INDUCTION OF FUZZY CLASSIFICATION SYSTEMS VIA EVOLUTIONARY ACO-BASED ALGORITHMS

MOHAMMAD SANIEE ABADEH, JAFAR HABIBI, and EMAD SOROUSH

Computer Engineering Department, Sharif University of Technology, Tehran, Iran
saniee@ce.sharif.edu, jhabibi@sharif.edu, soroush@csc.uvic.ca

Abstract: In this paper we have proposed an evolutionary algorithm to induct fuzzy classification rules. The algorithm uses an ant colony optimization based local searcher to improve the quality of final fuzzy classification system. The proposed algorithm is performed on Intrusion Detection as a high-dimensional classification problem. Results show that the implemented evolutionary ACO-Based algorithm is capable of producing a reliable fuzzy rule-based classifier for intrusion detection.

Keywords: Evolutionary Computation, Fuzzy Rule Learning, Ant Colony Optimization, Intrusion Detection

1. INTRODUCTION

Intrusions into computer systems have caused many quality and reliability problems with those systems. Furthermore, the number of intrusions into computer systems grows rapidly because new automated hacking tools appear each day, and these tools along with various system vulnerability information are easily available on the web. The problem of intrusion detection is studied extensively in computer security [1] - [4], and has received a lot of attention in machine learning and data mining [5, 6]. Intrusion Detection Systems (IDSs) generally takes one of two approaches: anomaly detection, and signature recognition. Signature recognition techniques store patterns of intrusion signatures, and compare those patterns with the observed activities for a match to detect an intrusion. Signature recognition techniques are used in most existing intrusion detection systems, commercial or freeware. Anomaly detection techniques identify an intrusion when the observed activities in computer systems demonstrate a large deviation from the norm profile built on long-term normal activities. The goal of this paper is to evolve a classification system capable of detecting known intrusions into a computer network. Hence, our goal is to develop a signature recognition system. The technique, which we have used to detect intrusion in a computer network, is based on fuzzy genetic learning. Genetic algorithms (GA's) are search algorithms that use operations found in natural genetic to guide the journey through a search space. GA's have been theoretically and empirically proven to provide robust search capabilities in complex spaces offering a valid approach to problems requiring efficient and effective searching. One of the interesting applications of GA's is in pattern classification problems in which our goal is to develop a classifier capable of dealing with different classes of a specific problem. Genetic algorithms [7] have been used as rule generation and optimization tools in the design of fuzzy rule based systems [8-13]. Those GA-based studies on the design of fuzzy rule-based systems are usually referred to as fuzzy genetics-based machine learning methods (fuzzy GBML methods), each of which can be classified into the Michigan, Pittsburgh or Iterative Rule Learning (IRL) approaches [14]. Some studies are categorized as the Michigan approach [12, 13] where a single fuzzy if–then rule is coded as an individual. In Pittsburgh approach, a set of fuzzy if–then rules is coded as an individual [15] - [17]. 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 [24]. In this paper, we have extended our previous Michigan-based intrusion detection algorithm from a problem with two classes [18] to a five class classification problem. To accomplish this purpose we have used a hybrid genetic algorithm, which is boosted with an ant colony optimization heuristic. This heuristic is capable of increasing the searching power of the main evolutionary algorithm. The ACO heuristic is combined to the evolutionary algorithm in the form of a local search step. In other words, we have improved the global search capability of the evolutionary algorithm by adding a local search step to its main structure. The proposed approach has been tested using the public KDD CUP'99 intrusion detection data set available at the University of California, Irvine web site [19].

The rest of the paper is as follows: Section 2 is dedicated to related works in the field of intrusion detection. Induction of fuzzy-rule-based classifiers using a Michigan based evolutionary algorithm is introduced in Section 3. Section 4 describes the presented heuristic local search, which is based on Ant Colony Optimization approach. Experimental results are reported in Section 5. Section 6 is conclusions.

2. RELATED WORKS

There are several approaches for solving intrusion detection problems. Lee built an intrusion detection model using association rule and frequent episode techniques on system audit data [25].

Neural networks have been extensively used to detect both misuse and anomalous patterns [26]-[30]. An n-layer network is constructed and abstract commands are defined in terms of sequence of information units, the input to the neural in the training data.

Each command is considered with pre-defined w commands together to predict the next coming command expected from the user. After training, the system has the profile of the user. At the testing step, the anomaly is said to occur as the user deviates from the expected behavior [31, 32]. Short sequences of system calls carry out the prediction process. In this system, Hamming distance comparison with a threshold is used to discriminate the normal sequence from the abnormal sequence [33, 34].

Some recent researches have utilized Artificial Immune Systems to detect intrusive behaviors in a computer network [35]-[37].

Some other applied techniques on intrusion detection problem are genetic algorithms [18, 24], Bayesian parameter estimation [38] and clustering [39]-[43].

The main contribution of this paper is hybridizing evolutionary algorithms with Ant Colony Optimization meta-heuristic for intrusion detection. In the next section, we will discuss about the details of the proposed hybrid algorithm.

3. PROPOSED ALGORITHM

First, let us explain about the method of coding fuzzy rules, which is used in this paper. Each fuzzy if-then rule is coded as a string. The following symbols are used for denoting the five linguistic values (Fig. 1): small (A_1), medium small (A_2), medium (A_3), medium large (A_4) and large (A_5). For example, the following fuzzy if-then rule is coded as $(A_3, A_2, A_5, A_1), C_j, CF_j$: If x_1 is medium and x_2 is medium small and x_3 is large and x_4 is small then Class C_j with $CF = CF_j$. Outline of the proposed Michigan approach based IDS algorithm is as follows:

Step 1: Generate an initial population of fuzzy if–then rules. (Initialization)

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

Step 3: Each individual can live according to its age number. The procedure that accomplishes this lifetime is based on a ACO heuristic algorithm. (ACO-Based Local Search)

Step 4: Replace a part of the current population with the newly generated rules. (Replacement)

Step 5: Terminate the algorithm if a stopping condition is satisfied, otherwise return to Step 2. (Internal termination test)

Step 6: Save the best individuals of the resulted population. Terminate the whole algorithm if a maximum number of iterations is satisfied. Otherwise, go to step 1.

This section will discuss about each of the above steps in detail.

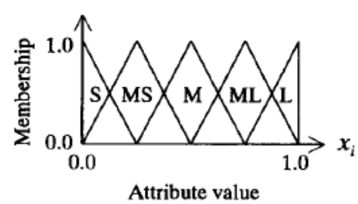

Figure 1: Membership functions of five linguistic values (S: small, MS: medium small, M: medium, ML: medium large, L: large).

3.1. Introduction: 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 [12]. The proposed evolutionary fuzzy system (EFS) is considered for each of the classes of the classification problem separately. One of the important benefits of this separation is that the learning system can focus on each of the classes of the classification problem. According to this fact, 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. Next, for this random pattern, we determine the most compatible combination of antecedent fuzzy sets using only the five linguistic values (Figure. 1). The compatibility of antecedent fuzzy rules with the random pattern is calculated by equation (1).

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

where $\mu_{A_{ji}}(.)$ is the membership function of A_{ji} and m is the total number of training patterns.

After generating each fuzzy if-then rule, the consequent class of this rule is determined according to (2).

$$\beta_{Class\,\hat{h}_j}(R_j)=\max\{\beta_{Class\,1}(R_j),\ldots,\beta_{Class\,c}(R_j)\} \tag{2}$$

where,

$$\beta_{Class\,h}(R_j)=\sum_{x_p\in Class\,h}\mu_j(x_p)\Big/N_{Class\,h}, \quad h=1,2,\ldots,c \tag{3}$$

where $\beta_{Class\,h}(R_j)$ is the average 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 $Class\,h$. Each of the fuzzy rules in the final classification has a certainty grade, which denotes the strength of that fuzzy rule. This number is calculated according to (4).

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

where,

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

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:

$$\mu_{\hat{j}}(x_p)\,.\,CF_{\hat{j}}=\max\left\{\mu_j(x_p)\,.\,CF_j\;|R_j\in S\right\} \tag{6}$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade CF_j. The classification is rejected if no fuzzy if–then rule is compatible with the input pattern x_p (i.e., $\mu_j(x_p)=0$ for $\forall R_j\in S$). 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. The fitness value of the fuzzy if–then rule is evaluated by the following fitness function:

$$fitness(R_j)=NCP(R_j) \tag{7}$$

where, $NCP(R_j)$ denotes the number of correctly classified training patterns by rule R_j. [12]

3.2. Generation: 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 by the tournament selection procedure (the size of the tournament is 3 in the experiments of this paper). A crossover operation is 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 used the one-point 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 until a pre-specified iteration number for each individual that its consequent class is not the same as its parents. We call this 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}. The above procedure is iterated until a pre-specified number of pairs of fuzzy if-then rules are generated.

3.3. Local Search: In order to improve the classification rate of the population, we consider a lifetime for each individual of the current population. We simulate this lifetime using a local search algorithm, which is based on an Ant Colony Optimization heuristic. Before describing the details of the ACO-based local search procedure, which will be done in section 4, it is essential to introduce the concept of age for each individual. An individual can search its neighborhood (or can live) according to its age. The age of each individual depends on its fitness. In other words, fitter individuals have greater age or opportunity to live. This rule is based on the concept of "survival of the fittest" [7]. To accomplish this idea we determine the age of each individual using equation (8).

$$Age(R_j) = w_{age} \frac{fitness(R_j)}{N_{Class\,h}} \tag{8}$$

where, w_{age} denotes a weight which depends on the CPU speed of the computer that the algorithm runs on (The faster the CPU the lower the w_{age}). Note that the division in the equation is for normalizing the fitness of R_j. According to equation (8), each individual can live and improves its quality (fitness) as much as its age number determines. The unit of age number is in milliseconds.

3.4. Replacement: 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_{repR} percent of the worst rules with the smallest fitness values are removed from the current population and (100 - P_{repR}) percent of the newly generated fuzzy if–then rules are added.

3.5. Termination Tests: We can use any stopping condition for terminating the internal cycle of the Michigan based fuzzy classification algorithm. In computer simulations of this paper, we used the total number of generations as a stopping condition. Fuzzy rules of the final population that their fitness is not zero are stored in the Fuzzy Rule Pool (FRP). This pool is updated until a pre-specified number of iterations for the external cycle of the evolutionary ACO-Based algorithm.

4. ACO-BASED LOCAL SEARCH

ACO was developed by Dorigo et al. [21] based on the fact that ants are able to find the shortest route between their nest and a source of food. This is done using pheromone trails, which ants deposit whenever they travel, as a form of indirect communication. The main originality of this paper is the introduction of a heuristic local search algorithm, which is inspired from the life of real ants. In other words, we have used ACO as a cooperative learning algorithm in the structure of the main Michigan based evolutionary fuzzy rule learning system. As we have mentioned in the previous section, each individual in the population of the main evolutionary algorithm has an age, which determines its time of living. Since we have mimicked the rule of life in a colony of ants, it is possible to say that the age number indicates the whole colony surviving time for improving the quality of current fuzzy rule. The pseudo-code of the ACO-Based Local Search algorithm is presented in figure 2. Note that the input of this algorithm is a fuzzy rule and the output is an improved version of that fuzzy rule. The improvement is accomplished by some modifications (local search) to the current (input) fuzzy rule. The algorithm is capable of searching for the best modification according to the lifetime (age) of the current fuzzy rule. According to figure 2, in each step the algorithm performs K changes to the current (input) fuzzy rule. For each K value, a complete ACO process is done with a specific number of ants. The desirability of each ants for changing the i th antecedent of current rule R_C to A_j is given by (9).

$$p_a(R_C, i, A_j) = \frac{[\tau(R_C, i, A_j)][\eta(R_C, i, A_j)]^{\beta}}{\sum_{u=1}^{5}[\tau(R_C, i, A_u)][\eta(R_C, i, A_u)]^{\beta}} \tag{9}$$

In this equation τ is the pheromone and η is a heuristic probability, which is determined according to equations (10) to (12).

$$\eta(R_C, i, A_j) = \frac{N(i, A_j)}{\sum_{v=1}^{5} N(i, A_v)} \tag{10}$$

$$N(i, A_j) = \sum_{p=1}^{m} n_p(i, A_j) \tag{11}$$

$$n_p(i, A_j) = \begin{cases} 1 & \arg\max_{v=1}^{5} \{\mu_{A_v}(x_{pi})\} = \mu_{A_j}(x_{pi}) \\ 0 & otherwise \end{cases} \tag{12}$$

```
Initialize the pheromone trails and parameters;
For K=1 to maximum rule length  // Performing K changes to the fuzzy rule
  While maximum number of iterations or best-rule-change converged
    For a=1 to number of ants
      For i=1to K
        Randomly change an antecedent part of the fuzzy rule to a linguistic value with maximum
          desirability;
        Performing local pheromone updating;
      End-For i;
      Saving rule-change as the single ant searching result;
    End-For a;
    Performing global pheromone updating;
    Saving best-rule-change as the whole ants searching result;
  End-While;
End-For K;
```

Figure 2: Pseudo-code for the ACO-Based Local Search algorithm

According to this heuristic, the most probable linguistic value for a specific antecedent part of the fuzzy rule is computed. Note that this computation is based on the train data of the current class, which the main evolutionary algorithm works on (the evolutionary algorithm performs for each of the classes of the classification problem separately). After performing any single change, an ant accomplishes a local pheromone updating according to equation (13).

$$\tau(R_C,i,A_j) \leftarrow \tau(R_C,i,A_j) - \rho.(\tau(R_C,i,A_j) - \Delta\tau(R_C,i,A_j)) \qquad \textbf{(13)}$$

where $0 < \rho < 1$ is the local pheromone decay parameter, and $\Delta\tau(R_C, i, A_j) = \tau_0$ where τ_0 is the initial pheromone level. When all of the ants perform their search, the global pheromone updating performs according to equation (14).

$$\tau(R_C, i, A_j) \leftarrow (1-\alpha).\tau(R_C, i, A_j) + \alpha.\Delta\tau(R_C, i, A_j) \qquad \textbf{(14)}$$

where,

$$\Delta\tau(R_C, i, A_j) = \frac{fitness(R_N)}{N_{Class\,h}} \qquad \textbf{(15)}$$

$0 < \alpha < 1$ is the global pheromone decay parameter, and R_N stands for the new rule which is obtained after modifying ith antecedent part of current rule R_C to A_j. The result of local search algorithm will be the best change to the current (input) fuzzy rule. Here the best means a fuzzy rule with maximum fitness according to equation (7).

5. EXPERIMENTAL RESULTS

Experiments were carried out on a subset of the database created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program [21]. We used the subset that was pre-processed by the Columbia University and distributed as part of the UCI KDD Archive [19].

Table 1: Parameters specification in computer simulations

Parameter	Value
population size (N_{pop})	20
crossover probability (P_c)	0.9
mutation probability (P_m)	0.1
age weight parameter (w_{age})	10000
number of ants (N_{ant})	50
global pheromone decay (α)	0.02
local pheromone decay (ρ)	0.02
replacement percentage (P_{rep})	20
stopping condition of the internal algorithm cycle	100
stopping condition of the external algorithm cycle	4

Table 2: Comparison of some algorithms

Algorithm	CLASS NORMAL	U2R	R2L	DOS	PRB
Hybrid EFS	98.5	76.3	89	98.5	82.5
Simple EFS	99.2	44	79.4	92.9	68.2
C4.5 [44]	95.9	21.1	30.2	97.1	76.3
NB [45]	94.2	25	5.4	79.4	90.4
5-NN [46]	96.3	25.4	3.8	96.7	87.5
SVM [47, 48]	96.2	17.1	3.8	82.7	87.3
EFRID [6]	92.78	88.13	7.41	98.91	50.35
Winner Entry [22]	94.5	13.2	8.4	97.1	83.3

This section will compare the performance of the proposed hybrid evolutionary fuzzy system with other classification methods for the intrusion detection case study. Table 1 presents parameter settings for the ACO-Based Evolutionary Fuzzy System, which is used in this paper. The classification accuracies of hybrid and simple evolutionary fuzzy systems are shown in Table 2 along with the classification rates achieved by the some other algorithms.

Table 3: Performance comparison

Algorithm	Detection Rate	False Alarm Rate
Hybrid EFS	95.5	0.001831
Simple EFS	95.5	0.00341
C4.5	94.7	6.5
NB	91.1	3.4
5-NN	91.4	4.5
SVM	91.3	6.7
EFRID	98.15	7.0
RIPPER-Artificial Anomalies	94.26	2.02

Although the evolutionary fuzzy systems use smaller percent of the original intrusion dataset, the classification accuracies of them are competitive to other classification algorithms. Moreover the hybrid evolutionary system that uses the proposed ACO-Based Local Search procedure outperforms the simple evolutionary fuzzy system (here simple means without local search step). The Detection and False Alarm rates for the evolved fuzzy classifier systems are compared with that of several classification methods (RIPPER algorithm [23] is added) in Table 3. According to Table 3, it is clear that using the evolved fuzzy classifiers, resulted in a more reliable Intrusion Detection Systems since the false alarm rate decreases significantly. This feature is so critical for a security system, because false alarms might deteriorate the reliability and performance of the Intrusion Detection System. However, the price of achieving such a low false alarm rate is the decrement of Detection Rate in our proposed genetic fuzzy systems.

6. CONCLUSION

In this paper, a novel ACO-Based Local Search procedure is introduced. The proposed local search procedure is used in the structure of a Michigan based evolutionary fuzzy system. The capability of the resulted hybrid fuzzy system is investigated according to the intrusion detection classification problem. Computer simulations on DARPA datasets demonstrate high performance of ACO-Based evolutionary fuzzy system for intrusion detection. Our future work is to consider the comprehensibility of the final detection system as another objective for our classification problem. To accomplish this objective we should minimize number of antecedents of each fuzzy rule while maximizing the classification rate of it.

REFERENCES

[1] R. Heady, G. Luger, A. Maccabe, and M. Sevilla (1990). The Architecture of a Network-level Intrusion Detection System. Technical report, CS90-20. Dept. of Computer. Science, University of New Mexico, Albuquerque, NM 87131, August 1990.

[2] E. Amoroso (1999). Intrusion detection. In Intrusion.net Books, January 1999.

[3] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. (1999). 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.

[4] S. Axelsson (2000). Intrusion detection systems: A survey and taxonomy. Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[5] J. Sundar, J. Garcia-Fernandez, D. Isaco, E. Spafford, and D. Zamboni, “An architecture for intrusion detection using autonomous agents,” Technical report 98/05, Purdue University, 1998.

[6] J. Gomez, and D. Dasgupta, "Evolving Fuzzy Classifies for Intrusion Detection," Proceedings of the 2002 IEEE Information Assurance Workshop, Jun. 2001.

[7] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

[8] C. L. Karr, and E. J. Gentry, “Fuzzy control of pH using genetic algorithms,” IEEE Trans. on Fuzzy Systems, vol. 1, pp. 46-53, February, 1993.

[9] Herrera, M. Lozano, and J. L. Verdegay, “Tuning fuzzy logic controllers by genetic algorithms,” International J. of Approximate Reasoning, vol. 12, no. 3/4, pp. 299-315, April/May, 1995.

[10] B. Carse, T.C. Fogarty, and A. Muntro, “Evolving fuzzy rule based controllers using genetic algorithms,” Fuzzy Sets and Systems, vol. 80, no. 3, 3, pp. 273-293, June, 1996.

[11] M. Valenzuela-Rendon, “The fuzzy classifier system: A classifier system for continuously varying variables,” Proc. Of 4th International Conference on Genetic Algorithms, pp. 346-353, University of California, San Diego, CA, July, 1991.

[12] 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, December 1999.

[13] H. Ishibuchi, T. Nakashima and T. Muratam “Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems,” IEEE Trans. On Systems, Man, and Cybernetics, October, 1999.

[14] 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.

[15] S. F. Smith, “A learning system based on genetic algorithms,” Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA, 1980.

[16] Herrera, M. Lozano, and J. L. Verdegay, “Tuning fuzzy logic controllers by genetic algorithms,” International J. of Approximate Reasoning, vol. 12, no. 3/4, pp. 299-315, April/May, 1995.

[17] B. Carse, T.C. Fogarty, and A. Muntro, “Evolving fuzzy rule based controllers using genetic algorithms,” Fuzzy Sets and Systems, vol. 80, no. 3, 3, pp. 273-293, June, 1996.

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

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

[20] Dorigo M,Maniezzo V,ColorniA.Ant system: optimization by a colony of cooperating agents. IEEE Trans SystMan Cybern 1996;26(1):29–41.

[21] http://www.ll.mit.edu.

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

[23] W. Fan, W. Lee, M. Miller, S. J. Stolfo, and P. K. Chan, “Using artificial anomalies to detect unknown and know network intrusions,” Proceedings of the First IEEE International Conference on Data Mining, 2001.

[24] 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.

[25] W. Lee, JS. Salvatore, and KWM. Mok, “Mining audit data to build intrusion detection models,” In: Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining, pp. 66–72, 1998.

[26] J. Cannady, “Artificial neural networks for misuse detection,” In: National information systems security conference, pp. 368–381, 1998.

[27] H. Debar, and B. Dorizzi, “An application of a recurrent network to an intrusion detection system,” In: Proceedings of the international joint conference on neural networks, pp. 78–83, 1992.

[28] H. Debar, B. Becke, and D. Siboni, “A neural network component for an intrusion detection system,” In: Proceedings of the IEEE Computer Society symposium on research in security and privacy, pp. 240–250, 1992.

[29] S. Mukkamala, AH. Sung, “Feature selection for intrusion detection using neural networks and support vector machines,” Journal of the Transport Research Board National Academy, Transport Research Record No. 1822, pp. 33–39, 2003.

[30] M. Riedmiller, H. Braun, “A direct adaptive method for faster back propagation learning: the RPROP algorithm,” Proceedings of the IEEE international conference on neural networks, San Francisco, 1993.

[31] KL. Fox, RR. Henning, JH. Reed, and R. Simonian, “A neural network approach towards intrusion detection,” In: Proceedings of the national security conference, Washington DC, pp. 125–134, 1990.

[32] J. Ryan, M. Lin, R. Miikkulainen, “Intrusion detection with neural networks,” In: Advances in neural information processing systems, vol. 10. Cambridge, MA: MITPress; 1998.

[33] S. Bridges, MS, RB. Vaughn, “Fuzzy data mining and genetic algorithms applied to intrusion detection,” In: Proceedings of the national information systems security conference, 2000.

[34] SA. Hofmeyr, S. Forrest S, A. Somayaji, “Intrusion detection using sequences of system calls,” Journal of Computer Security, vol. 6, pp.151–180, 1998.

[35] D. Dasgupta, and F. González, “An Immunity-Based Technique to Characterize Intrusions in Computer Networks,“ IEEE Transactions on Evolutionary Computation, vol. 6, no. 3, Jun. 2002.

[36] P. K. Harmer, P. D. Williams, G. H. Gunsch, and G. B. Lamont, “An Artificial Immune System Architecture for Computer Security Applications,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 3, Jun. 2002.

[37] Yang XR, Shen JY, Wang R. “Artificial immune theory based network intrusion detection system and the algorithms design”, Proceedings of 2002 International Conference on Machine Learning and Cybernetics, Beijing, pp.73-77, November 2002.

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

[39] Baoguo Xu, Apin Zhang, ”Application of Support Vector Clustering Algorithm to Network Intrusion Detection”, Neural Networks and Brain, 2005. ICNN&B '05. International Conference on Volume 2, 13-15 Oct. Page(s):1036 – 1040, 2005.

[40] Oh SH and Lee WS. “An anomaly intrusion detection method by clustering normal user behavior”, Computers & Security, Vol 22, No.7, pp.596-612, November, 2003.

[41] Baoguo Xu, Apin Zhang, ”Application of Support Vector Clustering Algorithm to Network Intrusion Detection”, Neural Networks and Brain, 2005. ICNN&B '05. International Conference on Volume 2, 13-15 Oct. Page(s):1036 - 1040, 2005.

[42] E. Leon, 0. Nasraoui, and J. Gomez, "Anomaly detection based on unsupervised niche clustering with application to network intrusion detection," in Proceedings of IEEE Conference on Evolutionary Computation (CEC), pp. 502-508, 2004.

[43] Y. Guan, A. A. Ghorbani, and N. Belacel, "Y-MEANS: a clustering method for intrusion detection," in Canadian Conference on Electrical and Computer Engineering, pp. 1083-1086, 2003.

[44] R. Quinlan, “C4.5: Programs for Machine Learning”, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[45] G.H. John, and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers”, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo, 1995.

[46] D. Aha, and D. Kibler, “Instance-based learning algorithms”, Machine Learning, vol. 6, pp. 37-66, 1991.

[47] J. Platt, “Fast Training of Support Vector Machines using Sequential Minimal Optimization”, Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press, 1998.

[48] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy, “Improvements to Platt's SMO Algorithm for SVM Classifier Design”, Neural Computation, vol. 13, no. 3, pp 637-649, 2001.

AUTHOR BIOGRAPHIES

MOHAMMAD SANIEE ABADEH received his B.S. degree in Computer Engineering from Isfahan University of Technology, Isfahan, Iran, in 2001, the M.S. degree in Artificial Intelligence from Iran University of Science and Technology, Tehran, Iran, in 2003 and his Ph.D. degree in Artificial Intelligence at the Department of Computer Engineering in Sharif University of Technology, Tehran, Iran in February 2008. His research has focused on developing advanced meta-heuristic algorithms for data mining and knowledge discovery purposes. His interests include computational intelligence, evolutionary algorithms, fuzzy genetic systems, memetic algorithms, biological computing, and data mining.

JAFAR HABIBI received his B.S. degree in computer engineering from the Supreme School of Computer, his M. S. degree in industrial engineering from Tarbiat Modares University and his Ph.D. degree in Computer engineering from Manchester University. At present, he is a faculty member at the computer engineering department at Sharif University of Technology and the managing directorputing Machine Services. He is supervisor of Sharif’s Robo-Cup Simulation Group. His research interests are mainly in the areas of computer engineering, simulation systems, MIS, DSS and evaluation of computer systems performance.

EMAD SOROUSH received his B.S. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran. He is now pursuing his graduate degree at CSC Dep of University of Victoria in Canada. His interests include data mining, machine learning, evolutionary optimization algorithms and planning.