A Novel Optimum Approach for Misuse Detection

AbdolHamid MomenZadeh [a,*]

[a,*] Corresponding Author: Department of Computer Engineering, Masjed-Soleiman Branch, Islamic Azad University (I.A.U), Masjed-Soleiman, Iran.
E-mail: momenzadeh.hamid@gmail.com Phone: +989166810031

***Abstract*—Detect Intrusion is an indispensable part of a security system. Activity of network becomes an essential part in modern life; on the other hand, number of threats and attacks in private and corporate networks are increasing. Therefore, there is a need for a performance method for detecting of intrusion networks. Detect of intrusion is defined by the problem of identifying misuse and abuse in computer systems. In this paper, a novel fuzzy-evolutionary system is presented to effectively detect the intrusion in networks at computer. So this scheme employs a hybridization of fake annealing empirical and tabu search procedure to recover the accuracy of fuzzy if-then guidelines as intrusion gauges. Both of these systems have its beneficial and detrimental. So, using the cross models of both routes, the anticipated classification occupations the good structures of them to progress the exactness of gained rules. Valuation of the future system is done on the dataset which has info about standard and intrusive activities in linkages. Rests of our archetypal have been matched with quite a few well-known intrusion recognition systems.**

***Keywords*—component; Intrusion detection; simulated annealing; search; fuzzy if-then rules.**

I. Introduction

Intrusion detection systems, aptly called the second line of defense, play a key role in providing comprehensive security. So Due to the rapid growth of computer networks in recent decades, its security has been the one of the most important features to be handled. An intrusion is defined as any set of actions that try to compromise the integrity, confidentiality or availability of a resource [1].

Prevention of intrusion is not sufficient as the system becomes more complex and security mechanisms become weak. At the previous years due to the greater usage of smart devices and internet, the grid circulation increases rapidly.

There are many techniques to detect intrusions effectively and efficiently; so the security goals of a system -confidentiality, integrity, and availability- could be satisfied. Researchers have been working on finding answers to the following questions: How to find attacks effectively, which responses to give against detected attacks, how to proximate adapt to new attack strategies, and such like. The anomaly detector monitors network divides to compare their state to the normal baseline and looks for anomalies. Selected zones, such as recognition approaches, have already been far studied; there are only a few glances on areas such as try and valuation, and response.

Data mining methods for detect intrusion were first implemented in mining audit data for automated models for intrusion detection [2]. Neural networks have been extensively played to detect both misuse and anomalous patterns [3]. Also, some recent prowls have utilized Artificial Immune Systems to detect intrusive behaviors in a computer network [4, 5]. In addition, some other applied techniques on intrusion discovery problem are genetic algorithms [6], Bayesian parameter estimation [7] and clustering [8-10]. As in [11], the studies on this immature research area have accelerated in recent years, and a survey on adversarial Sally against IDSs was recently proposed.

A system that evolves attack signatures, by using GP, is presented in [12]. Another signature-based intrusion detection was proposed recently [13].

There are two different architectures considered: stand-alone, distributed and cooperative architecture in the neighborhood. Other architecture proposals are in [14]. Recently, the adversarial capabilities against intrusion detection networks (IDNs) are presented in [15]. Their adaptation ability to the concept drift correlate on the accuracy of the detector. Therefore, authors Pointed that misclassified instances included in updating could considerably decrease the efficiency of anomaly-based detection approaches [16].

As in [12], employ GP for differentiate malicious peers from benign ones in peer-to-peer (P2P) networks. So, they play some simulations for each individual to see how derived solutions are effective in preventing malicious peers from participating in the networks [17].

A recent GA application plays on principal components instead of working on features for both increases the performance of SVM and to use less number of features. Principal Component Analysis (PCA) computes components of principal, a conventional method for feature subset selection [18].

It is evidence that optimal monitoring node selection problem is NP-hard [19]. Also, another important contribution of IDS deployment on MANETs by using GA is presented in [20]. The applications of evolutionary computation mainly focus on evasion attacks as covered in the Foundations section [21]

In recent years, computational intelligence methods have attracted a considerable interest due to their characteristics suitable for detecting of intrusion such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information [22].

In [23] Chen et al, discovers the applicability of GA with multi-objective optimization methods for place IDS sensors by satisfying conflict objectives. Various attack detection rates are traded-off with false alarm rates and costs [23].

Fuzzy if-then systems rules have been successfully employed in many areas [24]. Also recently, fuzzy rule-based systems have been applied to classification problems, where non-fuzzy input vectors are to be assigned to one of a given set of classes. Fuzzy if–then rules were traditionally profit from human experts. Recently, various methods have been proposed for automatically generating and calibrating fuzzy if–then rules, without using the support of human experts [25]. Also, one of the challenges that is key, in building fuzzy systems is to guarantee that it can automatically extract optimal classification rules from training data, and the extracted rules must be accurate and interpretable for human comprehension.

Simulated Annealing (SA) is an iterative search algorithm motivated by the annealing of metals which was firstly proposed by Metropolis in 1953 [26]. The first effort to take simulated annealing into optimization problems was by Kirkpatrick in 1983 [27], who used it as a new optimization search model to escape local optima and expectantly come close to the global optimum. Since then, simulated annealing has been used on a wide range of optimization problems and attained fine results.

So, the suitability of proposed solutions is discussed for each problem. Tabu Search is an iterative search method for combinatorial optimization problems which was presented by Glover [28]. The goal of this algorithm is to obtain a list of forbidden solutions in the neighborhood of a solution to avoid cycling between solutions while allowing a solution, which may degrade the solution although it may help in escaping from the local optima.

In this article, we have used a hybrid version of simulated annealing and Tabu search with a heuristic fuzzy classification model to develop detect of intrusion system based on misuse detection approach. So, the proposed approach tested by the public intrusion detection dataset available at the University of California, Irvine web site which has information about the normal and intrusive behavior on computer networks.

The rest of the paper is as follows. Sections II and III describe the presented fuzzy rules for classification system for intrusion detection based fuzzy classification system for intrusion detection. Empirical results and the comparison of proposed approach are reported in Section IV, Section V is conclusion.

II. FUZZY RULES FOR CLASSIFICATION

We assume that *m* real patterns, $x_p=(x_{p1},x_{p2},\ldots, x_{pn})$, $p=1,2,\ldots,m$, are given as training pattern from the *c* classes. Since the pattern space is $[0,1]^n$, attribute values of each pattern are $x_{pi}\in[0,1]^n$ for $p=1,2,3,\ldots,m$ and $i=1,2,3\ldots,n$.

In the presented fuzzy classification system, we use fuzzy if-then rules of the following form:
R_j: If x_1 is A_{j1}, … and x_n is A_{jn}, then Class C_j by CF=CF_j where R_j is the label of the *j*th fuzzy if–then rule, $A_{j1},\ldots,A_{jn}$ are antecedent fuzzy sets on the unit interval [0,1], C_j is the consequent, and CF_j is the certainty grade of the fuzzy if–then rule R_j. In computer simulation, we use a typical set of linguistic values in fig. 1 as antecedent fuzzy sets.

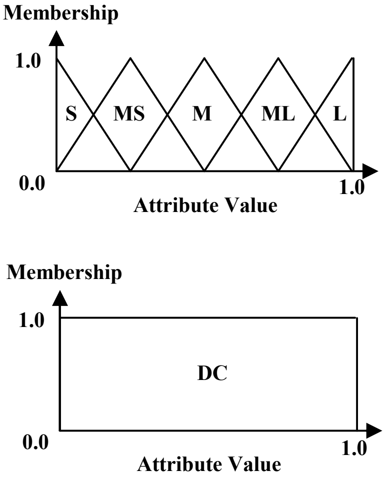

Figure 1. Used antecedent fuzzy sets in this paper. 1: Small, 2: medium small, 3: medium, 4: medium large, 5: large, and 6: don't care.

So, our fuzzy classifier system searches for a relatively small number of fuzzy rules with high classification ability.

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

$$\mu_j(x_p)=\mu_{j1}(x_{p1})\times\ldots\times\mu_{jn}(x_{pn}), \tag{1}$$

where $\mu_{ji}(x_{pi})$ is the membership function of *i*th attribute of *p*th pattern.

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

$$\beta_{Class\ h}(R_j)=\sum_{x_p\in Class\ h}\mu_{R_j}(x_p), h=1,2,\ldots,c \tag{2}$$

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 .

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

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

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 .

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:

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

where

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

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 $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ is classified by a single winner rule R_{j^*} in S , which is determined as follows:

$$\mu_{j^*}(x_p)\ .\ CF_{j^*} = \max\left\{\mu_j(x_p)\ .\ CF_j \mid R_j \in S\right\} \tag{6}$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade CF_j . The next section will discuss about the proposed tabu search based fuzzy system for intrusion detection.

The total number of fuzzy if–then rules is 6^n, so 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). The proposed approach searches for a set of fuzzy rules with the highest classification accuracy. Consequent class and the certainty grade of each fuzzy if–then rule can be determined from training patterns as follows [25]:

Step 1: Calculate the compatibility of training pattern $x_p=(x_{p1},x_{p2},\ldots,x_{pn})$ with the rule R_j by the product operation:

$\mu_j(x_p) \square \mu_{j1}(x_{p1}) \square \ldots \square \mu_{jn}(x_{pn}),\ p \square 1,\ldots, m$

where $\mu_{ji}(x_{pi})$ is the membership function of A_{ji}.

Step 2: For every class, calculate the relative compatibility grades sum of the training patterns with the R_j. That is, the winner rule has the maximum product of the compatibility and the certainty grade.

III. The Proposed Method For MISUSE DETECTION

For this part, outline of the hybrid fuzzy model based on simulated annealing and tabu search is presented in fig. 2. Each step of proposed model is described as follows.

A. Initialization

The method of coding fuzzy if-then rules is as follows: Every fuzzy if-then rule is coded as a string. The following symbols are used for denoting the five linguistic values (Fig. 1). 1.small, 2.medium-small, 3.medium, 4.medium large, 5.large, and 6.DC. For instant, the following fuzzy if-then rule is coded as “36”:

R_j: If x_1 is medium and x_2 is DC then Class C_j with CF=CF_j.

N_{init} denote the numbers of fuzzy if-then rules in the initial set. To create an initial set, one approach is to set:

$$NNCP(S) = m - \sum_{j=1}^{N} NCP(R_j) \tag{7}$$

where N is the number of rules in the rule set, m is the number of every patterns in the training set, and *NNCP(S)* is the number of non-correctly classified training patterns by *S*.

B. One-Point Crossover

A pair of fuzzy if-then rules is choice from the current population to production new fuzzy if-then rules for the next population. Every fuzzy if-then rule in the current population is choices with the following selection probability:

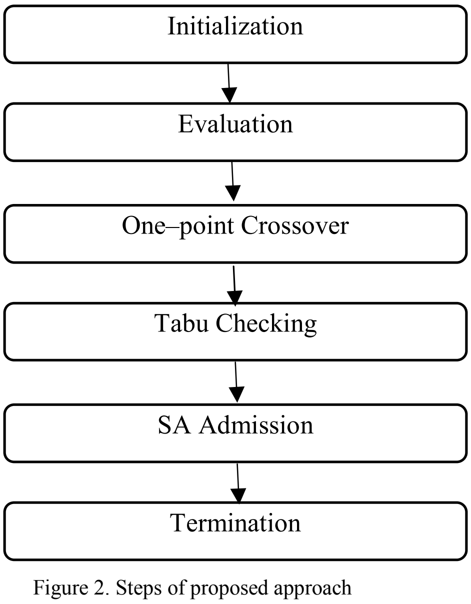

Figure 2. Steps of proposed approach

Here, we increase the probability of DC, when specifying the antecedent fuzzy sets to create general rules to cover larger decision area of the state space. Then generation of N_{init} fuzzy if-then rules as initial set of rules.

S_{init}, the consequent class and the certainty grade of each are specified from training patterns by forms described in section II.

The hybrid model needs to start from a high temperature. If this initial temperature, Tmax, is too high, it causes a waste of processing time. The initial temperature value should be such that it allows all

TABLE I
CLASSES IN THE 10% OF THE KDD-CUP 99 DATA SET

SAMPLES	SUB-CLASSES	CLASS
97278 (19.6911%)		NORMAL
4107 (0.8313%)	ipsweep, nmap, portsweep, satan	PRB
391458 (79.2391%)	back, land, neptune, pod, smurf, teardrop	DOS
52 (0.0105%)	buffer_overflow, loadmodule, multihop, perl, rootkit	U2R
1126 (0.2279%)	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster	R2L

presented good or bad moves to be accepted.

An important parameter is to choose proper tabu list size, a problem-dependent parameter, since the choice of a large size would be inefficient in terms of memory capacity and the time required for scanning the list. On the other hand, choosing a small size would result in a cycling problem; that is, revisiting the same state again.

C. Evaluation

Set of fuzzy if-then rules should have high accuracy. Thus, we use the following function to evaluate each rule set. Where $fitness_{\min}(S)$ is the minimum fuzzy if-then rules fitness value of in the population. Until a pre-specified number of pairs of fuzzy if-then rules are selected this procedure is iterated. A crossover operation is applied to a selected random of fuzzy if–then rules. Note that the selected fuzzy if-then rules for crossover operation should be different. In computer simulations of this paper, we used the one-point crossover in Fig. 3.

D. Tabu Checking

If the new rule set S_{new} is better than the best rule set found so far S_{best}, it is accepted and saved as the best found by setting S_{best}=S_{new}. If the new rule set is better than the current rule set $S_{current}$, it is accepted and saved as the current rule set by setting $S_{current}$=S_{new} . If the new rule set is not better than the current rule set, and the new generated rule.

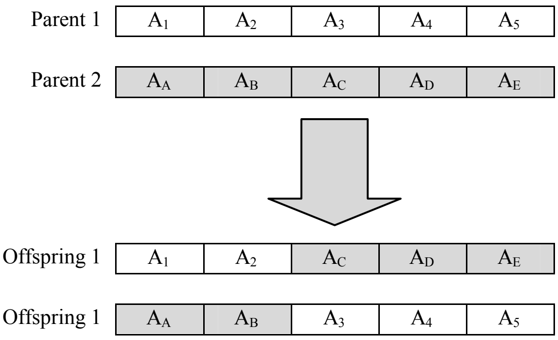

Figure 3. One-point crossover. Note that the cutoff point is determined randomly and the length of parents and offspring's are the same.

R_{new} is not in a direction within the tabu list TL, it is accepted as the current rule set and the search continues from there. If the R_{new} is in tabu list, the current rule set remains unchanged and a new rule set is generated. After accepting a new rule set, TL is updated and contains the new generated rule R_{new} to forbid returning to this rule.

E. SA Admission

If the new rule set is better than the current rule set S current, it is accepted and saved as the current rule set by setting $S_{current}$=S_{new}. If the new rule set S_{new} is better than the best rule set found so far Sbest, it is accepted and saved as the best found by setting S_{best}=S_{new}. If the new rule set is not better than the current rule set, ISAF will accept the new set of rules on a probabilistic basis. A digit by random numbers is generated in the range 0 to 1. If this random number is smaller than value given by (10) the new set of rules is accepted.

As shown in Table I, the digits of records in the 10% data set is very large (494021). So, share of trials per class is not unbroken, for sample from class DOS the number of samples is 391458 from class U2R the digit of samples in the training facts set is 52 while. According to this fact, we have recycled a subset of this large data set as our train and test data sets; hence, the exercise data set comprehends 20752 arbitrarily spawned tasters. The distribution of different classes in the train and test data sets is presented in Table II.

F. Termination

At each temperature, the inner loop of proposed approach is called a constant number of times. Cooling rate parameter α, used for updating the temperature. When the temperature reaches to the minimum temperature, the algorithm terminates.

IV. Experimets

Experiments were in the framework of the 1998 Intrusion Detection Evaluation Program and carried out on a subset of the database created by DARPA. We distributed as part of the UCI KDD Archive used the subset that was pre-processed by the Columbia University and [29]. The available database is made up of malicious traffic a large and number of network connections related to normal. Every connection is represented with a 41-dimensional feature vector. Connections are also labeled as belonging to one out of five classes. One of these classes is the rest indicates four different intrusion classes and the normal class. These intrusion classes are classification of 23 different types of spells in a computer network.

We consigned the train and test data sets, that each numerical value in the data set is normalized between 0.0 and 1.0 according to the following equation:

$$\chi_{Normalized} = \frac{\chi - \chi_{\min}}{\chi_{\max} - \chi_{\min}} \qquad (8)$$

42 numeric features are constructed and consigned to the interval [0, 1]. This section consists of two subsections. First, we present some experiments of applying proposed approach to the intrusion detection classification problem. at next subsection we compare the

TABLE II
DISTRIBUTION OF DIFFERENT CLASSES IN THE TRAIN AND TEST SETS

Test	Train	Class
60593	10000	NORMAL
4166	4107	PRB
229853	5467	DOS
228	52	U2R
16189	1126	R2L

performance of our algorithm to some of other intrusion detection algorithms.

The average accuracy rate obtained from proposed model varies from 93% to 98% with the number of rules ranging from 50 to 100. The hybrid model which extracts 67 fuzzy if-then rules with the average rule length of 5.54 from training data and obtains 97.81% accuracy rate is applied for test validation. Table IV is the confusion matrix of hybrid model. The top -left entry of Table III of the actual Normal test set were detected to be Normal shows that 59976 instances.

Classification performance of TSFS is measured and compared with that of different classification algorithms including pruning C4.5, Naïve Bayes (NB), *k*-Nearest Neighbor (*k*-NN), Support Vector Machine (SVM), and Multi-Objective Genetic Fuzzy Intrusion Detection System (MOGFIDS) [7]. In *k*-NN parameter by using the well-known fast sequential minimal optimization method with a polynomial kernel *k* is set to 5, and the SVM is trained. The results compare with the winner of KDD-Cup99 contest [30].

Precision and f-measure of algorithms for PRB class. By to this table, we can see that the model obtains of hybrid the best value for precision and f-measure. According to the above discussion, it shows that our proposed model obtains comparative results in comparison with several famous intrusion detection algorithms.

V. Conclusions

In this paper, we have introduced a novel hybrid model for intrusion detection in computer networks. This model used a fuzzy classification system armed with a hybrid heuristic algorithm based on simulated annealing and tabu search algorithms. Simulations of computer on data sets demonstrate presented method achieves robust performance for intrusion in normal traffic. Results of proposed model on KDDCup99 dataset, which has information about normal and intrusive behaviors in networks, show that the hybrid model achieves good outcomes in comparison with several well-know algorithms in this field.

TABLE III
CONFUSION MATRIX FOR HYBRID MODEL

Detected Class						
recall	R2L	U2R	DOS	PRB	Normal	Real-Class
98.98	196	54	39	236	59976	Normal
84.18	3	17	121	3507	364	PRB
98.32	53	14	225992	94	2570	DOS
17.98	7	41	1	18	143	U2R
13.94	2258	7	3596	17	9981	R2L
93.80	89.70	30.82	98.36	90.57	82.12	precision

References

[1] R. Heady, G. Luger, A. Maccabe, and M. Servilla, “The architecture of network level intrusion detection system,” Technical Report, Department of Computer Science, University of New Mexico, 1990.

[2] Barbara D, Couto J, Jajodia S, Wu N. ADAM: a testbed for exploring the use of data mining in intrusion detection. SIGMOD Rec 2001;30(4):15–24.

[3] 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.

[4] 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.

[5] 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.

[6] C.H. Tsang, S. Kwong, H. Wang, “Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection,” Pattern Recognition, Volume 40, Issue 9, September 2007, pp. 2373-2391.

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

[8] 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.

[9] 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.

[10] 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.

[11] Corona, I., Giacinto, G., Roli, F. (2013). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences, Vol. 239, pp. 201-225.

[12] Lu, W, Traore, I, (2004) “Detecting New Forms of Network Intrusion Using Genetic Programming”. Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494.

[13] Gomez, J., Gil, C., Banos, R., Marquez, A.L., Montoya, F.G., Montoya, M.G. (2013). A Pareto-based multi-objective evolutionary algorithms for automatic rule generation in network intrusion detection systems, Soft Computing, Vol. 17, pp. 255-263.

[14] Hassanzadeh, A., Stoleru, R. (2013). On the optimality of cooperative intrusion detection for resource constrained wireless networks, Computers & Security, Vol. 34, pp. 16-35.

[15] Pastrana, S. (2014). Attacks against intrusion detection networks: evasion, reverse engineering and optimal countermeasures. PhD Thesis, Computer Science and Engineering Department, Universidad Carlos III De Madrid,

[16] Sen, S. (2014). Using instance weighted naive Bayes for adapting concept drift in masquerade detection. International Journal of Information Security.

[17] Tahta, U.E., Can, A.B., Sen, S. (2014). Evolving a trust model for peer-to-peer networks using genetic programming, In Proc. of EvoStar, LNCS Series, Springer. (to appear)

[18] Ahmad, I., Hussain, M., Alhgamdi, A., Alelaiwi, A. (2014). Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. Neural Computing and Applications, Vol. 24, Issue 7-8, pp. 1671-1682.

[19] Hassanzadeh, A., Stoleru, R., Shihada, B. (2011). Energy efficient monitoring for intrusion detection in battery-powered wireless mesh networks. In Proc. of ADHOC-NOW, LNCS 6811, Springer, pp. 44-57.

[20] Hassanzadeh, A., Stoleru, R. (2011b). Towards optimal monitoring in cooperative ids for resource constrained wireless networks. In Proc. of the 20th International Conference on Computer Communications and Networks.

[21] Kayacik, H. G., Zincir-Heywood, A. N., Heywood, M. I. (2011). Evolutionary computation as an artificial attacker: generating evasion attacks for detector vulnerability testing. Evolutionary Intelligence, Vol. 4, issue 4, pp. 243-266.

[22] Wu, S. X., Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: a review, Applied Soft Computing, Vol. 10(1), pp. 1-35.

[23] Chen, H., Clark, J.A., Shaikh, S.A., Chivers, H., Nobles, P. (2010). Optimising IDS Sensor Placement, In Proc. of International Conference on Availability, Reliability, and Security.

[24] C.C. Lee, “Fuzzy logic in control systems: fuzzy logic controller, Part I and Part II,” IEEE Transactions on Systems, Man, and Cybernetics, 20(2), 1990, pp. 404–435.

[25] H. Ishibuchi, K. Nozaki, and H. Tanaka, “Distributed representation of fuzzy rules and its application to pattern classification,” Fuzzy Sets and Systems, 52(1), 1992, pp. 21-32.

[26] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, “Equation of state calculation by fast computing machines,” Journal of Chemical Physics, vol. 21, pp. 1087-1092, 1953.

[27] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, pp. 671-680, 1983.

[28] F. Glover, M. Laguna, “Tabu Search,” Kluwer Academic Publishers, Dordrecht, 1997.

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

[30] C. Elkan, “Results of the KDD'99 classifier learning,” ACM SIGKDD Explorations 1, 2000, pp. 63–64.