Security Audit Trail Analysis Using Harmony Search Algorithm

Mourad Daoudi
Faculty of Electronics and Computer Science, Laboratory LSI, USTHB
BP 32 16111 El Alia, Bab-Ezzouar,
Algiers, Algeria
m-daoudi@usthb.dz

Abstract—**Security Audit trail Analysis can be accomplished by searching audit trail logs of user activities for known attacks. The problem is a combinatorial optimization problem NP-Hard. Metaheuristics offer an alternative to solve this type of problems. In this paper, we propose to use Harmony Search metaheuristic as intrusion detection engine. It is a population-based evolutionary algorithm well suited for constrained optimization problems. Experimental results for simulated attacks are reported. The effectiveness of the method is evaluated by its ability to make correct predictions. Our new approach has proven effective and capable of producing a reliable method for intrusion detection.**

Keywords-metaheuristics; harmony search; evolutionary Algorithm; intrusion detection; security audit.

I. INTRODUCTION

Intrusion Detection System (IDS) is one of the primary approaches to the crucial problem of computer security in order to preserve the confidentiality, integrity and availability of data stored in computers [1], [2].The study of effective methods for intrusion detection by audit file analysis is an important part of the vast effort to improve the security of computer systems. Different methods, like Neural Networks, Immune Systems, and Genetic Algorithms have been developed. Intrusion detection by security audit trail analysis can be performed by searching audit trail logs of user activities for predefined attacks. Because the problem is a combinatorial optimization problem NP-Hard [3], heuristic methods will need to be used as data bases of events and attacks grow.

This paper presents a method to perform misuse detection using Harmony Search metaheuristic (HS) [4], under the guidelines of previous works [5]-[10].

Harmony Search algorithm is a population-based evolutionary algorithm taking inspiration from the music improvisation process, where musicians improvise their instruments' pitches searching for a perfect state of harmony. It has proven its abilities in solving various optimization problems [11], [12]. We define HS as an intrusion detection engine, and then evaluate its performances.

The rest of the paper is as follows: Section 2 outlines intrusion detection systems. A formalization of security audit trail analysis problem as a combinatorial optimization problem is given in Section 3. HS metaheuristic is presented in Section 4. Our contribution using HS for misuse detection is presented in Section 5. Experimental results are reported in Section 6. Comparisons with a biogeography inspired intrusion detection approach are performed and experimental results are presented in Section 7. Our conclusions are given in Section 8.

II. INTRUSION DETECTION SYSTEM

An intrusion is defined as a series of actions that attempt to compromise the integrity, confidentiality or availability, or to bypass the security mechanisms of a computer or network. Any operation undertaken on the computer system results in a sequence of actions performed by the system called system activities. A system activity occurring at a point of time is called event. These events are recorded chronologically in a log file.

Intrusion Detection Systems usually process log records received from the operating system for a specific period of time in order to have a complete set of user activity and then perform analysis of the current activity [13],[14]. According to Intrusion Detection Working Group of IETF [15], IDSs include three vital functional elements: information source, analysis engine and response component. Five concepts are defined for their classification: the detection method, the behavior on detection, the audit source location, the detection paradigm and the usage frequency. The detection method is one of the principal characters of classification. When the IDS uses information about the normal behavior of the system it monitors (anomaly detection or behavioral approach), we qualify it as behavior-based. When the IDS uses information about the attacks (misuse detection or scenario approach), we qualify it as knowledge-based. However, both approaches may be complementary [16].

The behavioral approach was first proposed approach, introduced by Anderson [17], and extended by Denning[18]. It is to define a profile of normal activity of a user, and consider the significant deviations of the current activity of users, compared with normal patterns of behavior, as an anomaly.The techniques developed in behavioral approach include the expert systems, statistical models [19],and artificial immune systems [20], [21]. Other techniques like Bayesian parameter estimation [22] and clustering [23]-[26], Genetic Algorithms[27], [28] are also used.

In Misuse detection, IDS processes log records from the operating system for a specific period of time in order to have a complete set of user activity [19]. Then, the IDS performs analysis of the current activity, using a rule base system, statistics, or a corresponding heuristic, in order to determine the possible occurrence of intrusion. The misuse mechanism uses a predefined matrix of intrusion patterns, so the system knows in advance the appearance of the misuse and/or abuse.

An intrusion can be specified by an array of activities to check, where each entry specifies the number of activities of a specific type that should occur in order to have an intrusion. Likewise, the results of the user records gathered can be seen as an array, where each entry specifies the total number of activities of that specific type performed by a user. If an intrusion array pattern is such that each entry of it is less or equal than each entry of the user's activity, then, it is possible that intrusion H has occurred. However, looking at some possible intrusions together, it is possible that one or several can occur, but not all together, because adding each corresponding entry, some results could be greater than the corresponding entry of the user activity vector. This is called a violation of the constraint. Neural networks have been extensively used to detect both misuse and anomalous patterns [29]-[31].

We investigate in what follows Security Audit Trail Analysis Problem.We give in the next section the formulization of the problem.

III. SECURITY AUDIT TRAIL ANALYSIS PROBLEM

Formally, the Security Audit Trail Analysis Problem can be expressed by the following statement [5]:

Let:

- N_e the number of type of audit events
- N_a the number of potential known attacks
- AE an N_e x N_a attack-events matrix which gives the set events generated by each attack. $AE_{i\ j}$ is the number of events of type i generated by the attack j ($AEi\ j \geq 0$)
- R a N_a-dimensional weight vector, where R_i ($R_i > 0$) is the weight associated with the attack i (R_i is proportional to the inherent risk in attack scenario i)
- O a N_e-dimensional vector, where O_i is the number of events of type i present in the audit trail (O is the observed audit vector)
- H a N_a-dimensional hypothesis vector, where H_i = 1 if attack i is present according to the hypothesis and H_j= 0 otherwise (H describes a particular attack subset).

To explain the data contained in the audit trail (i.e. O) by the occurrence of one or more attacks, we have to find the H vector which maximizes the product RxH (it is the pessimistic approach: finding H so that the risk is the greatest), subject to the constraint $(AE.H)_i \leq O_i$, $1 \leq i \leq N_e$ (Fig. 1).

Because finding H vector is NP-Complete, the application of classical algorithm is impossible where N_a equals to several hundreds.

The heuristic approach that we have chosen to solve that NP-complete problem is the following: a hypothesis is made (e.g. among the set of possible attacks, attacks i, j and k are present in the trail), the realism of the hypothesis is evaluated and, according to this evaluation, an improved hypothesis is tried, until a solution is found.

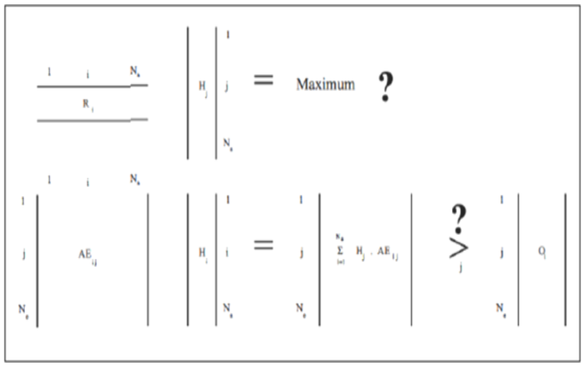

Figure 1. Security Audit Trail Analysis Problem.

In order to evaluate a hypothesis corresponding to a particular subset of present attack, we count the occurrence of events of each type generated by all the attacks of the hypothesis. If these numbers are less than or equal to the number of events recorded in the trail, then the hypothesis is realistic.

To derive a new hypothesis based on the past hypothesis,we propose to use HS algorithm we present in the next section.

IV. HARMONY SEARCH METAHEURISTIC - AN OVERVIEW

Harmony Search was devised as a new metaheuristic algorithm, taking inspiration from the music improvisation process, where musicians improvise their instruments' pitches searching for a perfect state of harmony. Analogies with optimization process are such that:

- Instrument i ↔ Decision variable x_i , i ϵ{1,2,…,n}
- Music note from instrument i ↔Value of variable x_i
- Harmony ↔ Solution vector
- Musical easthetic ↔ Fitness

The HS metaheuristic algorithm consists of the following steps [12]:

A. *Initialization of the Optimization Problem and Algorithm Parameters*

In the first step, the optimization problem is specified as follows:

Minimize (or Maximize) f(x)
subjected to $x_i \in X_i$, i= 1, 2, . . .,n.

where f(.) is a scalar objective function to be optimized; x is a solution vector composed of decision variables $x_{i,}$ i= 1, 2, . . ., n; X_i is the set of possible range of valuesfor each decision variable x_i , that is, ${}_Lx_i \leq X_i \leq {}_Ux_i$, where ${}_Lx_i$ and ${}_Ux_i$ are the lower and upper bounds for each decision variable in the case of continuous decision variables and $x_i \in \{x_i(1), \ldots, x_i(k), \ldots, x_i(K)\}$, when the decision variables are discrete. N is the number of decision variables.

In addition, the control parameters of HS are also specified in this step. These parameters are the Harmony

Memory Size (HMS) i.e., the number ofsolution vectors (population members) in the HM (in each generation); the HM considering rate (HMCR); the pitch-adjusting rate (PAR) and the number ofimprovisations (NI) or stopping criterion.

B. *Harmony Memory Initialization*

In this step, each component of each solution vector in the parental population (HM) is randomly chosen. Then, obtained solutions are reorderedin terms of the objective function value:f $(x^1) \leq f(x^2) \ldots \leq f(x^i) \ldots \leq f(x^{HMS})$, where $x^i = (x^i_1, x^i_2, \ldots, x^i_n)$ is the ith solution vector.

Harmony Memory is represented by the following matrix:

$$HM = \begin{bmatrix} x^1_1 & x^1_2 \ldots & \ldots & x^1_n \\ x^2_1 & x^2_2 \ldots & \ldots & x^2_n \\ \vdots & \ldots & \ldots & \ldots \\ x^{HMS}_1 & x^{HMS}_2 & \ldots & x^{HMS}_n \end{bmatrix}$$

C. *New Harmony Improvisation*

In this step, anew harmony vector x'= $(x'_1, x'_2, \ldots, x'_n)$ is generated based on three rules: memory consideration; pitch adjustment and random selection. Generating a new harmony is called 'improvisation'.

The memory consideration rule stipulates that the decision variable $x_{i,}$,i= 1…n, takes a new value $x'_i$from HM matrix (in the set $\{x^1_i, x^2_i, \ldots, x^{HMS}_i\}$), with a probability HMCR (parameter value between 0 and 1), and takes a fresh value randomly selected from the set Xi with probability (1-HMCR). This rule can be summarized in equation (1).

$$x'_i \leftarrow \begin{cases} x'_i \in (x^1_i, x^2_i, \ldots, x^{HMS}_i) \text{with probability HMCR} \\ x'_i \in Xi \quad \text{with probability } (1 - \text{HMCR}) \end{cases} \quad (1)$$

Every component obtained in the memory consideration step is further examined to determine whether it should be pitch adjusted. This operation uses the parameter PAR (pitch adjustment rate) as follows:

- In the case of discrete variables:$x'_i = x_i(k)$

$$x_i \leftarrow \begin{cases} xi(k+m) \quad \text{with probability PAR} \\ x_i \quad \text{with probability } (1 - \text{PAR}) \end{cases} \quad (2)$$

where : m∈ {-1, 1}.

- In the case of continuous variables:

$$x_i \leftarrow \begin{cases} x_i \pm \text{rand}(0\ 1) * \text{bw with probability PAR} \\ x_i \quad \text{with probability } (1 - \text{PAR}) \end{cases} \quad (3)$$

where bw is an arbitrary distance bandwidth (a scalar number), and rand() is a uniformly distributed random number between 0 and 1. Evidently, the new harmony improvisation strategy is responsible for generating new potential variation in the algorithm and is comparable to mutation in standard evolutionary algorithms.

D. *Harmony Memory update*

If the new harmony vector x'= $(x'_1, x'_2, \ldots, x'_N)$ is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM, and the existing worst harmony is excluded from the HM. This is actually the selection step of the algorithm, where the objective function value is evaluated to determine if the new variation should be included in the population (HM).

E. *Check Stopping Criterion*

If the stopping criterion (maximum NI) is satisfied, the computation is terminated. Otherwise, steps *C* and *D* are repeated.

V. HARMONY SEARCH-BASED SECURITY AUDIT TRAIL ANALYSIS

The approach aims to determine if the events generated by a user correspond to known attacks, and to search in the audit trail file for the occurrence of attacks by using a heuristic method (HS algorithm) because this search is an NP–complete problem. The goal of the heuristic used is to find the hypothesized vector H that maximizes the product R*H, subject to the constraint $(AE.H)_i \leq O_i$,$1 \leq i \leq N_e$, whereR is a weight vector that reflects the priorities of the security manager, AE is the attack-events matrix that correlates sets of events with known attacks, and N_e the number of types of audit events.

HS method is based on a stochastic optimization process and on the musical performance process of finding the perfect harmony in a musical orchestra where each musician plays a note to find a better harmony. We are in a situation where the coding of harmonies is immediate since the solution of the problem is expressed specifically in the form of a binary sequence. A Harmony is considered as a series of N_a notes with values 0 or 1.The Harmony Memory matrix is thus a matrix of dimension HMS x N_a.Each harmony (that is a line vector in HM matrix) is a particular instance of the vector H. In other words, the note i in the harmony will be 1if the harmony is a solution in which the attack i is declared present. Otherwise, this note takes the value 0. We note that the sum of the component elements in a harmony indicates the number of attacks that are detected.

The HS method should return a binary N_a –vector H = $(H_1, \ldots, H_{Na})$, where the value H_i=1 in the ith position indicates the presence of the attack I in the audit file O and 0 its absence. As we have to solve a maximization problem, the best solution is associated with the hypothesis H of larger value of the selective function

$$F(H) = R * H = \sum_1^{N_a} R_j * H_j \quad (4)$$

which represents the total risks incurred by the system under surveillance. In addition, as we deal with a constrained problem, any solution to the problem must verify the inequalities: $(AE \times H)_I \leq O_i$ avec $0<i<n_e$. Thus, we have to eliminate the solutions that do not comply, setting to zero the value of the corresponding objective function (that is a

harmony in which each note has a zero value). There combination process is repeated until a solution satisfying the constraints is generated. Recall that R_i is the weight of the attack i, that is the risk incurred to the system if the attack is not detected. For simplicity, the R_i value is taken equal to 1 for all i, i=1 … N_a. In this case, the objective function F resumes in computing the number of detected attacks.

It is now necessary to correctly represent the matrix HM. It is a ($HMS*N_a$) - matrix, where N_a is the number of predefined attacks and HMS the population size parameter. We associate to each row i of the matrix HM (a solution vector to our problem) a value of the objective function F. The different parameters HMS, HMCR, PAR and NI must be initialized. Then, HMS initial solution vectors are randomly generated and saved in the HM matrix. The fitness value F(i), corresponding to the ith solution i = 1…HMS, is evaluated. Then, the optimization process evolves. It generates a new solution $x = (x_1, ..., x_n)$ from the HM matrix, using the parameters HMCR and PAR. These parameters help the algorithm to obtain better solutions locally or globally.

VI. EXPERIMENTATION RESULTS

During the simulations, all the attacks actually present in the analyzed audit file must be known in advance. Thus, the events corresponding to one or more attacks are included in the observed audit vector O. Each of the experiments conducted is characterized by the 5-tuple (NI, HMS, HMCR, PAR, Ia), where NI is the number of generations, HMS the population size, HMCR the harmony memory consideration rate, PAR the adjustment rate and Ia the number of attacks actually present in the audit file (number of attacks injected).

Ratios TPR, FPR, Accuracy and Precision [32] are used to evaluate our approach intrusion detection quality, with:

- TPR (True positive rate): TP / (TP+FN)
- FPR (False positive rate): FP / (TN+FP)
- Accuracy: (TN+TP) / (TN+TP+FN+FP)
- Precision: TP / (TP+FP)
-

where True negatives (TN) as well as true positives (TP) correspond to correct intrusion detection: that is, events are successfully labeled as normal and attacks, respectively. False positives (FP) refer to normal events being predicted as attacks; false negatives (FN) are attack events incorrectly predicted as normal events.

To evaluate our approach, many tests were performed using Attack-Events matrices of different sizes. All results are obtained as the average of 10 executions carried out for the same better value of the 5-tuple (NI, HMS, HMCR, PAR, I_a).

We observe that after a certain number of generations, all injected attacks are detected, and no false attack (TPR= 1 and FPR =0). In addition, the number of attacks injected has no influence on these results. Best results are obtained with HMS value between 20 and 30. Further, the study of the influence of the two rates HMCR and PAR is important because they contribute significantly in the algorithm in finding the best solution. There is a strong correlation between HMCR and PAR on the Harmony Search metaheuristic optimization process [33]. To study their influence on the quality of intrusion detection, we varied the parameter PAR between 0.1 and 0.9 with a step of 0.2; HMCR is selected it in the set {0, 0.3, 0.5,0.7, 0.9,0.98}. We observe that the best results are obtained for values of HMCR and PAR, such as: $0.8 \leq HMCR \leq 0.98$ and $PAR \geq 0.3$.

Reported results concern tests performed on an attack-events matrix of size (28x24) issued from [5] and on larger data randomly generated (AE-matrices of larger size). Table I and Table II show the quality of our intrusion detection approach.

Data considered in Table I, are issued from [5], and use an attack-events matrix of size 28x24, with 24 attacks and 28 types of events. The HS parameters values are: HMS = 30,HMCR= 0.98, PAR =0.3, and 24 attacks are injected. We observe that all attacks are detected after 0.056 seconds (TPR=100% and FPR=0).

The results given in Table II concern tests performed on data with an attack-events matrix of size (100x200). The different parameters are such that: HMS = 30,HMCR= 0.98, PAR =0.3. The number of injected attacks is200.

TABLE I. INTRUSION DETECTION QUALITY (28x24)

Generation number (NI)	Execution time (sec)	TPR %	FPR %	Exactitude %	Precision %
1	0.002	66.67	0	66.67	100
5	0.003	70.83	0	70.83	100
50	0.020	83.83	0	83.83	100
100	0.037	91.67	0	91.67	100
200	**0.056**	**100**	**0**	**100**	**100**
220	0.064	100	0	100	100
250	0.077	100	0	100	100

TABLE II. INTRUSION DETECTION QUALITY (100x200)

Generation number (NI)	Execution time (sec)	TPR %	FPR %	Exactitude %	Precision %
0	0.010	57	0	78.3	100
50	0.030	66	0	83	100
100	0.043	71	0	85.5	100
200	0.075	82	0	91	100
400	0.135	94	0	97	100
500	0.165	97	0	98.5	100
600	0.196	99	0	99.5	100
700	**0.225**	**100**	**0**	**100**	**100**
750	0.238	100	0	100	100
800	0 .251	100	0	100	100

All attacks are detected after 0.225 seconds (TPR=100% and FPR=0).

The results indeed show the good performance of the proposed approach. An important result was the consistency of results, independently of the number of attacks actually present in the analyzed audit file. This means that the performance of the detection system is not deteriorated in the case of multipleattacks. The execution time is very satisfying.

An intrusion detection approach based on security audit trail analysis usinga “Biogeography Based Optimization” algorithm (BBO) has been developed [9], [10]. It showed good performances. Comparisons with our new method are made.

VII. COMPARISONS WITH A BIOGEOGRAPHY INSPIRED INTRUSION DETECTION APPROACH

Biogeography-based optimization (BBO) was first presented by D. Simon in 2008 [34]. In BBO, problem solutions are represented as islands, and the sharing of features between solutions is represented as migration between islands. Islands that are well suited as habitats for biological species are said to have a high island suitability index (ISI). Features that correlate with ISI include rainfall, topographic diversity, area, temperature, etc. The variables that characterize these features are called suitability index variables (SIVs). SIVs are the independent variables of the island, and ISI is the dependent variable. Islands with a high ISI tend to have a large number of species, and those with a low ISI have a small number of species. Islands with a high ISI have many species that emigrate to nearby islands because of the accumulation of random effects on its large populations. Emigration occurs as animals ride flotsam, fly, or swim to neighboring islands. Suppose that we have some problem, and that we also have several candidate solutions. A good solution is analogous to an island with a high ISI, and a poor solution is like an island with a low ISI. High ISI solutions are more likely to share their features with other solutions, and low ISI solutions are more likely to accept shared features from other solutions. An intrusion detection approach with BBO has been developed [9]. We compare the latter with our new method. Several tests are performed on different data using the same machine.

The results reported in Table III concern:

- a (28x24) - Attack-Events matrix (noted T1) issued from [5] with NA=24 attacks injected,
- a (100x300)- Attack-Events matrix (noted T2) with NA=300 attacks injected

a (100x700) - Attack-Events matrix (noted T3) with 700 attacks injected.

ET represents the execution time in seconds.

TABLE III. HS vs. BBO-BASED APPROACH

	T1		**T2**		**T3**	
	NA	ET	NA	ET	NA	ET
BBO	24	0.109	300	9.040	700	498
HS	24	**0.056**	300	**3.773**	700	**7.125**

The different parameters involved in BBO algorithm are set to: Mutation rate = 0.005, Population size = 50 and Elitism value = 2. In our HS based method, the parameters values are: HMS = 30,HMCR= 0.98 andPAR =0.3.

In both approaches, all attacks are detected. However, we clearly observe that the intrusion detection process duration for all attacks is better using our new approach, and the difference increases as the AE matrix size grows.

VIII. CONCLUSION

Security Audit trail Analysis can be accomplished by searching audit trail logs of user activities for known attacks. The problem is a combinatorial optimization problem NP-Hard. Metaheuristics offer an alternative for solving this type of problem when the size of the database events and attacks grow. We proposed to use Harmony Search metaheuristic as detection engine.

Experimental results of simulated intrusions detection are given. The effectiveness of the approach is evaluated by its ability to make correct predictions. It proved to be effective and capable of producing a reliable method for intrusion detection. The results indeed show the good performance of the proposed approach. An important result was the consistency of results, independently of the number of attacks actually present in the analyzed audit file. This means that the performance of the detection system is not deteriorated in the case of multiple attacks. The execution time is very satisfying. All this allows us to consider that the proposed approach constitutes an efficient and reliable intrusion detection system. Comparisons with a biogeography inspired approach is made. We observe clearly that the intrusion detection duration of all attacks is significantly lower when using our new HS-based approach. However, these systems are usually developed for predefined environments and do not offer a solution to some network characteristics such as changes in behavior of users and services, the increasing complexity and evolution of the types of attacks that they may be subject, the speed of attacks that can occur simultaneously on several machines, etc.

REFERENCES

[1] E. Cole, R. K. Krutz, and J. Conley, Network security bible, Wiley Publishing, 2005.

[2] B. C. Tjaden, Fundamentals of secure computer systems. Franklin and Beedle & Associates, 2004.

[3] L. Mé, Audit de sécurité par algorithmes génétiques. Thèse de Doctorat de l’Institut de Formation Supérieure en Informatique et de Communication de Rennes, 1994.

[4] Z.W. Geem, J.H. Kim, and G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation, vol. 76, no. 2, pp. 60-68, 2001.

[5] L. Mé, GASSATA, A genetic algorithm as an alternative tool for security audit trails analysis. Proc. of the 1st International Workshop on the Recent Advances in Intrusion Detection (RAID 98). Louvain-la-Neuve, Belgium, pp. 14–16, 1998.

[6] P. A. Diaz-Gomez, and D. F. Hougen, Improved off-line intrusion detection using a genetic algorithm. Proc. of the seventh International Conference on Enterprise Information Systems, pp. 66-73, 2005.

[7] P. A. Diaz-Gomez, and D. F. Hougen, A genetic algorithm approach for doing misuse detection in audit trail files. Proc. of International Conference on Computing (CIC-2006), pp. 329-335, 2006.

[8] A. Ahmim, N. Ghoualmi, and N. Kahya, Improved off-line intrusion detection using a genetic algorithm and RMI. Proc. of the International Journal of Advanced Computer Science and Applications (IJACSA), vol. 2, no. 1, 2011.

[9] M. Daoudi, A. Boukra, and M. Ahmed-Nacer, A biogeography inspired approach for security audit trail analysis. Journal of Intelligent Computing, vol. 2, no. 4, December 2011.

[10] M. Daoudi, A. Boukra, and M. Ahmed-Nacer, Security audit trail analysis with biogeography based optimization metaheuristic. Proc. of the International Conference on Informatics Engineering & Information Science: ICIES 2011, A. Abd Manafet al. (Eds.): ICIEIS 2011, Part II, CCIS 252, pp.218–227, 2011.© Springer-Verlag Berlin Heidelberg; 2011.

[11] L. S. Coelho, and D. L. Andrade Bernert, An improved harmony search algorithm for synchronization of discrete time chaotic systems. Industrial and Systems Engineering Program, LAS/PPGEPS, Pontifical Catholic University of Parana PUCPR, Imaculada Conceição, 1155, 80215-901 Curitiba, Paraná, Brazil, 2008.

[12] S.Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, Exploratory power of theharmony search algorithm: analysis andimprovements for global numerical optimization. IEEE. Transactions on systems, man, and cybernetics—part b: cybernetics. 1083-4419/© 2010IEEE, 2010.

[13] R. Bace, and P. Mell, NIST Special publication on intrusion detection systems, 2001.

[14] P. Roberto, and V. Luigi, Intrusion detection systems, ISBN: 978-0-387-77265-3, 2008.

[15] H. Debar, M. Dacier, and A. Wespi, A revised taxonomy for intrusion detection systems. Annales des Telecommunications, 55(7-8), 2000.

[16] E. Tombini, Amélioration du diagnostic en détection d'intrusions: etude et application d'une combinaison de méthodes comportementale et par scénarios. Thèse de Doctorat de l'Institut National des SciencesAppliquées de Rennes, 2006.

[17] J. Anderson, Computer security threat monitoring and surveillance. Technical Report 79F296400, James Anderson, Co., Fort Washington, PA, 1980.

[18] D. Denning, An intrusion detection model. In Proc.of the 1986 IEEE Symposium on Security and Privacy, pp. 118-131, 1987.

[19] R. G. Bace, Intrusion detection systems. Mac Millan Technique Publication, USA, 2000.

[20] Y. Haidong, G. Jianhua, and D. Feiqi, Collaborative RFID intrusion detection withan artificial immune system. Journal of Intelligent Information System, Vol. 36, N°1, pp. 1-26, 2010.

[21] Z. Banković, A. Álvaro, and J.M. de Goyeneche, Intrusion detection in sensor networks using clustering and immune systems. Lecture Notes in Computer Science, Vol. 5788, Intelligent Data Engineering andAutomatedLearning - IDEAL 2009, pp. 408-415.

[22] S. Cho, Web session anomaly detection based on parameter estimation. Computers & Security,vol. 23, no. 4, pp. 265-351, 2004.

[23] B. Xu, and A. Zhang, Application of support vector clustering algorithm to network intrusion detection. Proc. of the International Conference on Neural Networks and Brain, ICNN & B '05. vol. 2, pp. 1036 – 1040, 2005.

[24] O. SH, and L. WS, An anomaly intrusion detection method by clustering normal user behavior. Computers & Security, vol. 22, no. 7, pp. 596-612, 2006.

[25] E. Leon, O. Nasraoui, and J. Gomez, Anomaly detection based on unsupervised niche clustering with application to network intrusion detection. Proc. of the IEEE Conference on Evolutionary Computation (CEC),pp. 502-508, 2004.

[26] Y. Guan, A. Ghorbani, and N. Belacel, Y-MEANS: A clustering method for intrusion detection. Proc. of the Canadian Conference on Electrical and Computer Engineering, pp. 1083-1086, 2004.

[27] 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.

[28] T. Ozyer, R. Alhajj, and K. Barker, Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule prescreening. Journal of Network and Computer Applications , pp. 99–113, 2003.

[29] G. Kumar, and M. Sachdeva, The use of artificial intelligence based techniques for intrusion detection: Artificial Intelligence Review, vol. 34, no. 4, pp. 369-387, 2010.

[30] M. Castellano, G. Mastronardi, and G. Tarricone, (2009). Intrusion detection using neural networks: a grid computing based data mining approach. Lecture Notes in Computer Science, 2009, vol. 5864, Neural Information Processing, pp. 777-785, 2009.

[31] S.Mukkamala, D.Xu,and A.Sung, intrusion detection based on behavior mining and machine learning techniques. Lecture Notes inComputer Science, Vol. 4031. Advances in Applied Artificial Intelligence, pp. 619-628, 2006.

[32] S.Wu, and W. Banzhaf, The use of computational intelligence in intrusion detection systems: a review. Computer Science Department, Memorial University of Newfoundland, St John's, NL A1B3X5, Canada. 2008.

[33] W. Ling, Y. Xu,M. Yunfei, F. Minrui, Discrete harmony search algorithm. Shanghai Key. Laboratory of Power Station Automation Technology, School of Mechatronics and Automation, University of Shanghai,2007.

[34] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation, vol. 12, pp. 702-713, (Decembre 2008), 2008.