operators

Fuzzy rules have the form:

IF condition *THEN* consequent [weight]

Where,

- *condition* is a complex fuzzy expression, i.e., that uses fuzzy logic operators and atomic fuzzy expressions
- *consequent* is an atomic expression, and
- *weight* is a real number that defines the confidence of the rule.

The following an example of a fuzzy rule:

R: IF x is HIGH and y is LOW THEN pattern is normal [0.4]

The fuzzy rule truth-value is calculated as the product the condition truth-value by the weight, i.e.:

$$TV(R) = TV(\text{condition}) * weight$$

For the previous example, if the degree of membership of the parameter *x* to the fuzzy set *HIGH* is *0.2*, the degree of membership of *y* to *LOW* is *0.4* and the weight is *0.4* then the truth-value of the fuzzy rule is:

$$TV(R) = TV(x \text{ is HIGH and } y \text{ is LOW}) * 0.4 = \min\{0.2, 0.4\} * 0.4 = 0.2 * 0.4 = 0.08$$

B. Fuzzy classifiers and the two classes classification problem

In the two classes classification problem, there are two classes where every object should be classified. These classes are called positive (abnormal) and negative (normal). The data set used by the learning algorithms consists of a set of objects, each object with *n+1* attributes. The first *n* attributes define the object characteristics (monitored parameters) and the last attribute defines the class that the object belongs to (the classification attribute).

A fuzzy classifier for solving the two class classification problem is a set of two rules, one for the normal class and other for the abnormal class, where the condition part is defined using only the monitored parameters and the conclusion part is an atomic expression for the classification attribute. For example,

R_N *: IF x is HIGH and y is LOW THEN pattern is normal [0.4]*
R_A *: IF x is MEDIUM and y is HIGH THEN pattern is abnormal [0.6]*

Is a fuzzy classifier for the two classes classification problem. There are several techniques to determine the class that an object belongs to. One of these techniques is the maximum technique, which classifies the object as the class in the conclusion part of the rule that has the maximum truth-value, i.e.:

$$class = \begin{cases} N & if\ TV(R_N) > TV(R_A) \\ A & if\ TV(R_N) < TV(R_A) \end{cases}$$

Where,
N represents the normal class,
A the abnormal class,
R_N is the rule for the normal class, and
R_A is the rule for the class abnormal class

If the two rules produce the same truth-value, one class can be picked randomly.

III. EVOLVING FUZZY CLASSIFIERS

We used a genetic algorithm to generate fuzzy classifiers for intrusion detection using datasets with patterns of the system behavior during normal and under intrusive (abnormal) conditions.

A genetic algorithm is the computational equivalent of the natural evolutionary process. In a genetic algorithm a set of chromosomes (population), each chromosome codifying a possible solution for the given problem, is evolved using a set of genetic operators (mutation, crossover, selection). Each chromosome has probability to be used by one of the genetic operators, and this probability depends on the adaptability of the chromosome (efficiency of the organism to solve the given problem).

There are different approaches to evolve fuzzy classifiers with genetic algorithms [13], [14] y [15]. We used the approach proposed in [16], where, a free-parameters genetic algorithm with special operators (gene addition, gene deletion), is used for each class (normal and

ISBN 555555555/$10.00

abnormal). In this way, the genetic algorithm for the normal class, tries to find a fuzzy rule:

Rule: IF condition$_{normality}$ THEN pattern IS normal [0.5]

Because, the variable element in this fuzzy rule is the condition part (the consequent and the confidence weight are fixed), only the condition part is encoded as a linear chromosome, with variable length that uses precedence values in the logic operators. For example, the expression:

(x is C Ù w is not D) Ú z is E

Can be represented without parenthesis and using complete expression trees as:

x is C Ú z is E Ù w is not D

With complete expression tree, the chromosome is defined as a set of *n* genes, each gene is composed of an atomic condition <variable> is [not] <set> and a logic operator, as is shown in figure 3.

Gen_1				...	Gen_n				Gen_{n+1}				
ac_1			op_1	...	can			op_n	ac_{n+1}				*
var_1	ro_1	set_1		...	var_n	ro_n	set_n		var_{n+1}	ro_{n+1}	set_{n+1}		*

Figure 3: Representation of the condition part of a fuzzy rule using operator precedences

An example expression encoded in the chromosome using operator priority is as follows:

Gen_1				Gen_2				Gen_3					
Ac_1			op_1	Ac_2			op_2	ac_3				op_3	
X	YES	C	∨	Z	YES	E	∧	W	NOT	D		*	*

Figure 4: Codification of the expression X is C or Z is E and W is not D

The authors in [16] used the fuzzy confusion matrix to calculate the fitness of a chromosome. In the fuzzy confusion matrix the fuzzy truth degree of the condition represented by the chromosome and the fuzzy negation operator are used directly. In our case, the fitness of a chromosome for the normal class is evaluated according to the following set of equations:

$$TP = \sum_{i=1}^{p} predicted\ (normal\ \_data_i)$$

$$TN = \sum_{i=1}^{p} [1 - predicted(abnormal\ \_data_i)]$$

$$FP = \sum_{i=1}^{p} predicted\ (abnormal\ \_data_i)$$

$$FN = \sum_{i=1}^{p} [1 - predicted\ (normal\ \_data_i)]$$

$$sensitivity = \frac{TP}{TP+FN}, specificity = \frac{TN}{TN+FP}$$

$$length = 1 - \frac{chrom\ \_length}{10},$$

$$fitness = w_1 * sensitivity + w_2 * specificity + w_3 * length$$

Here,

TP, *TN*, *FP*, and *FN* are the true positive, true negative, false positive, and false negative values for the codified rule respectively
predicted is the fuzzy value of the condition part of the codified rule
p and *q* are the number of normal and abnormal samples in the training data set used by each chromosome respectively,
w_1, w_2, and w_3 are the assigned weights for each rule characteristic respectively,
$normal\_data_i$ is the subset of normal training patterns, and,
$abnormal\_data_i$ is the subset of abnormal training patterns.

The set of equations to calculate the fitness for the abnormal class can be obtained by changing abnormal for normal in previous equations. The best chromosome in the population is chosen and the fuzzy rule: *if <condition> then pattern is <class>*, is added to the fuzzy classifier. Here, *<condition>* is the condition represented by such chromosome, and *<class>* is the class pattern evolved by the genetic algorithm.

IV. EXPERIMENTATION

A. Test Data Sets

In order to evaluate the performance of the proposed approach to generate comprehensible fuzzy classifiers for intrusion detection, tests were conducted using the ten percent of the kdd-cup'99 data set, [1]. This data set is a version of the 1998 DARPA intrusion detection evaluation data set prepared and managed by MIT Lincoln Labs [18]. In this data set, forty-two attributes (or fields) that usually characterize network traffic behavior compose each record. Also, the number of records in this data set is 494021.

B. Preprocessing

We applied two preprocessing algorithms to the original ten percent KDD-cup99 data set: Uniform distribution by pattern class and normalization of numerical attributes.

ISBN 555555555/$10.00

The uniform distribution algorithm creates a data set from the original data set with the following property: If the samples number of pattern *k* is *m* and the original data set has *n* samples, then the probability to find a sample of class *k* in the first *n/m* samples of the final data set is *1.0*. Therefore each portion of the final data set has almost the same distribution of the full data set.

In the normalization algorithm each numerical value in the data set is normalized between 0.0 and 1.0 according to the following equation:

$$x = \frac{x - MIN}{MAX - MIN}$$

Where,
x is the numerical value,
MIN is the minimum value for the attribute that *x* belongs to, and
MAX is the maximum value for the attribute that *x* belongs to.

For each numerical attribute we assign the following fuzzy space:

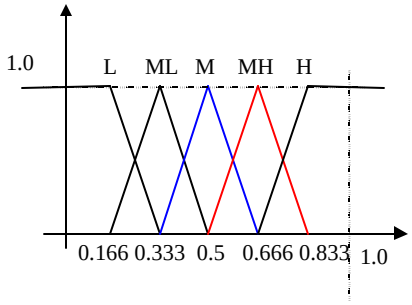

Figure 5 Fuzzy space for numerical attributes in the KDD-cup99data set

For non-numerical attributes like logged-in we used the categorical values as crisp sets (fuzzy sets that does not overlapping each other). Figure 6 shows as example the fuzzy space associated to the logged-in attribute. Therefore a value of false for this attribute has a degree of membership to the crisp set FALSE equal to 1.0 and degree of membership to the fuzzy set TRUE equal to zero.

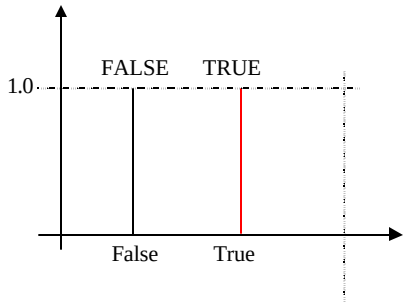

Figure 6 Fuzzy space for the non-numerical attribute logged-in

C. Experimental Setting

A five-fold testing was employed [19]. That is, the training-test data set is divided randomly in five groups, each group was taken as testing set for the fuzzy classifier trained by the genetic algorithm with the others four groups. We repeated the process five times and the score of the trained classifier was calculated as the average of the twenty-five test applied.

We used the free parameters genetic algorithm proposed in [17], each genetic algorithm was initialized with a random chromosomes population of 200 individuals, with length between one and six genes, and maximum number of iterations fixed at 200. We performed several test with different values for the fitness function weights. The reported results were obtained using random assignation of the weights as is explained in [16], these values showed good performance in the evolution of fuzzy classifiers for the intrusion detection problem.

Also, we fixed the number of records that each chromosome in the population could use in 1% of the original data set size. Because we applied the uniform distribution algorithm we ca be sure that each chromosome is using almost the same type of records, but not all the data set. This allowed to us increase the speed of the genetic algorithm with huge data sets.

D. Results and Analysis

There are two elements that define the cost function of an intrusion detection system: the false alarm rate (the system produces an alarm in a normal condition), and the undetected attacks rate (the system considers an abnormal behavior as normal). The average performance of the proposed approach (Evolving Fuzzy Rules for Intrusion Detection EFRID) over the twenty-five test performed is shown in table 3.

ISBN 555555555/$10.00

FALSE ALARMS	DETECTION RATE
10.63%	95.47%

Table 3: Average performance reached by EFRID

Table 4 compares the performance of EFRID against different methods found in the literature. As is shown in this table our approach compare well with such methods.

Algorithm	FA %	DR %	O(n)
EFRID	10.63	95.47	347.19n

Table 4: Comparisson of the proposed approach

In EFRID, each individual in the population only used 1% of the data-set, the number of individuals per iteration is 200, the number of iterations is 200, and the genetic algorithm is run 2 times (one per each class normal-abnormal) then the number of times that the data set is used is bounded by 800 times. Therefore EFRID is a linear algorithm respect to the size of the data set. But because there are some individuals per iteration that are not feasible (the result of the genetic operator can produces not valid fuzzy rules), we calculated the average number of times that the data set is read.

Because the intrusion detection problem is a two-class classification problem, where the positive class is the abnormal class and the negative class is the normal class, we applied the Receiver Operating Characteristic (ROC) analysis to evaluate the performance of the evolved classifiers [20]. In the ROC analysis, for classifier systems that produce a continuous output with respect to some parameter α, the coordinate point $(FP, TP)_\alpha$ is plotted in the coordinate system. Here, TP is the true positive rate (the percentage of abnormal behavior (intrusions) classified as abnormal) and FP is the false positive rate (the rate of false alarms).

It is possible to generate three different ROC curves for the intrusion detection problem from the evolved fuzzy classifiers:

1. Using only the fuzzy rule for the normal class and varying a threshold (α) for the truth-value of the rule, between 0.0 and 1.0
2. Using only the fuzzy rule for the abnormal class and varying a threshold (α) for the truth-value of the rule, between 0.0 and 1.0
3. Because there are two rules, it is possible to assign to each rule the confidences 1-α and α, varying α between 0.0 and 1.0.

The plotted points define the ROC curve for the given classifier. This ROC curve can be used to determine when a classifier is better than other. If the ROC curve of a classifier *A* dominates the ROC curve of the classifier *B* then classifier *A* is better than classifier *B* [20]. The ROC curve for an intrusion detection classifier shows how the fuzzy rule confidence value affects the rate of alarms and the rate of detected abnormal activities.

Because a five folding cross-validation was applied, we generated the average ROC curve, with the strategy explained in [21], i.e., for each of the five folds, the ROC curve is treated as a function TP = R(FP), it is done with linear interpolations, and the average ROC curve is the mean of those functions. The average **ROC** curve for the five repetitions is obtained in the same way. The average ROC curves for the evolved fuzzy classifier systems are shown in figure 7.

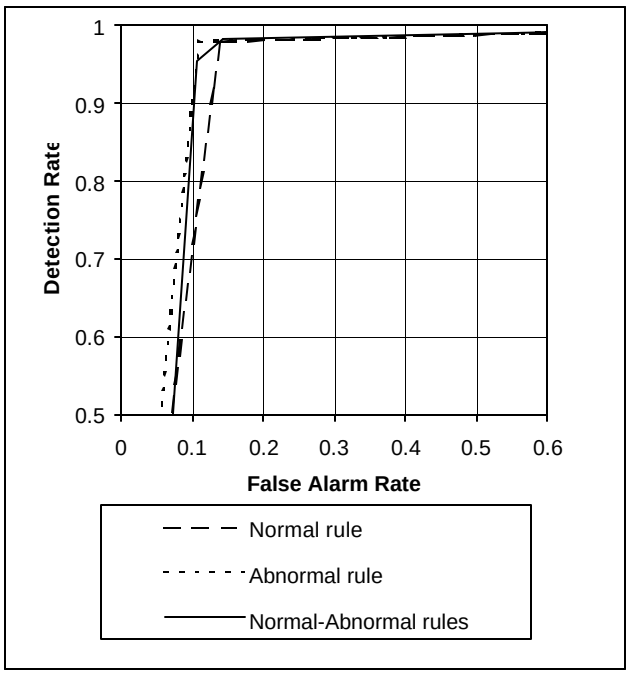

Figure 7: Average ROC curve for evolved fuzzy classifiers

According to the ROC curves only using the fuzzy rule for the abnormal class produces the best results (lower false alarm rate with a higher detection rate), i.e., the fuzzy rule for the normal class has an important effect: it reduces the detection rate of anomalies, without reducing the false alarm rate.

V. CONCLUSIONS

Our experiments showed that the proposed approach works well in detecting different attacks. The accuracy of the fuzzy classifier was good and comparable to those reported in the literature. Also, the accuracy can be further

ISBN 555555555/$10.00

improved applying specific strategies to generate the fuzzy space for each monitored parameter.

The evolved fuzzy rules are not complex as no more than six attributes are used in each rule. It allows characterization of the normal and abnormal behaviors in human words.

In order to reduce the dimensionality of the problem, several statistical methods can be applied before the evolution process is performed.

The main contribution of the present work is the design of a classification process for the intrusion detection problem. It allows apply fuzzy logic and genetic algorithms for the detection of various types of attacks.

VI. ACKNOWLEDGES

This work was supported by the Defense Advanced Research Projects Agency (no. F30602-00-2-0514) and National Sciences Foundation (no. NSF-EIA-9818323).

VII. REFERENCES

[1] KDD-cup data set. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2] R. Heady, G. Luger, A. Maccabe, and M. Sevilla. The Architecture of a Network-level Intrusion Detection System, Technical report, CS90-20. Dept. of Computer Science, University of New Mexico, Albuquerque, NM 87131.
[3] Edward Amoroso, "Intrusion detection", Intrusion.net Books, January 1999.
[4] Julia Allen at all, "State of the practice of intrusion detection technologies", Technical Report CMU/SEI99 - TR-028, ESC-99-028, Carnegie Mellon, Software Engineering Institute, Pittsburgh, Pennsylvania, 1999.
[5] Stefan Axelsson, "Intrusion detection systems: A survey and taxonomy", Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.
[6] Jai Sundar at all, "An architecture for intrusion detection using autonomous agents", Tech. Rep. 98/05, Purdue University, 1998.
[7] Mark Crosbie, "Applying genetic programming to intrusion detection", In Proceedings of the AAAI 1995 Fall Symposium series, November 1995.
[8] Wenke Lee at all, "Mining audit data to build intrusion detection models", Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD'98), pages 66-72, 1998.
[9] Yingjiu Li at all, "Enhancing profiles for anomaly detection using time granularities", Center for secure information systems. To appear in Journal of Computer Security, 2002.
[10] Susan Bridges and Rayford Vaughn, “Fuzzy data mining and genetic algorithms applied to intrusion detection”, Proceedings twenty third National Information Security Conference, October 1-19, 2000.
[11] Steve Hofmeyr at all, "Intrusion detection using sequences of systems call", Journal of Computer Security, 6:151-180, 1998.
[12] Dipankar Dasgupta and Hal Brian, "Mobile security agents for network traffic analysis", Published by the IEEE Computer Society Press in the proceedings of DARPA Information Survivability Conference and Exposition II (DISCEX-II), June 12-14, 2001, Anaheim, California.
[13] Bojarczuk C.E., Lopes H.S. y Freitas A.A. “Discovering comprehensible classification rules using genetic programming: a case study in medical domain”. Proceedings Genetic and Evolutionary Computation Conference GECCO99, 1999.
[14] Ishibuchi H. y Nakashima T. “Linguistic rule extraction by genetic-based machine learning”. Proceedings Genetic and Evolutionary Computation Conference GECCO00, 2000.
[15] Liu J. y Kwok J. “An extended genetic rule induction algorithm”. Proceedings of the Congress on Evolutionary Computation Conference, 2000.
[16] J. Gomez, F. Gonzalez, and D. Dasgupta, “Complete Expression Trees for Evolving Fuzzy Classifier Systems with Genetic Algorithms”, Submitted to the Evolutionary Computation Conference GECCO02, 2002
[17] J. Gomez, D. Dasgupta, and R. Kozma, “Using Competitive Operators and a Local Selection Scheme in Genetic Search”, Submitted to the Evolutionary Computation Conference GECCO02, 2002
[18] Lincoln Laboratory MIT. http://www.ll.mit.edu/
[19] T. Lim and W. Loh. "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Technical Report, Department of Statistics, University of Wisconsin-Madison, No. 979, 1997.
[20] Foster Provost, et all, “Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions”, Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 1997.
[21] Foster Provost, et all, “The case against accuracy estimation for comparing induction algorithms”, Proceedings of the Fifteenth International Conference on Machine Learning, 1998.

ISBN 555555555/$10.00