Computer Intrusion Detection Using an Iterative Fuzzy Rule Learning Approach

Mohammad Saniee Abadeh and Jafar Habibi

Abstract—**The process of monitoring the events occurring in a computer system or network and analyzing them for sign of intrusions is known as intrusion detection system (IDS). The objective of this paper is to extract fuzzy classification rules for intrusion detection in computer networks. The proposed method is based on the iterative rule learning approach (IRL) to fuzzy rule base system design. The fuzzy rule base is generated in an incremental fashion, in that the evolutionary algorithm optimizes one fuzzy classifier rule at a time. The performance of final fuzzy classification system has been investigated using intrusion detection problem as a high-dimensional classification problem. Results show that the presented algorithm produces fuzzy rules, which can be used to construct a reliable intrusion detection system.**

I. Introduction

An intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [1]. Intrusion Detection Systems (IDS) are effective security tools, placing inside a protected network and looking for known or potential threats in network traffic and/or audit data recorded by hosts. Basically, an IDS analyzes information about users' behaviors from various sources such as audit trail, system table, and network usage data.

The problem of intrusion detection has been studied extensively in computer security [3]-[6], and has received a lot of attention in machine learning and data mining [7]-[9].

Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Signature or misuse detection is based on patterns of known intrusions [10]-[12]. In this case, the intrusion detection problem is a classification problem. This approach allows the detection of intrusions which the system has learned their signatures perfectly. To remedy the problem of detecting novel attacks, anomaly detection attempts to construct a model according to the statistical knowledge about the normal activity of the computer system [13]–[15].

The above discussion points out that the tradeoff between the ability to detect new attacks and the ability to generate a low rate of false alarms is the key point to develop an effective IDS. In this paper, we exploit a new evolutionary fuzzy system to develop an IDS based on misuse detection. The goal of our algorithm is to find high quality fuzzy if-then rules to predict the class of input patterns correctly.

Evolutionary algorithms (EA) have been used as rule generation and optimization tools in the design of fuzzy rule-based systems [16, 17]. Those EA-based studies on the design of fuzzy rule-based systems are usually referred to as Evolutionary Fuzzy Systems (EFS), each of which can be classified into the Michigan, Pittsburgh or Iterative Rule Learning (IRL) approaches [16].

Some studies are categorized as the Michigan approach where a single fuzzy if–then rule is coded as an individual [11, 18]. Many fuzzy EFS methods are categorized as the Pittsburgh approach where a set of fuzzy if–then rules is coded as an individual [19, 20]. In the third approach, the iterative one, chromosomes code individual rules, and a new rule is adapted and added to the rule set, in an iterative fashion, in every run of the GA [10, 21, 25].

In this paper, we have extended our previous Michigan-based intrusion detection algorithm [11] from a problem with two classes to a five-class classification problem. To accomplish this purpose we have used an IRL-based evolutionary fuzzy system that learns the final fuzzy classification rule set in an iterative fashion. The proposed evolutionary fuzzy system has been tested using the public KDD CUP'99 intrusion detection data set available at the University of California, Irvine web site [22]. As our proposed classification system is an IRL-based evolutionary fuzzy system for computer intrusion detection, we call it CID-IRL through the rest of the paper.

The rest of the paper is as follows: Fuzzy rule base for pattern classification is presented in section II. The proposed IRL-based evolutionary fuzzy system is discussed in Section III. Experimental results are reported in Section IV. Section V is conclusions.

II. Fuzzy Rule Base for Pattern Classification

Let us assume that our pattern classification problem is a c-class problem in the n-dimensional pattern space with continuous attributes. We also assume that M real vectors $x_p = (x_{p1}, x_{p2}, ..., x_{pn}), p = 1, 2, ..., M$, are given as training patterns from the c classes ($c << M$).

Because the pattern space is $[0,1]^n$, attribute values of each pattern are $x_{pi} \in [0,1]$ for $p = 1, 2, ..., M$ and $i = 1, 2, ..., n$. In computer simulations of this paper, we normalize all attribute values of each data set into the unit interval $[0,1]$.

All of the authors are with the Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran, Iran (phone: +98 2166164636; email: saniee@ce.sharif.edu).

This work was supported by Iran Telecommunication Research Center.

In the presented fuzzy classifier system, we use fuzzy if-then rules of the following form.

Rule R_j : If x_1 is A_{j1} and ... and x_n is A_{jn}, then Class C_j with $CF = CF_j$.

where R_j is the label of the *jth* fuzzy if–then rule, $A_{j1},...,A_{jn}$ are antecedent fuzzy sets on the unit interval $[0,1]$, C_j is the consequent class (i.e., one of the given c classes), and CF_j is the grade of certainty of the fuzzy if–then rule R_j. In computer simulations, we use a typical set of linguistic values in Fig. 1 as antecedent fuzzy sets. The membership function of each linguistic value in Fig. 1 is specified by homogeneously partitioning the domain of each attribute into symmetric triangular fuzzy sets. We use such a simple specification in computer simulations to show the high performance of our fuzzy classifier system, even if the membership function of each antecedent fuzzy set is not tailored. However, we can use any tailored membership functions in our fuzzy classifier system for a particular pattern classification problem.

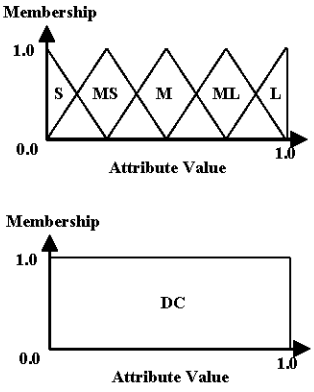

Fig. 1. The used antecedent fuzzy sets in this paper. 1: Small, 2: medium small, 3: medium, 4: medium large, 5: large, and 0: don't care.

The total number of fuzzy if–then rules is 6^n in the case of the n-dimensional pattern classification problem. It is impossible to use all the 6^n fuzzy if–then rules in a single fuzzy rule base when the number of attributes (i.e. n) is large (e.g., intrusion detection problem which n = 41).

Our fuzzy classifier system searches for a relatively small number of fuzzy if–then rules with high classification ability. Since the consequent class and the certainty grade of each fuzzy if–then rule can be determined from training patterns by a simple heuristic procedure [24], the task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for a set of fuzzy if–then rules. While this task seems to be simple at first glance, in fact it is very difficult for high-dimensional pattern classification problems, since the search space involves 6^n combinations.

In our fuzzy classifier system, the consequent Class C_j and the grade of certainty CF_j of each fuzzy if–then rule are determined by a modified version of the heuristic procedure which is discussed in [24].

To determine C_j and CF_j of each rule in the population the following steps should be done:

Step 1: Calculate the compatibility of each training pattern $x_p = (x_{p1}, x_{p2}, ..., x_{pn})$ with the fuzzy if–then rule R_j by the following product operation:

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) \times \ldots \times \mu_{jn}(x_{pn}), \qquad p = 1,2,\ldots,m, \qquad (1)$$

where $\mu_{ji}(x_{pi})$ is the membership function of i^{th} attribute of p^{th} pattern and M denotes total number of patterns.

Step 2: For each class, calculate the relative sum of the compatibility grades of the training patterns with the fuzzy if–then rule R_j :

$$\beta_{Class\ h}(R_j) = \sum_{x_p \in Class\ h} \mu_j(x_p) \Big/ N_{Class\ h}, \quad h = 1,2,\ldots,c \qquad (2)$$

where $\beta_{Class\ h}(R_j)$ is the sum of the compatibility grades of the training patterns in *Class h* with the fuzzy if–then rule R_j and $N_{Class\ h}$ is the number of training patterns which their corresponding class is *Class h* .

The described modification of the heuristic procedure has occurred in this step, since in the procedure discussed in [24] the sum of the compatibility grades is calculated instead of calculating the relative sum of the grades. This is because in intrusion detection problem some of the classes are very similar to each other. Moreover, the number of training patterns for each of the classes is significantly different. So if we use the traditional heuristic method of [24], the consequent class of R_j might be specified incorrectly.

Step 3: Find Class $\hat{h}_j$ that has the maximum value of $\beta_{Class\ h}(R_j)$:

$$\beta_{Class\ \hat{h}_j}(R_j) = \max\left\{\beta_{Class\ 1}(R_j),...,\beta_{Class\ c}(R_j)\right\}. \qquad (3)$$

If two or more classes take the maximum value, the consequent Class C_j of the fuzzy if–then rule R_j cannot be determined uniquely. In this case, let C_j be φ. If a single class takes the maximum value, let C_j be Class $\hat{h}_j$. If there is no training pattern compatible with the fuzzy if–then rule R_j (i.e., if $\beta_{Class\ h}(R_j) = 0$ for $h = 1,2,\ldots,c$) the consequent Class C_j is also specified as φ.

Step 4: If the consequent Class C_j is φ, let the grade of certainty CF_j of the fuzzy if–then rule R_j be $CF_j = 0$. Otherwise, the grade of certainty CF_j is determined as follows:

$$CF_j = \left(\beta_{Class\ \hat{h}_j}(R_j) - \overline{\beta}\right) \Big/ \sum_{h=1}^{c} \beta_{Class\ h}(R_j) \tag{4}$$

where

$$\overline{\beta} = \sum_{h \neq \hat{h}_j} \beta_{Class\ h}(R_j) / (c-1) \tag{5}$$

By the proposed heuristic procedure we can specify the consequent class and the certainty grade for any combination of antecedent fuzzy sets. Such a combination is generated by a fuzzy classifier system, which its construction steps will be explained in the next subsections.

The task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for generating a rule set S with high classification ability. When a rule set S is given, an input pattern $x_p = (x_{p1}, x_{p2}, ..., x_{pn})$ is classified by a single winner rule R_{j^*} in S, which is determined as follows:

$$\rho_{j^*}(x_p) \,.\, CF_{j^*} = \max\left\{\rho_j(x_p) \,.\, CF_j \,|\, R_j \in S\right\}. \tag{6}$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade CF_j.

The method of coding fuzzy if-then rules which is used in this paper is the same as the method which we employed in [11]. Each fuzzy if-then rule is coded as a string. The following symbols are used for denoting the five linguistic values: (Fig. 1)

0: don't care (DC), 1: small (S), 2: medium small (MS), 3: medium (M), 4: medium large (ML), 5: large (L).

Intrusion Detection is a high-dimensional classification problem with a 41-dimensional feature vector as its input and 5 classes as its output. The CID-IRL consists of c classifiers, where c is the number of classes. Each classifier contains a subset of rules with the same labels. The proposed algorithm focuses on learning of each class to improve the total accuracy of the main classifier. Therefore, the proposed evolutionary fuzzy rule learning algorithm is repeated for each class of the classification problem separately.

By considering the above feature of CID-IRL, the goal classifier consists of c classifiers. Each of these classifiers develops independently. The combination of the obtained fuzzy rule sets are used in the structure of the final classification system.

III. IDS BASED ON CID-IRL

CID-IRL is a kind of boosted evolutionary fuzzy system that learns fuzzy if-then rules in an incremental fashion, in that the evolutionary algorithm optimizes one fuzzy classifier rule at a time. The boosting mechanism reduces the weight of those training instances that are classifier correctly by the new rule. Therefore, the next rule generation cycle focuses on fuzzy rules that account for the currently uncovered or misclassified instances. At each iteration the fuzzy rule that can classifies the current distribution of training samples better than other rules of the population is selected out to be included in the final classification fuzzy rule base. The idea behind using the boosting mechanism is to aggregate multiple hypotheses generated by the same learning algorithm invoked over different distributions of the training data into a single composite classifier.

In the above learning framework we have used the fitness function which is computed according to equations (7) to (9).

$$f_P = \frac{\sum_{k|c^k = c_i} w^k \mu_{R_i}(x^k)}{\sum_{k|c^k = c_i} w^k} \tag{7}$$

$$f_N = \frac{\sum_{k|c^k \neq c_i} w^k \mu_{R_i}(x^k)}{\sum_{k|c^k \neq c_i} w^k} \tag{8}$$

$$fitness(R_j) = w_P\, f_P - w_N\, f_N \tag{9}$$

where,

f_P : rate of positive training instances covered by the rule R_i (correct classification).

f_N : rate of negative training instances covered by the rule R_i (misclassification).

w^k : a weight which reflects the frequency of the instance x^k in the training set.

w_P : the weight of positive classification

w_N : the weight of negative classification (misclassification).

Outline of the proposed iterative evolutionary fuzzy system is presented as follows:

Step 1: Generate an initial population of fuzzy if–then rules based on weight of training samples. (Initialization)

Step 2: Generate new fuzzy if–then rules by genetic operations. (Generation)

Step 3: Replace a part of the current population with the newly generated rules. (Replacement)

Step 4: Terminate the inner cycle of the algorithm if a stopping condition is satisfied, otherwise return to Step 2. (Inner Cycle Termination Test)

Step 5: Terminate the outer cycle of the algorithm if a stopping condition is satisfied, otherwise go to the next step (Outer Cycle Termination Test)

Step 6: Adjust the new weight of each training sample that covers by the new added fuzzy rule. Go to step 1. (Weight Adjustment)

Each step of CID-IRL is described as follows:

Step 1: Let us denote the number of fuzzy if-then rules in the population by N_{pop}. To produce an initial population, N_{pop} fuzzy if-then rules are generated according to a random pattern in the train dataset [24]. As it was mentioned in the previous section, the proposed evolutionary fuzzy system is considered for each of the classes of the classification problem separately. Therefore, the mentioned random pattern is extracted according to the patterns of the training dataset, which their consequent class is the same as the class that the algorithm works on. Note that the probability for each training pattern to be chosen in this step is proportional to its current weight. This means that the algorithm considers a greater probability for those patterns that have not been learned in previous iterations. Next, for this random pattern, we determine the most compatible combination of antecedent fuzzy sets using only the five linguistic values (Fig. 1). The compatibility of antecedent fuzzy sets with the random pattern is measured by (1). After generating each fuzzy if-then rule, the consequent class of this rule is determined according to the heuristic method, explained in the previous section. The generation of each fuzzy rule is accepted only if its consequent class is the same as its corresponding random pattern class. Otherwise, the generated fuzzy rule is rejected and the rule generation process is repeated. After generation of N_{pop} fuzzy if-then rules, the fitness value of each rule is evaluated by classifying all the given training patterns using the set of fuzzy if–then rules in the current population. Each fuzzy if–then rule is evaluated according to the fitness function, which is presented in equation (8):

Step 2: A pair of fuzzy if-then rules is selected from the current population to generate new fuzzy if-then rules for the next population. Each fuzzy if-then rule in the current population is selected using the tournament selection strategy. This procedure is iterated until a pre-specified number of pairs of fuzzy if-then rules are selected. A crossover operation is then applied to a selected random pair of fuzzy if–then rules with a pre-specified crossover probability. Note that the selected individuals for crossover operation should be different. In computer simulations of this paper, we have used the uniform crossover. After performing the crossover operation, consequent classes of the generated individuals are determined. If these classes are the same as their parent classes then the generated individuals are accepted, otherwise the crossover operation is repeated according to a pre-defined iteration number for each individual that its consequent class is not the same as its parents. We call the above-mentioned iteration number X_{repeat}. With a pre-specified mutation probability, each antecedent fuzzy set of fuzzy if–then rules is randomly replaced with a different antecedent fuzzy set after the crossover operation. After performing the mutation operation, consequent class of the mutated individual is determined. If the result class is the same as the class of the individual before the mutation operation the mutated individual is accepted, otherwise the mutation operation is repeated until a pre-specified iteration number. We call this number M_{repeat}. After performing selection, crossover and mutation steps, the fitness value of each of the generated individuals is evaluated according to equation (8).

Step 3: A pre-specified number of fuzzy if–then rules in the current population are replaced with the newly generated rules. In our fuzzy classifier system, P_R percent of the worst rules with the smallest fitness values are removed from the current population and $(100 - P_R)$ percent of the newly generated fuzzy if–then rules are added. (P_R is the replacement percentage) After performing the mentioned replacement procedure, the fitness value of each of the individuals is evaluated according to equation (8).

Step 4: We can use any stopping condition for terminating the inner cycle of the IRL-based fuzzy rule-learning algorithm. In computer simulations of this paper, we used the total number of generations as a stopping condition.

Step 5: After termination of the inner cycle of CID-IRL, the algorithm adds the best fuzzy rule of the evolved population to the final classification rules list and checks if this added fuzzy rule is capable of improving the classification rate of final classification system. If the classification rate is not improved the algorithm removes the added fuzzy rule from the final classification rules list and terminates. Otherwise, it goes to the next step.

Step 6: At each step, GA is run and rule R_t with the best fitness value is inserted into the fuzzy rule base. Since each inserted rule is an incomplete weak classifier, rules in the fuzzy rule base have a classification error value, denoted $E(R_t)$:

$$E(R_t) = 1 - CF_t \tag{10}$$

After each rule extraction process, instances that are misclassified will end up having the same weight, and those instances that classified correctly are reduced by some factor β^k. Hence, after the extraction of rule R_t, the weight at iteration $t+1$ becomes:

$$w^k(t+1) = \begin{cases} w^k(t) & if\ c_i \neq c_k \\ w^k(t) * \beta^k & if\ c_i = c_k \end{cases} \tag{11}$$

where β^k is calculated for each instance by using the following equation:

$$\beta^k = \left(\frac{E(R_t)}{1 - E(R_t)} \right)^{\mu_{R_t}(x^k)} \tag{12}$$

Note that initially $w_k = 1$. After this step, the algorithm jumps to step 1.

I. Experimental Results

We applied our proposed method to the Knowledge Discovery and Data (KDD) Mining Cup 1999 intrusion-detection data set. Each object in the data set is a network connection. Each object is defined in 41D space, and belongs to one of five classes: normal, probe, denial-of-service (DOS), unauthorized access to root (U2R), and unauthorized access from remote machine (R2L). Objects in the normal class are harmless connections, whereas objects in the other four classes are different types of attacks. The training set contains 494,021 connections; the text data includes 311,029. The KDD Cup 1999 data set is the only large-scale, publicly available data for evaluating intrusion-detection tools. A detailed description of the data set is available at [22]. We have used a subset of the 10% KDD-Cup 99 dataset as our train dataset. The test dataset is the same as that, which was used in evaluating classification algorithms in KDD-Cup 99 contest. We normalized the train and test data sets, where each numerical value in the data set is normalized between 0.0 and 1.0. Table I shows parameter specification that we have used in our computer simulations for CID-IRL. The evolutionary process of CID-IRL is investigated in Fig. 2. According to this figure, we can comprehend that our proposed iterative fuzzy rule learning algorithm is capable of evolving fuzzy if-then rules that cooperate and compete with one another efficiently.

Classification performance of CID-IRL is measured and compared with that of different baseline classifiers including pruning C4.5, Naïve Bayes (NB), *k*-Nearest Neighbor (*k*-NN) and Support Vector Machine (SVM). In *k*-NN parameter *k* is set to 5, and the SVM is trained using the well-known fast sequential minimal optimization method with a polynomial kernel. Table II shows the results of Recall, Precision, and F-measure of different classifiers for each class of intrusion detection problem. This table shows that our proposed evolutionary fuzzy system is within the best three top classifiers for all of the classes in the investigated classification problem. Therefore, we can conclude that our proposed evolutionary fuzzy system is a reliable approach for generating a high performance classification system.

II. Conclusions

In this paper, the use of an iterative evolutionary fuzzy system (CID-IRL) is investigated to develop an intrusion detection system capable of detecting intrusive behaviors in a computer network. Computer simulations on DARPA datasets demonstrate high performance of CID-IRL for intrusion detection. As intrusion detection is a high-dimensional classification problem one of the important properties of the proposed EFSs in this paper is that the class labels of all of the rules in the population are the same. This feature allows the algorithm to focus on learning of each class independently. An initialization procedure is used to generate fuzzy if-then rules directly from the training data set. These rules enable the algorithm to focus on finding fuzzy rules, which are related to a special class. Moreover, the probability of choosing an instance from the training data was depended on the instance weight. This technique enabled the learning algorithm to guide its evolutionary process at its start up significantly. The performance of CID-IRL was compared to several classification algorithms. Results showed that the performance of the presented iterative algorithm is competitive to several well-known classification algorithms such as pruning C4.5, Naïve Bayes (NB), *k*-Nearest Neighbor (*k*-NN) and Support Vector Machine (SVM).

It would be interesting to investigate the performance of other kinds if evolutionary fuzzy systems (e.g. Michigan and Pittsburgh approaches) for the intrusion detection classification problem. Moreover, the use of multi-objective evolutionary fuzzy systems to extract a comprehensible fuzzy classifier for intrusion detection is another considerable investigation topic, which is left for our future work.

TABLE I
PARAMETERS SPECIFICATION IN COMPUTER SIMULATIONS FOR CID-IRL

Parameter	Value
population size (N_{pop})	200
crossover probability (P_c)	90
mutation probability (P_m)	10
Crossover attempts (X_{repeat})	20
Mutation attempts (M_{repeat})	20
Weight of positive class (w_P)	0.01
Weight of negative class (w_N)	0.99
replacement percentage (P_{repR})	20
maximum number of generations	200

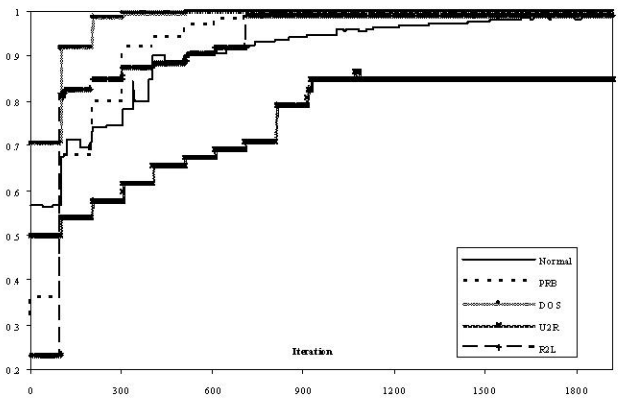

Fig. 2. Classification rate progress for different classes of intrusion detection problem during several iterations of CID-IRL

TABLE II

RECALL, PRECISION, AND F-MEASURE FOR DIFFERENT CLASSIFIERS. THE BESTS ARE BOLD-UNDERLINED, THE SECONDS ARE BOLD, AND THE THIRDS ARE UNDERLINED.

Class	Algorithm	C4.5	NB	5-NN	SVM	CID-IRL
NORMAL	Recall	**98.3**	55.4	95.8	**97.9**	**98.3**
	Precision	**74.7**	43.3	74.1	73.4	**74.5**
	F-measure	**84.9**	48.6	83.6	83.9	**84.8**
PRB	Recall	81.8	**90.4**	81.6	**86.2**	82.5
	Precision	52.2	64.1	55.4	**77.7**	**72.1**
	F-measure	63.7	75	66	**81.7**	**77.1**
DOS	Recall	96.9	82.7	97	**97.5**	**97**
	Precision	99.6	94	99.4	**99.8**	**99.8**
	F-measure	98.3	88	98.1	**98.7**	**98.4**
U2R	Recall	14.4	13.1	**14.9**	10	**24.5**
	Precision	**9.3**	2	5.4	**53.4**	6.7
	F-measure	**11.3**	3.5	8	**16.9**	10.5
R2L	Recall	1.4	**62.7**	**6.9**	3.5	4.3
	Precision	30.3	42.7	**66.9**	62.3	**74.4**
	F-measure	2.7	**50.8**	**12.5**	6.7	8.2

REFERENCES

[1] Ajith Abraham, Ravi Jain, Johnson Thomas, Sang Yong Han, "D-SCIDS: Distributed soft computing intrusion detection system", Journal of Network and Computer Applications 30, pp. 81–98, 2007.

[2] Murali, A., Rao, M., "A Survey on Intrusion Detection Approaches," First International Conference on Information and Communication Technologies, Page(s):233 – 240, 27-28 Aug. 2005.

[3] Nong Ye, Qiang Chen, Borror, C.M., "EWMA forecast of normal system activity for computer intrusion Detection," IEEE Transactions on Reliability, Volume 53, Issue 4, Page(s):557 – 566, Dec. 2004.

[4] Axelsson S. Intrusion detection systems: a survey and taxonomy. Technical report no. 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden. March 2000.

[5] Idris, N.B., Shanmugam, B., "Artificial Intelligence Techniques Applied to Intrusion Detection," Annual IEEE INDICON, 2005 11-13, Page(s):52 – 55, Dec. 2005.

[6] Sung-Bae Cho, "Incorporating soft computing techniques into a probabilistic intrusion detection system," IEEE Transactions on Systems, Man and Cybernetics, Part C, Volume 32, Issue 2, ,Page(s):154 – 160, May 2002.

[7] Jun-feng Tian, Yue Fu, Ying Xu, Jian-ling Wang, "Intrusion Detection Combining Multiple Decision Trees by Fuzzy Logic," Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, Page(s):256 – 258, 05-08 Dec. 2005.

[8] Cho S. Cha S, "SAD: web session anomaly detection based on parameter estimation", Computers & Security, Vol.23, No.4, pp.265-351, June 2004.

[9] Hai-Hua Gao, Hui-Hua Yang, Xing-Yu Wang, "Ant Colony Optimization Based Network Intrusion Feature Selection and Detection", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.

[10] T. Ozyer, R. Alhajj, K. Barker, "Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening," Journal of Network and Computer Applications 30, pp. 99–113, 2007.

[11] M. Saniee Abadeh, J. Habibi, and C. Lucas, "Intrusion Detection Using a Fuzzy Genetics-Based Learning Algorithm," Journal of Network and Computer Applications, 414-428, 2007.

[12] S. Axelsson, "The base-rate fallacyand the difficultyof intrusion detection," ACM Trans. Informat. Syst. Security 3 (3), pp. 186–205, 2000.

[13] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), pp. 251-261, Oct. 2003.

[14] Yong Feng, Zhong-Fu Wu, Kai-Gui Wu, Zhong-Yang Xiong, Ying Zhou, "An Unsupervised Anomaly Intrusion Detection Algorithm Based On Swarm Intelligence", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.

[15] Ahmed Awad E. Ahmed, and Issa Traore, "Anomaly Intrusion Detection based on Biometrics", Proceedings of the 2005 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 2005.

[16] O. Cordon, F. Gomide, F. Herrera, F. Hofmann, L. Magdalena, "Ten years of genetic fuzzy systems current framework and new trends", Fuzzy Sets and Systems 141, pp. 5–31, 2004.

[17] Yi-Chung Hu a, Ruey-Shun Chen a, Gwo-Hshiung Tzeng, "Finding fuzzy classification rules using data mining techniques," Pattern Recognition Letters 24, pp. 509–519, 2003.

[18] Hisao Ishibuchi, Takashi Yamamoto, and Tomoharu Nakashima, "Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 35, NO. 2, APRIL 2005.

[19] S. E. Rouwhorst and A. P. Engelbrecht, "Searching the forest: Using decision trees as building blocks for evolutionary search in classification databases," in Proc. IEEE Congr. Evolutionary Computation, vol. 1, pp. 633-638, 2000.

[20] H. Ishibuchi, T. Nakashima, and T. Murata, "Three-objective genetics-based machine learning for linguistic rule extraction", Information Sciences, pp. 109-133, 2001.

[21] F. Hofmann, "Combining boosting and evolutionary algorithms for learning of fuzzy classification rules," Fuzzy Sets and Systems, pp. 47–58, 2004.

[22] KDD-cup data set: http://kdd.ics.uci.edu/databases/kddcup99/task.html.

[23] C. Elkan, "Results of the KDD_99 classifier learning," ACM SIGKDD Explorations 1, pp. 63–64, 2000.

[24] H. Ishibuchi, and T. Nakashima, "Improving the Performance of Fuzzy Classifier Systems for Pattern Classification Problems with Continuous Attributes", IEEE Transactions on Industrial Electronics, vol. 46, no. 6, Dec., 1999.

[25] A. Gonzalez and R. Perez, "SLAVE: A genetic learning system based on an iterative approach," IEEE Transaction on Fuzzy System, vol 7(2) pp. 176-191, 1999.