An Evolutionary Algorithm to Generate Ellipsoid Network Intrusion Detectors

Joseph M. Shapiro, Gary B. Lamont, Gilbert L. Peterson
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
WPAFB, Dayton, Ohio 45433
{joseph.shapiro,gary.lamont,gilbert.peterson}@afit.edu

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence - problem solving, control methods, and search

General Terms

Design, Algorithms

Keywords

Evolutionary computation, artificial immune systems, computational geometry, negative selection

1. INTRODUCTION

This paper introduces the ellipsoid as a geometric structure for detecting network intrusions. Section 2 describes and analyzes the design of the ellipsoid generation algorithm. Experimental design is set forth in Section 3. In Section 4 we analyze experimental results. Section 5 summarizes the paper and provides direction for continued research.

2. DESIGN

This section describes a mathematical ellipsoid model and an algorithm that evolves a set of ellipsoids to cover network intrusion space.

2.1 Ellipsoid Model

An n-d ellipsoid is defined as follows:

$$(\mathbf{x}-\omega)^T\mathbf{A}(\mathbf{x}-\omega) = 1 \tag{1}$$

where $\mathbf{A}$ is a real symmetric positive-definite $n \times n$ matrix and ω, an $n \times 1$ matrix, is the center of the ellipsoid. Any vector $\mathbf{x}$ that satisfies Equation 1 is on the surface of the ellipse.

Volume of Ellipsoid

The volume of an ellipsoid is

$$V = \Omega_n \ell_1 \ell_2 \cdots \ell_n \tag{2}$$

where Ω_n is the volume of an n-d hyper-sphere and $\ell_1, \ell_2, \ldots, \ell_n$ are the lengths of the n semiaxes of the ellipsoid. A can be rewritten so that the equation for an ellipsoid is

$$(\mathbf{x}-\omega)^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T(\mathbf{x}-\omega) = 1 \tag{3}$$

The diagonal entries in $\mathbf{\Lambda}$ are the inverses of the squares of the lengths of the semiaxes of the ellipsoid defined by Equation 1.

Membership of a Point in an Ellipsoid

Kelly et. al. [2] report that the Mahalanobis distance (left side of Equation 4) can be used to determine whether or not $\mathbf{p}$ lies inside of e. $\mathbf{p}$ is inside of e if and only if the inequality in Equation 4 holds.

$$(\mathbf{p}-\omega)^T\mathbf{A}(\mathbf{p}-\omega) < 1 \tag{4}$$

2.2 Evolving a Set of Ellipsoids

Producing a set of ellipsoids that maximizes coverage of intrusion space while minimizing coverage of self space is not a trivial problem. For this reason, we use an evolutionary algorithm (EA) to "evolve" good sets of ellipsoids. This section addresses the mapping of the ellipsoid model into representation, crossover, mutation, and objective function in the evolutionary algorithm domain.

Representation

The objective is to obtain an optimal set of ellipsoids. This implies that each individual should be a set of ellipsoids. However, to avoid computational complexity, we let each individual be one ellipsoid and evolve one set of ellipsoids.

Crossover With Ellipsoids

Crossover is not used because of our choice for representation. Since an individual is not an entire solution, there is no justification for trading "building blocks."

Mutating an Ellipsoid

Conceptually, there are three independent ways to mutate an ellipsoid: semiaxis orientations ($\mathbf{V}$), center (ω), and semiaxis lengths ($\mathbf{\Lambda}$).

GECCO'05, June 25–29, 2005, Washington, DC, USA.

The EA accomplishes orientation mutation by rotating the ellipsoid in a 2d plane. To accomplish this rotation, a small angle θ is chosen from a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = \frac{\pi}{2}$ radians. The vectors that represent the randomly chosen semiaxes to produce new semiaxes.

The EA center mutation operator mutates each of the n components of ω individually. The center mutation operator chooses each new center component from a Gaussian distribution with mean $\mu = \omega_i$. The standard deviation for the Gaussian distribution is a parameter that can be changed.

The third type of mutation, semiaxis length, results when $\mathbf{\Lambda}$ is manipulated. The EA semiaxis length mutation operator mutates each of the n semiaxis lengths individually. The semiaxis length mutation operator chooses the new length from a Gaussian distribution whose mean is the old length. The standard deviation is a parameter value that can be set to reflect the desired variability of the mutation.

2.3 Objective Function

The objective function is divided into a reward function and a penalty function. The reward function uses a structure called a 2^n-way tree [3, p.336-7] to approximate the area covered an ellipsoid and not covered by a larger ellipsoid in the population.

The penalty function discourages ellipsoids from covering self points. If an ellipsoid e covers β self points, its penalty function is

$$PENALTY(e) = 1.00 - (REWARD(E, e)/(2^{\beta} + 1)) \quad (5)$$

Evaluation of Equation 5 requires β, the number of self points that e covers. β is obtained by traversing the same 2^n-way tree used in the reward function (see [4].

The objective function is the result of the penalty function subtracted from the reward function.

3. EXPERIMENTAL DESIGN

We test our algorithm against pedagogical problems to validate the model. Then, the MIT DARPA ID data is used for real world testing.

3.1 Pedagogical Data Sets

Our pedagogical data sets provide a proof of concept by validating that an algorithm produces expected results on problems with known characteristics. Such pedagogical problems also afford an opportunity for visualization techniques because they can be smaller and lower dimension.

Three artificial data sets are referred to as Val1, Val2 and Val3. Figures 1 and 3 present Val1 and Val3, two self data sets for which the optimal solution is two ellipsoids. Val3 tests whether the algorithm can find an optimal solution when overlapping ellipsoids are required. Val2, presented in Figure 2, is an inverse problem. It tests how well the algorithm can find a set of ellipsoids to fill in a space that is not elliptically shaped. Although the optimal solution is not known for Val2, a visual inspection of the results and analysis of test data provide a good approximation as to how well the algorithm performs.

Test data are generated in the inverse of the self area for Val1-Val3. Part (b) of Figures 1 - 3 shows the test data.

3.2 Network Data

We also test against data from the 1999 DARPA IDS Evaluation Data Set [1]. For training, we use the week one data, which contains only normal traffic. For testing, we use week two data, which consists of normal traffic mixed with attacks. The data has three features: number of bytes per second, number of packets per second, and number of Internet Control Management Protocol (ICMP) packets per second.

4. RESULTS AND ANALYSIS

Subfigure (c) in Figures 1-3 shows the results of running the ellipsoid algorithm against the corresponding self data sets. From a visual perspective, the algorithm is successful. It finds the known solutions, covers non-elliptically shaped nonself space well, and even finds the optimal solution when it requires overlapping. When tested against the nonself test data shown in subfigure (b) of Figures 1-3, the algorithm also performs successfully. The ellipsoid algorithm covers all of the nonself test points in Figure 1 and about 95% of the nonself test points in Figures 2 and 3. These results are impressive, especially in Figure 2, since the ellipsoids must cover and area in the shape of inverted ellipsoids.

Our algorithm also performs well against the MIT intrusion detection data, achieving ∼91% true positive with ∼ 0% false alarm (see Table 1). Although this proves nothing about performance against other intrusion detection data, it provides good reason to continue research in the current direction.

5. CONCLUSION

Our current testing shows that our algorithm can successfully model spaces around a set of training points. Testing against the MIT intrusion detection data results in success, although further testing is necessary for more concrete validation.

6. REFERENCES

[1] Lincoln Laboratory at Massachusetts Institute of Technology, http://www.ll.mit.edu/IST/ideval/data/. *Lincoln Laboratory: DARPA Intrusion Detection Evaluation.*

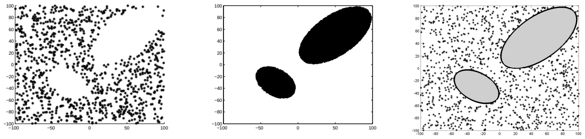
[2] Don R. Hush Patrick M. Kelly and James M. White. An adaptive algorithm for modifying hyperellipsoidal decision surfaces. *Journal of Artificial Neural Networks*, 1:49–480, 1994.

[3] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction.* Texts and Monographs in Computer Science. Springer-Verlag, 1985.

[4] Joseph M. Shapiro. An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. Master's thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, 2005.

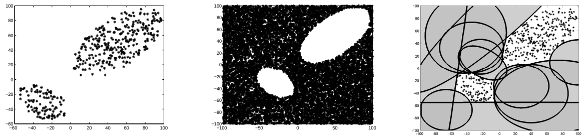
Training Data	Alorithm	Detection Rate		False Alarm Rate		Detectors
		Mean	SD	Mean	SD	
Pedagogical 1	**Spheres**	94.27	0.03	0	0	34
	Ellipses	94.05	0.03	0	0	**2**
Pedagogical 2	**Spheres**	95.56	0.01	0	0	20
	Ellipses	95.48	0.01	0	0	**12**
Pedagogical 3	**Spheres**	96.52	0.01	0	0	22
	Ellipses	96.82	0.02	0	0	**12**
MIT Data	**Spheres**	91.52	0.00	0.00	0.00	10
	Ellipses	91.64	0.00	0.00	0.00	**10**

Table 1: Comparison of spherical and elliptical detectors.



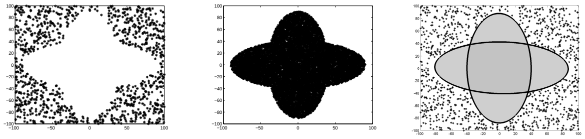
(a) self (b) nonself (c) results

Figure 1: Data set Val1. (a) is a data set with with two elliptical holes. The ellipses are oriented differently and are different sizes. (b) is its associated test data set. (c) is the ellipsoids found by the ellipsoid algorithm.



(a) self (b) nonself (c) results

Figure 2: Data set Val2. (a) has points inside of two ellipses. (b) is its associated test data set. (c) is the ellipsoids found by the ellipsoid algorithm.



(a) self (b) nonself (c) results

Figure 3: Data set Val3. In (a), the optimal solution is obviously two ellipses in a cross formation. (b) is its associated test data set. (c) is the ellipsoids found by the ellipsoid algorithm.