

Analysis and Mathematical Justification of a Fitness Function used in an Intrusion Detection System

Pedro A. Diaz-Gomez *
Ingenieria de Sistemas Universidad El Bosque
Bogota, Colombia
pdiazg@ou.edu

Dean F. Hougen
Robotics, Evolution, Adaptation, and Learning
Laboratory (REAL Lab)
School of Computer Science, University of
Oklahoma Norman, OK, USA
hougen@ou.edu

ABSTRACT

One of the principal issues in using Genetic Algorithms is choosing the fitness function. Much of the success of the solution can be attributed to the fitness function in the sense that convergence to correct solutions depends in large part on the fitness function. A fitness function that captures all goals and constraints can certainly be used. However, setting parameters that join all goals and constraints appropriately can be difficult. This paper gives a mathematical justification for a fitness function that has previously been demonstrated experimentally to be effective.

1. INTRODUCTION

One of the most important objectives of every organization is to preserve the integrity, confidentiality, and availability of its data and information. Many efforts have been made to accomplish that goal: security policies, firewalls, Intrusion Detection Systems, anti-virus software, and standards to configure services in operating systems and networks [1]. This paper focuses on one of those topics: Intrusion Detection Systems, using a case-study of an off-line intrusion detection system known as *GASSATA* [6] that uses a Genetic Algorithm to search for matches in the audit trail. It has the unfortunate quality that the parameters for the fitness function cannot be tuned to effectively detect all possible attacks found in an audit trail while still avoiding false positives (warnings of attacks that do not exist) and false negatives (failing to detect real intrusions). We suggest a new fitness function independent of parameters to find intrusions in audit trail files and justify it mathematically.

2. GASSATA & INTRUSION DETECTION

*Conducting research at the Robotics, Evolution, Adaptation, and Learning Laboratory (REAL Lab). School of Computer Science, University of Oklahoma.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

GASSATA [6] is a tool to increase security audit trail analysis efficiency, that performs misuse detection by comparing the user's behavior (*OV* Vector) against a matrix of known attacks (*AE* Matrix). *GASSATA* explains the data contained in the audit trail by hypothesizing the occurrence of one or more attacks (*I* Vector), and it uses a heuristic method—GAs—to solve it because explaining the data is an NP-Complete problem. The fitness function suggested in *GASSATA* is:

$$F(I) = \alpha + \sum_{i=1}^{N_a} W_i * I_i - \beta * T^2. \quad (1)$$

where α is a parameter used to maintain $F(I) > 0$, N_a is the number of known attack types, W is a one-dimensional array of length N_a called the *weighted vector* that gives the risk of each attack, I is a one dimensional vector of length N_a called the *hypothesis vector*, $I_i = 1$ if attack i is present and $I_i = 0$ otherwise, and β is a parameter used to slope the penalty function T^2 .

To explain the data contained in the audit trail by the occurrence of one or more attacks, *GASSATA* attempts to find the I vector that maximizes the $W \cdot I$ product. In order to evaluate the constraint, the algorithm counts the number of events of each type generated by all the attacks hypothesized in I . If these numbers are less than or equal to the number of events recorded in the audit trail, then the hypothesis is realistic. On the contrary, if some of those numbers are greater than the number of events that occurred, then the solution proposed is penalized. i.e., if $(AE \cdot I)_i > OV_i$, a fails added [6].

Good results with *GASSATA* have been reported [6] although our use of the method has been problematic, as explained in [2].

We set various values to parameters α , W and β , however we obtain always a great number of false positives and some false negatives. As an example we use the fitness function $F(I) = 4.0 + \sum_{i=1}^{N_a} I_i - (1/20) * T^2$, we obtained 217% of false positives, and 20% of false negatives—100% corresponds to 4 actual intrusions. Continuous experimental results guide us to propose a new fitness function to solve the problem.

3. NEW FITNESS FUNCTION PROPOSED

As stated in [3], the term $\sum_{i=1}^{N_a} I_i$ was incorrectly guiding the fitness function, and the term T^2 was quite high counting the fails. So the solution proposed cuts the positive

side $\sum_{i=1}^{N_a} I_i$, and it uses as a penalty function T^1 , having into account that if two intrusions hit the same event with an overestimate of the number of events hypothesized then count them twice, and so forth. Call this T' .

With this in mind and the experience gained with testing, the fitness function proposed only has the penalty function. As the number of events is N_e , the new fitness function suggested is $F(I) = N_e - T'$. It must be stated that this fitness function was found after much testing and that is not the goal of this paper to show all the steps followed; with the fitness function proposed *there are no false positives* and the number of *false negatives decreases dramatically*. This time 70 runs were performed with different data (some data was downloaded from the Lincoln Laboratory [4]) and only one time a false negative was present.

While the fitness function suggested here certainly appears to function well, given the empirical data we have obtained, the question remains as to whether this fitness function can be justified mathematically. The following section shows that it can be.

3.1 Mathematical Justification

The problem is to find the vector I that maximizes the inner product

$$F(I) = W \cdot I \quad (2)$$

subject to

$$(AE \cdot I)_i \leq OV_i \quad (3)$$

and $I_i \in \{0, 1\}$ for $0 \leq i \leq N_a$

We can abbreviate the constraints given in Equation 3 using polynomials c_j in I and $a_{jk} \in AE$, following the notation suggested by Ham [5] as

$$c_j(I) = a_{j1}I_1 + a_{j2}I_2 + \dots + a_{jn}I_n - OV_j \quad \text{for } 0 \leq j \leq N_e \quad (4)$$

and we can join the objective function in Equation 2 with constraint in Equation 3 to obtain the *energy function* [5] defined as

$$E(I, K) = W \cdot I + K \sum_{j=1}^{N_e} \Phi[c_j(I)] \quad (5)$$

where the positive parameter K controls how well the unconstrained optimization problem in 5 to 7 approximates the original *Linear Problem* in 2 to 3

$$\Phi[c_j(I)] = \begin{cases} = 0 & \text{if } c_j(I) \leq 0 \\ > 0 & \text{if } c_j(I) > 0 \end{cases} \quad (6)$$

$$I_i \in \{0, 1\} \quad \text{for } 0 \leq i \leq N_a \quad (7)$$

and as we want to maximize E , $\Phi(t)$ must be differentiable with the property in Equation 6.

For simplicity, the function $\Phi(t)$ commonly selected is [5]

$$\Phi(t) = \begin{cases} = 0 & \text{for } t \leq 0 \\ = \frac{1}{2} t^2 & \text{for } t > 0 \end{cases} \quad (8)$$

and now finding the partial derivative of the E in Equation 5 with respect to I we obtain

$$\frac{\partial E(I, K)}{\partial I} = W + K \sum_{j=1}^{N_e} \Psi[c_j(I)] \frac{\partial}{\partial I} [c_j(I)] \quad (9)$$

where $\Psi(v) = \frac{d\Phi(v)}{dv} = \Phi'(v) = v$.

Using Equation 4 to find $\frac{\partial}{\partial I} [c_j(I)]$, Equation 9 gives [5]

$$\frac{\partial E(I, K)}{\partial I} = W + K \sum_{j=1}^{N_e} \Psi[c_j(I)] [a_{j1} \ a_{j2} \ \dots \ a_{jn}]^T \quad (10)$$

and as $\Psi[c_j(I)] = c_j(I)$, substituting in 10 and equating to 0—we are finding a maximum—we obtain

$$\frac{\partial E(I, K)}{\partial I} = W + K \sum_{j=1}^{N_e} c_j(I) * [a_{j1} \ a_{j2} \ \dots \ a_{jn}]^T = 0 \quad (11)$$

then using Equation 4 again we obtain

$$\begin{aligned} & \sum_{j=1}^{N_e} (a_{j1}I_1 + a_{j2}I_2 + \dots + a_{jn}I_n) * [a_{j1} \ a_{j2} \ \dots \ a_{jn}]^T \\ &= -\frac{W}{K} + \sum_{j=1}^{N_e} OV_j * [a_{j1} \ a_{j2} \ \dots \ a_{jn}]^T \end{aligned} \quad (12)$$

taking vector components in Equation 12 we get for $1 \leq i \leq n = N_a$

$$\begin{aligned} & \sum_{j=1}^{N_e} (a_{j1}I_1 + a_{j2}I_2 + \dots + a_{jn}I_n) * a_{ji} \\ &= -\frac{W_i}{K} + \sum_{j=1}^{N_e} OV_j * a_{ji} \end{aligned} \quad (13)$$

that can be written, having in mind that a_{ji} corresponds to AE_{ji} , as

$$\sum_{j=1}^{N_e} (AE \cdot I)_j * a_{ji} = -\frac{W_i}{K} + \sum_{j=1}^{N_e} OV_j * a_{ji} \quad (14)$$

As W , the weighted vector, is such that $\forall i \ W_i \geq 0$ and K is a parameter that approaches positive infinity [5] then Equation 15 must be satisfied by a maximum I of Equation 5

$$(AE \cdot I) \cdot \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{N_e i} \end{bmatrix} \leq OV \cdot \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{N_e i} \end{bmatrix} \quad (15)$$

And what we actually did, with the fitness function proposed, was to find I such that

$$(AE \cdot I)_j \leq OV_j, \quad \text{for } 1 \leq j \leq N_a \quad (16)$$

that clearly satisfies Equation 15, because

$$\begin{aligned} & a_{ji} \geq 0, \text{ for } 1 \leq j \leq N_e, 1 \leq i \leq N_a, \text{ with } a_{ji} \in AE; \\ & I_i \in \{0, 1\}, \text{ for } 1 \leq i \leq N_a, \text{ with } I_i \in I; \text{ and} \\ & OV_j \geq 0, \text{ for } 1 \leq j \leq N_e, \text{ with } OV_j \in OV. \end{aligned}$$

Thus, the fitness function we have proposed is justified mathematically.

4. CONCLUSIONS & FUTURE WORK

This paper shows some difficulties in providing accurate values to parameters in the fitness function suggested in GASSATA [6] and proposes a solution independent of variable parameters, making the fitness function to solve this problem quite general and independent of the audit trail data.

This paper used, in part, methodology used in the *neural networks* field [5] for *linear programming with inequality constraints* to justify experimental results with a new fitness function proposed, in order to solve the problem of analyzing audit data for intrusions as suggested in *GASSATA* [6].

5. REFERENCES

- [1] R. G. Bace. *Intrusion Detection*. MacMillan Technical Publishing, USA, 2000.
- [2] P. Diaz-Gomez and D. Hougen. Improved off-line intrusion detection using a genetic algorithm. To appear in *Proceedings of the Seventh International Conference on Enterprise Information Systems*, 2005.
- [3] P. A. Diaz-Gomez and D. F. Hougen. Analysis of an Off-line Intrusion Detection System: A Case Study in Multi-Objective Genetic Algorithms. To appear in *the Florida Artificial Intelligence Research Society Conference.*, 2005.
- [4] D. Fried and M. Zissman. Intrusion detection evaluation. Technical report, Lincoln Laboratory, MIT, 1998. <http://www.ll.mit.edu/IST/ideval/>, accessed March 2004.
- [5] F. M. Ham and I. Kostanic. *Principles of Neurocomputing for Science & Engineering*. Mc Graw Hill, 2001.
- [6] L. Mé. GASSATA, a genetic algorithm as an alternative tool for security audit trail analysis. In *First International Workshop on the Recent Advances in Intrusion Detection*, Belgium, 1998.