Under consideration for publication in Knowledge and Information Systems

The Importance of Generalizability for Anomaly Detection

Gilbert L. Peterson[1] and Brent T. McBride[1]

[1]Department of Electrical and Computer Engineering,
Air Force Institute of Technology, WPAFB OH, USA

Abstract.
In security-related areas there is concern over novel "zero-day" attacks that penetrate system defenses and wreak havoc. The best methods for countering these threats are recognizing "non-self" as in an Artificial Immune System or recognizing "self" through clustering. For either case, the concern remains that something that appears similar to self could be missed. Given this situation one could incorrectly assume that a preference for a tighter fit to self over generalizability is important for false positive reduction in this type of learning problem. This article confirms that in anomaly detection as in other forms of classification that a tight fit, although important, does not supersede model generality. This is shown using three systems each with a different geometric bias in the decision space. The first two use spherical and ellipsoid clusters with a k-means algorithm modified to work on the one-class/blind classification problem. The third is based on wrapping the self points with a multidimensional convex hull (polytope) algorithm capable of learning disjunctive concepts via a thresholding constant. All three of these algorithms are tested using the Voting dataset from the UCI Machine Learning Repository, the MIT Lincoln Labs intrusion detection dataset, and the lossy-compressed steganalysis domain.

Keywords: clustering; anomaly detection; convex polytope; ellipsoid;

1. Introduction

Many popular data classification methods are not blind, indicating that for decisions with two or more classifications the training set must consist of instances of each classification. If they are tested against an unfamiliar class instance, the learned hypothesis is unable to reliably distinguish the foreign instance from the

Received Oct 3, 2005
Revised Dec 11, 2006
Accepted Feb 4, 2007

classes of the training set. A blind classification method, often handled through clustering, recognizes that a foreign instance is not a member of any of its training classes and identifies it as an anomaly given a learned model from a single class' data. This kind of anomaly detection is useful when there is incomplete domain knowledge available for training, or when we hope to block anomalies which have never been seen previously.

In order to detect attacks from an attacker trying to blend in with normal network traffic, we compare the benefits of the casting of the search problem as a generalization of the normal data and whether generalization reduces anomaly detection accuracy and if there should be a preference toward fitting the normal "self" data more closely. Where generalization, as defined by Mitchell (Mitchell, 1982), is the process that takes a large number of samples and creates a hypothesis (inductive bias) that retains the important features of each class. Figure 1 shows the results of applying the modified k-means sphere, ellipse, and the convex polytope algorithms to each class separately for a simple two class problem. As can be seen from this example, the generalizability of the model decreases as the model improves its tightness to the data points, apparent by the amount of attribute space each shape covers. At the same time as the model fits self tighter, there is less overlap with the other classes and fewer false positives. Given a domain in which the attackers attempt to craft an attack that appears as close to normal (self) as possible, a blind learning approach which fits the model closely could be seen as important. Although a tight fit is important for anomaly detection the reduction in generality results in an adverse effect in which the percentage of false alarms increase.

The empirical evaluation of generalization has been investigated for function approximation (Wah, 1999), explanation-based learning (Cohen, 1988; Mitchell, et al., 1986), and classification (Barron, 1991; Baum and Haussler, 1988), but has previously not been explored for anomaly detection. This paper presents an empirical comparison of three geometric constructs, spherical, elliptical, and hyper-convex polytope representations, each with decreasing bias and generalizability for anomaly detection on several problems demonstrating that some generality is required for best performance. Results show that the elliptical bias performs best due to its capability of accurately estimating a convex polytope (Melnik, 2002) while retaining the best performance due to its simpler bias. These results are important because only by learning the best model of normal are we going to be able to detect and prevent previously unseen security attacks.

2. Related Work

The application of anomaly detection as a classification technique has become widespread as the number of application areas increases. Anomaly detection has been most valuable in security domains such as Intrusion Detection Systems (IDS) (Dasgupta and Gonzales, 2002; Denning, 1987; Eskin, 2000; Eskin, 2002; Lazarevic, et al., 2003; Fan, et al., 2004; Peterson, et al., 2005), detecting spam e-mail (Gupta and Sekar, 2003; Delany and Cunningham, 2006), virus detection in the unix process list (Inoue and Forrest, 2002; Lane and Brodley, 2003), and for detecting novel steganography in jpeg images (McBride, et al., 2005). Beyond anomaly detection's application in security domains, it has also been applied to the domains of hyperspectral imagery (Chang and Chi-

ang, 2002), and prognostics and health management of embedded hardware systems (Brotherton and Johnson, 2001).

For each application domain, the number of learning algorithms used is just as extensive. Two of the most popular algorithms are the single class support vector machine in which a kernel function is used to separate the normal samples from the spatial origin (Farid and Lyu, 2003; Lazarevic, et al., 2003; Eskin, 2002), and k-means (Eskin, 2002; Peterson, et al., 2005; McBride and Peterson, 2004; McBride, et al., 2005), or mixture models (Eskin, 2002), which make use of a geometric representation or distribution to classify normal around the model means. Other learning algorithms applied to an anomaly detection problem have consisted of self organizing maps (Brotherton and Johnson, 2001), k-Nearest Neighbor (Lazarevic, et al., 2003), Artificial Immune Systems (Dasgupta and Gonzales, 2002; Inoue and Forrest, 2002), and Hidden Markov Models (Cho and Park, 2003; Lane and Brodley, 2003).

Common to all of the different domains and application areas are some fundamental research issues. Similar to other machine learning problems, one of the fundamental research issues concerns the data set. The data set must consists of a representative sampling from the decision, and each item must be represented by an applicable set of features in order to learn a good model for classification (Duda, et al., 2001).

In anomaly detection, collecting a representative sampling is exacerbated by two very difficult problems that must be addressed. The first of these is the often used assumption that for training, the normal data is clean and contains no anomalies (McBride, et al., 2005; Farid and Lyu, 2003; Dasgupta and Gonzales, 2002). This is an assumption that for real world domains, such as intrusion detection systems may not be achievable, and instead requires that the anomaly detection system attempt to statistically separate the anomalies from noise in the normal network traffic (Eskin, 2000; Eskin, 2002).

The second sampling issue is that there is a large skew between the amount of normal and abnormal data samples in most data sets. For example, in the week 2 Lincoln Labs IDS data set, only 1.06% of the samples are anomalous (Kubler, 2006). The result of this imbalance is that often algorithms will either not identify the anomalies because the overall accuracy of classifying all data as clean is often higher than systems which have even a small percentage of false positives mixed with missed detections. Because of this, in addition to trying to increase anomaly detection algorithm accuracy, much of the anomaly detection research focuses on finding a balance between reducing the number of false positives while increasing the number of detections(Dasgupta and Gonzales, 2002; Denning, 1987; Eskin, 2002; Lazarevic, et al., 2003; Peterson, et al., 2005). Another effect of the data skew concerns balancing the costs associated with incorrect classifications (Drummond and Holte, 2005). For example, does falsely labelling a normal object as an anomaly have the same operational costs as missing a true anomaly.

The second data set issue is that of determining a representative set of features. Many anomaly detection systems are faced with an abundance of possible attributes and make use of statistical features in order to reduce the scale of the data that must be dealt with (Chang and Chiang, 2002; Dasgupta and Gonzales, 2002; Farid and Lyu, 2003; Jackson, 2003; Peterson, et al., 2005). As a result of the inability of the algorithms to scale to ever larger datasets, or draw inferences from the data on their own, often the feature development becomes more

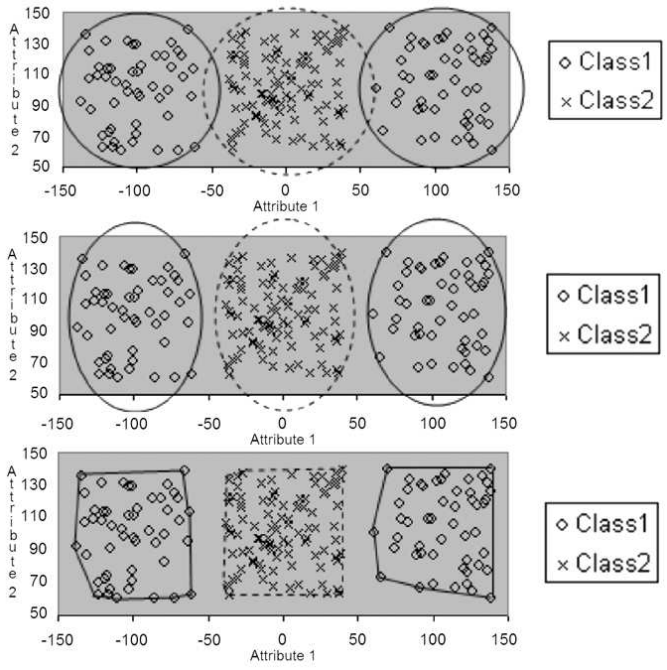

Fig. 1. A Simple 2-Class Problem with Sphere, Ellipse and Convex Polytope.

prominent that the learning algorithm, as techniques to detect specific anomalies are created (Farid and Lyu, 2003; Fridrich, et al., 2001; Avcibas, et al., 2002).

Another anomaly detection research issue is that of handling dynamic environments, whether it be represented as concept drift (Widmer and Kubat, 1996), or lifelong learning (Thrun, 1995). For example, if an anomaly detection algorithm were to function as a biometric security system based on a users typing rhythm. And the user were to come back a day later having injured their hand and disrupted their own typing rhythm is this an anomaly or is this just a change in the rhythm that the anomaly detection system must track. Because the system must separate the noise from the actual concept drift, this is most often handled through some form of feedback (Delany and Cunningham, 2006).

3. The Blind Classifiers

This section discusses the geometric biases used in each of the blind classifiers. The three geometric biases are convex polytopes, hyper-spheres, and hyper-ellipsoids.

3.1. Convex Polytope

Central to the first geometric classifier algorithm is the concept of a polytope. A *d-polytope* is a closed geometric construct bounded by the intersection of a

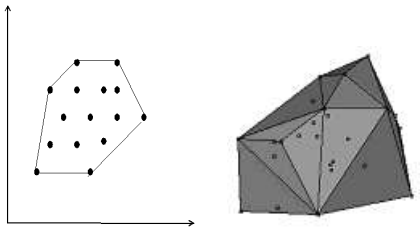

Fig. 2. Convex Hulls: (a) 2-D, and (b) 3-D (Lambert, 1998)

finite set of hyperplanes, or halfspaces, in d dimensions (Coxeter, 1973). As the number of dimensions rises, the polytope structure becomes increasingly complex and unintuitive.

A polytope is *convex* if a line segment between any two points on its boundary lies either within the polytope or on its boundary. A *convex hull* of a set of points S in d dimensions is the smallest convex d-polytope that encloses S (O'Rourke, 1998). Each vertex of this enclosing polytope is a point in S (Figure 2). The qhull program (Barber and Huhdanpaa, 2002), version 2002.1, is used with this convex polytope classifier which has a worst time complexity of O($n^{d/2}$) for n input points in d-space (Barber, et al., 1997).

Using a convex polytope for clustering requires mapping the training instances for a particular class C to a set T of d-vectors. A test point p is declared to be a member of class C iff it is bounded by the polytope defined by computing the convex hull of T. This is determinable by computing the convex hull of T unioned with p if the new polytope is the same as the previous then p matches the model and is part of class C.

Additionally, the possibility that a class attribute space is disjunctive, rather than contiguous exists. To compensate for disjunctions and lessen the impact of statistical outliers, a tolerance feature controlled by parameter $0 \leq \beta \leq 1$ is added. The samples are partitioned into unconnected sets where the distance squared between the two closest samples of each set are greater than $\beta^2 d(MAX_i - MIN_i)^2$ where MAX and MIN are the largest and smallest values for each attribute dimension (i).

One mechanism for guiding the selection of β determines the finite number of β values which produce unique partitionings of the data. This method works by sorting the upper-triangular distance-squared matrix for all instances of the training class. Each of these squared distances are then mapped to distinct β^2 values. This set of values, B, then represents the significant β values as only they may yield distinct polytopes (McBride, et al., 2005).

The convex polytope provides the least generalization and the tightest fit around training data of the three algorithms. However, its exponential-in-d time complexity limits its feasibility to classification problems containing a relatively small number of attributes.

3.2. k-means with Hyper-Spheres

The k-means algorithm assigns points to clusters by attempting to minimize the sum of squared within group errors (MacQueen, 1967). The algorithm performs

iterations re-assigning points to different clusters and adjusting the centroids until it can no longer reduce the sum of squared within group errors by further shuffling. Selection of the number of means k can be done via the Bayesian Information Criterion (x-means) (Pelleg and Moore, 2000), Gaussian means (G-means) (Hamerly and Elkan, 2003), or experimentation, as is done here in the interest of achieving the best results. The time complexity of the k-means algorithm is O(knr) for k clusters, n points, and r iterations (Wong, et al., 2000).

The cluster *centroids* produced by the k-means algorithm are the center points of the k hyper-spheres in class model S. The radius of each hyper-sphere is given by the distance between the corresponding centroid and the most distant point in its cluster. Point p is bounded by a hyper-sphere with center point c and radius r iff $dist(p, c) \leq r$. A point is declared a member of class if it is enclosed by any of the k hyper-spheres in S.

Testing a point for inclusion in the k hyper-spheres of S takes O(kd) time. The obvious advantage the hyper-sphere model has over a convex polytope is that its time complexity is linear, not exponential, in d. However, because of the sphere's greater bias, the algorithm does not fit the normal samples as closely and has a greater chance for classifying false positives. Thus a third classifier is presented that attempts to strike a balance between these two paradigms and leverage their relative strengths (i.e., the tighter fit of a convex polytope and the computational feasibility of a hyper-sphere).

3.3. k-means with Hyper-Ellipsoids

A hyper-ellipsoid, as observed by Nguyen, et al. (2003), can be used to approximate a convex polytope. Hyper-ellipsoids have been used to classify high-dimensional data in previous work. Specifically, Melnik (2002) makes use of a special kind of ellipsoid, the Minimum Volume Ellipsoid (MVE), in which the size of the ellipsoid, s, is equal to the dimensionality of the space and the shape of the ellipsoid,Σ^{-1}, is a scatter matrix of points. This research differs from the MVE ellipsoid definition in that Σ^{-1} is instead an inverse covariance matrix of points, which relates to the scatter matrix via a calculation of the mean and covariances and for the number of samples in the datasets requires far less space. Additionally, our methodology differs in that instead of the ellipsoid representing the entire decision space, mutliple ellipses represent the decision space and better represent the training sample topology.

Like the hyper-sphere model, the hyper-ellipsoid model first separates the training set T of class C into k clusters using the k-means algorithm. Each cluster ellipsoid is defined by $(x - \mu)^T \Sigma^{-1} (x - \mu) = s$ where s specifies the ellipsoid size, μ specifies the center point as a vector in the attribute space, Σ the ellipse shape, and x is a d-vector representing a point on the border (locus) of the ellipsoid. At this stage, Σ^{-1} and μ are computed, but s is still an unknown quantity. The size of each cluster ellipsoid must be chosen carefully, as it affects the fit and generality of the resulting class model.

Define L as the sorted-ascending list of s values that results from computing the minimum s for each cluster point as x, where $s = L_{|L|}$ defines the smallest ellipsoid size that encloses all cluster points. If the cluster contains extreme points (statistical outliers), then using $L_{|L|}$ as the s value results in an ellipsoid that encloses too much of the attribute space and that has a high probability of

declaring false-positive matches. Therefore, a tolerance parameter, $0 \leq \delta \leq 1$, is applied to allow the user to tweak the size of the hyper-ellipsoid.

A preliminary cluster ellipsoid size is $s = L_{\delta|L|}$. Thus, if $\delta = 0.9$ then the upper-tenth percentile of cluster points (the 10% that create the largest s values) are not enclosed by the hyper-ellipsoid, which prevents the most extreme points from affecting the size of the hyper-ellipsoid model. To purge their influence from the ellipsoid shape and location parameters as well, Σ^{-1} and μ must be recomputed for the cluster subset containing only the bottom δ-percentile of points. Then L is recomputed for the new hyper-ellipsoid parameters and the remaining cluster points. Now that the effects of the discarded points are completely purged, the final cluster s value is set to $L_{|L|}$.

Once s values are selected for each cluster, a test point p is declared to be a member of class C iff $(p-\mu)^T\Sigma^{-1}(p-\mu) \leq s$ for any of the k ellipsoids of C. The time complexity of testing a point for inclusion in the k clusters of C takes $O(k[d^2+d]) \approx O(kd^2)$ time, while creating the k ellipsoid models has a time complexity of $O(kn^2d^2)$.

In order to get the best performance from the classifier, the values for k and δ are determined experimentally for each test domain. Where increases in k and decreases in δ coincide with a decrease in generality in the interest of increased probability of detection and vice versus for a decrease in probability of false alarms. It is possible that the approach could be automated to make use of G-means methodology (Pelleg and Moore, 2000) for determining k where a Gaussian mean for each dimension is determined based on the covariance matrix.

The flexibility of this classification paradigm allows for uses in many possible domains. However, as this research focuses mostly on evaluating anomaly classification. The next section describes the testing regimen used for evaluating these three techniques and demonstrating the importance of the tradeoff between a tight fit to normal with generality.

4. Testing Methodology and Results

The convex polytope, hyper-sphere, and hyper-ellipsoid are tested against the Voting dataset from the UCI Machine Learning Repository (Blake, et al., 1998) to evaluate their strengths and weaknesses. The classifiers are then tested on the MIT Lincoln Labs Intrusion Detection dataset and the lossy-compression steganalysis domain to show performance on realistic anomaly detection problems.

For each dataset, 90% of training class instances are randomly selected and are used to create the class model. Next, the model is tested against the remaining 10% of the class instances plus a randomly-selected 10% sampling of the other class(es). This random model creation and test process is repeated ten times for each class. The means and standard deviations for the Probability of Detection (P_D) of the anomalous class(es) and the Probability of False Alarms (P_F) on the normal class are collected. These statistics are of interest as they demonstrate both how well each technique identifies anomalies as well as the percentage of normal samples misclassified.

For all convex polytope tests, the β value is ranged from 0.1 to 1.0 in steps of 0.05. For the hyper-sphere and hyper-ellipsoid k-means variants, k is tested at 1 to 5 in steps of 1, and 5 to 100 in steps of 5, with δ set to 0.9, 0.95, and 1.0. Because of the use of δ to make the spherical and elliptical models fit normal as closely as possible the choice was made to not use a k prediction method (Pelleg

and Moore, 2000; Hamerly and Elkan, 2003). The interactions of these variables are shown with respect to the Voting database in the following subsection.

4.1. Voting Database

The Voting database contains the voting records of members of the 1984 U.S. House of Representatives on 16 key votes. Each instance in the database represents an individual member of Congress who belongs to one of two classes: Democrat or Republican. The database includes 267 Democrat and 168 Republican instances. The instance attributes are the choice of each Congress member's 16 votes. Each attribute has one of three values: "yea", "nay", and "unknown" arbitrarily mapped to 1, -1, and 0, respectively.

Blind normal models are created for each of the two classes (Democratic and Republican). Due to the dimensional complexity of the convex hull algorithm, the convex polytope classifier trains on only the first seven of the 16 attributes.

The most accurate β values, as determined by the best balance between the detection and false alarm probabilities, for the Republican blind model range roughly between 0.45 and 1.0, shown in Figure 3. The best β for the Democrat model is at about 0.3. At these β values both models exhibit good and stable classification accuracy with low incidence of false positive and false negative matching errors. The Republican model has a $P_D = 95.7\%$ and $P_F = 24.5\%$ at $\beta = 0.75$, versus the Democrat model's $P_D = 95.6\%$ and $P_F = 9.8\%$ at $\beta = 0.3$ (Table 1). It is also important to note that there are only a few β values that modify the clustering of the convex polytope appearing as plateaus in Figure 3.

The hyper-sphere models do not perform as well as the convex polytope. At every value of k the models exhibit inferior balancing of false positive and false negative errors, shown in Figures 4 and 5. The best accuracy for the blind Republican and Democrat models results in a $P_D = 75.2\%$ and $P_F = 22.8$ at $k = 15$ and $P_D = 87.8\%$ and $P_F = 13.2\%$ at $k = 70$, respectively. However, these values reflect that the hyper-sphere model is not sufficiently stable as the k value changes can cause dramatic change in the accuracy.

The average accuracy of the best performing ellipsoid models at each δ value are summarized in Table 1 and Figures 4 and 5. The best performing models for all δ values (highlighted in the table) occur at $k = 1$, which suggests that the attribute space of each class is not disjunctive and is well represented by a single convex shape. The Democrat class appears to have a number of statistical outliers that cause false-positive problems when included in the class model ($\delta = 1$). When 5% of the most extreme points are discarded ($\delta = 0.95$) performance increases dramatically from 66.3% to 90.8%. It seems there are a few Democrats whose voting records are more typical of Republicans. The Republican model, on the other hand, performs best when no points are discarded ($\delta = 1$), indicating greater consistency within the class. Overall as seen in Figures 4 and 5, the value of k has a large effect on the performance of the hyper-ellipse where the setting of δ reduces the false alarms and detection probabilities for smaller values.

Overall, for this dataset the hyper-ellipsoid model outperforms both the more general (hyper-sphere) and the more specific (convex polytope) classifiers. This underlines the importance of the balancing of the degree of generalization, and is also evident in results from the Iris and Diabetes UCIMLR datasets (**?**).

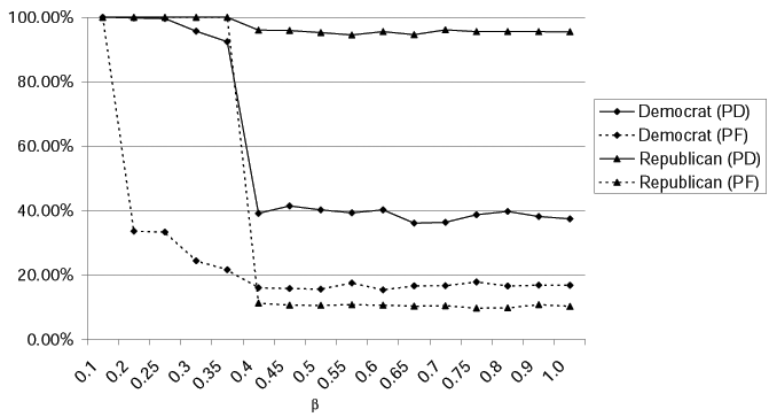

Fig. 3. Vote Results for Convex Polytope.

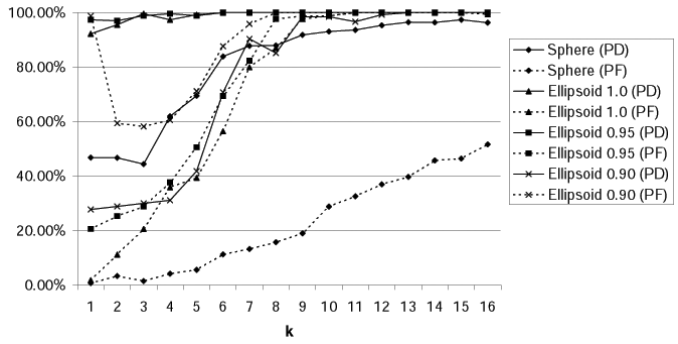

Fig. 4. Class Republican Vote Results for Hyper-Sphere and Hyper-Ellipse.

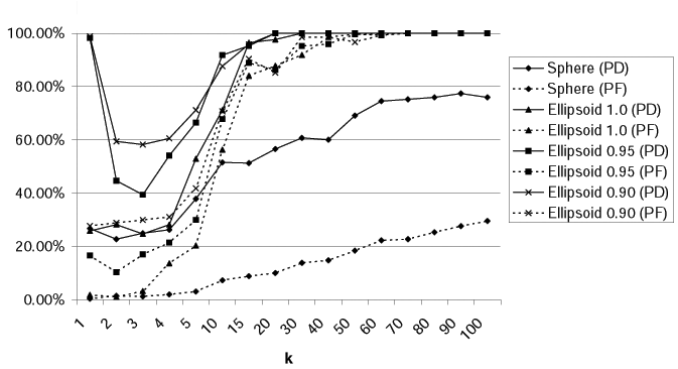

Fig. 5. Class Democrat Vote Results for Hyper-Sphere and Hyper-Ellipse.

Table 1. Voting Database: Best scores for each model type.

Class Modeled		Convex Polytope		Hyper-Sphere		Hyper-Ellipsoid		
		score	β	score	k	score	k	δ
Democrat	P_D	95.7 ± 5.4	0.30	75.2 ± 21.1	70	98.2 ± 5.9	1	0.95
	P_F	24.5 ± 9.1		22.8 ± 10.4		16.7 ± 3.4		
Republican	P_D	95.6 ± 3.6	0.75	87.8 ± 13.3	15	92.2 ± 3.9	1	1.00
	P_F	9.8 ± 6.1		13.2 ± 9.0		1.8 ± 2.7		

Table 2. Week 2 Attack Profile.

Day	Attack	Attack Type	Start Time	Duration
1	Back	DOS	9:39:16	00:59
2	Portsweep	Probe	8:44:17	26:56
3	SATAN	Probe	12:02:13	2:29
4	Portsweep	Probe	10:50:11	17:29
5	Neptune	DOS	11:20:15	4:00

4.2. IDS Experiment

The dataset used for this experiment was obtained from the Lincoln Laboratory of the Massachusetts Institute of Technology (Haines, et al., 1999). Although this data set has been shown to be statistically different from normal traffic (Mahoney and Chan, 2003), its many uses by the research community allow for comparison with other approaches. For this experiment, we used the 1999 data set, with week 1 (normal traffic) to train our classifiers, and week 2 (normal traffic mixed with attacks) for testing. Abnormal activity includes both internal (misuse) and external (hacking or denial of service) attacks, but not the external use of operating system or application exploits, as shown in Table 2.

We follow the same data preparation methodology as (Dasgupta and Gonzales, 2002) and collect statistics on the number of bytes per second, number of packets per second, and number of Internet Control Message Protocol (ICMP) packets per second for classification features. This results in 4800 normal data samples from week 1 for training, and 5202 data samples from week 2 for testing, of which 64 of these represent the attacks from Table 2. These features were sampled each minute from the raw tcpdump data files. Dasgupta and Gonzalez showed that while none of these features alone reliably detects the five attacks, combining the features was quite effective. They also explored overlapping the time series as a means of detecting temporal patterns, with their best results generated using a sliding window of three seconds. Detection and false alarm probabilities were calculated by comparing the classifier output with the Week 2 attack data. Table 3 shows the results of testing the k-means sphere and ellipse classifiers, the convex polytope, and the Artificial Immune System (AIS) results (Dasgupta and Gonzales, 2002). The table contains the best results found for Probability of False Alarm, and Probability of Detection, for each algorithm with the exception of the AIS which includes the results for 1 and 3 time slices from (Dasgupta and Gonzales, 2002).

As shown in Table 3, the ellipsoid model with its added capability of generalizing beyond the strict sampling better fits the training data over the convex polytope. In addition, the results show that the sphere version of k-means performs poorly predominantly because it inaccurately covers the training attribute space by also enclosing space including anomalous data points. This continues

Table 3. IDS Results.

	Convex Polytope		**Sphere**		**Ellipse**		**AIS**	
	$\beta > 0.3$	β=0.1	k=75 δ=1.0	k=100 δ=0.9	k=30 δ=1.0	k=75 δ=1.0	1 time slice	3 time slices
P_D (%)	98.2	100.0	1.82	5.45	98.2	100.0	92.8	98.0
P_F (%)	0.27	0.35	0.0	1.02	0.0	0.2	1.0	2.0

even as k increases and each cluster decreases in size. The reason the sphere does not perform as well as the other two geometric constructs is that the k-means classifier uses the point furthest from the mean to estimate the size of the hyper-sphere, resulting in an over-generalization. This contrasts with the ellipse and convex polytopes which maintain a closer fit to the training data. These results imply that the convex polytope and the hyper-ellipse k-means had little trouble fitting the training data, and that their ability to more tightly fit the self space improves their overall performance for classification based on these three statistical attributes. Additionally this shows that although both models fit the data closely that the added generality of the hyper-ellipse k-means reduces the false positives which is counter to the assumption that one would want the closest fit to the training data for anomaly detection.

4.3. Steganalysis Experiment

Steganography refers to hiding information in an innocuous place so that it may be transmitted without notice. With digital images, the message is hidden within a cover image. The steganograpy process varies the image's pixels in such a way that the changes are virtually undetectable to the human eye. The cover images that provide the most difficulty for message detection are JPEG images.

JPEG compression is a lossy image compression technique that exploits the fact that the eye cannot detect subtle changes in an image. In a JPEG image, a message is stored using the least significant bit (LSB) or by manipulating the rounding errors of the quantized discrete cosine transform (DCT) coefficients of each 8x8 iamge block.

For the lossy steganography problem there have only been a few applications of learning models for normal images, and none have used any type of clustering. Approaches which make use of both self and non-self data have used Fisher's linear discriminant, Support Vector Machines with image quality metrics, and wavelet statistics calculated from the suspect images (Farid and Lyu, 2003; Lyu and Farid, 2002; Lyu and Farid, 2004; Avcibas, et al., 2002). (Kharrazi, et al, 2005) provides a survey of the metrics available and their utility for steganalysis.

In steganalysis as in other security domains, difficulty arises when the classifier requires examples from the anomalous class in order to detect the anomaly, but may not have examples in the case of a novel embedding technique. In this case, anomaly detection provides the best means of detecting the novel embedding technique. and the blind or one-class learning methodologies applied to this leaning problem have consisted of Artificial Immune Systems (Jackson, 2003) and single class Support Vector Machines (Lyu and Farid, 2004).

For this domain we test using the wavelet coefficient statistics (Farid and Lyu, 2003) derived from a database of 1,100 grayscale images. The best three of the 36 coefficients determined by J-score are extracted from each image. In

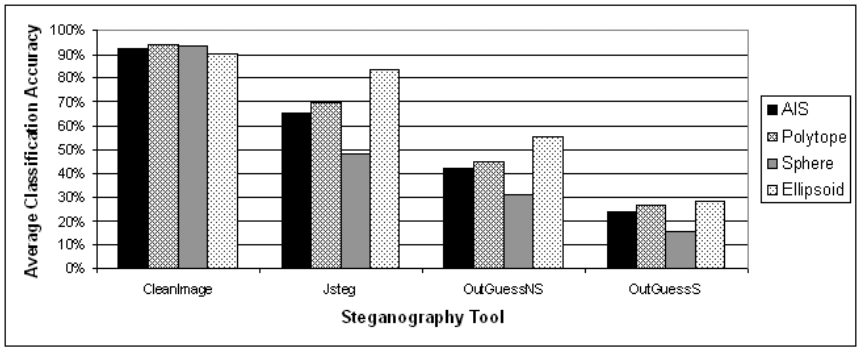

Fig. 6. Steganography Results.

addition to clean images, the testing set includes steganographic images created with Jsteg, and Outguess with (OutguessS) and without (OutguessNS) statistical correction. For each of these three steganography methods, images are created using 100%, 50%, 25%, and 12.5% of the cover image's embedding capacity. Ten iterations of training and testing are performed, where for each iteration, 18% of the clean image class is randomly selected for training and a random 9% of each class clean and dirty are used for testing. Testing is conducted on one embedding percentage at a time, and the results from the best performing parameter settings are averaged. Where the best performing parameters are for the convex polytope $\beta = 0.1$, for the hyper sphere $k = 40$, and for the hyper-ellipse $k = 1$ and $\delta = 0.85$.

Figure 6 shows the average detection percentage over the 4 embedding capacities from the steganography testing compared with the results using the same testing domain and an AIS from (Jackson, 2003). As seen with the IDS problem, the closer fit to self provided by both the convex polytope and ellipse k-means outperforms the more general sphere k-means. However, just as the results in the previous datasets show, striving for the closest fit possible, i.e. the convex polytope, creates a lack of generality, especially on the Jsteg dataset, that is detrimental to the convex polytope over the ellipse k-means.

4.4. Summary of Results

Of the test results shown, the steganalysis results are the most revealing because the information hiding community specifically strives to make the embedded cover image appear as normal as possible. Additionally, they have had a lot more practice at it than the network attacks seen in the IDS dataset. The outcome of the steganographer's experience results in an extremely difficult domain in which to perform anomaly detection.

Table 4 shows a summary of the results, listing for each domain, the number of classes and attributes in the domain as well as the probability of detection P_D and the probability of false alarms P_F for each class. The bolded values highlight the model which achieved the best overall accuracy. In the steganalysis domain as in the other datasets, the highest overall accuracy occurs with the hyper-ellipsoid. The reason for this is that while seeking to fit the normal space, the algorithm retains generality provided by the bias of it's geometric representation.

Table 4. Summary of Testing Results.

Database Info				Best Anomaly-Based Accuracy for Each Class Model %		
Name	**Classes**	**Attributes**		**Convex Polytope**	**Hyper-Sphere**	**Hyper-Ellipsoid**
Voting	2	16	P_D	95,95	75,88	**98,92**
			P_F	24,10	23,13	**17,2**
IDS	2	3	P_D	100.0	5.5	**100.0**
			P_F	0.4	1.0	**0.2**
Stego	2	3				
JSteg				69.3	48.1	**83.4**
OutguessNS				44.8	30.9	**55.5**
OutguessS				26.5	15.4	**28.1**
False Alarms(P_F)				5.9	6.6	9.4

The increase in generality tends to result in smaller false alarm probability, while the more complex models increase the detection probability. This aligns with the bias complexity of each of the geometric models which the bias decreases from sphere to polytope, the model fits the self space more closely with less generality. Which for anomaly detection against an adversary attempting to resemble normal behavior a close fit to self space could be considered advantageous. However, as is shown in Table 4, rarely does the most general (hyper-sphere) or most specific model (convex-polytope) outperform the other models. Because the hyper-ellipse is a good approximation of the convex polytope it provides the benefits of the approaching a tight fit of the space while maintaining the advantages of the more general model.

5. Conclusion

For security anomaly detection domains, a concern prior to fielding a detection system is whether it can be defeated by an attacker manipulating their attack to appear as normal traffic. From an anomaly detection problem view, we have compare the benefits of the casting of the search problem as a generalization of the normal data and whether generalization reduces the anomaly detection accuracy and if there should be a preference toward fitting the normal "self" data more closely. This has been tested on two security domains, intrusion detection and steganalysis, and additionally on the Voting, Iris, and Diabetes datasets. The results for all of these datasets demonstrate that for anomaly detection, generality is required to reduce the false alarm probability, but one must select a bias that fits self closely to improve the detection probability.

The three techniques demonstrated in this article each perform blind classification with different geometric biases in the decision space. This paper shows that while the more complex convex polytope provides the tightest fit to self, the hyper-ellipse provides the best balance between fit and generality, and that both outperform the simplest hyper-sphere model. The small amount of generality provided by the ellipse results in the hyper-ellipse k-means outperforming the other methods on 91% of the datasets.

The results have demonstrated that the elliptical bias performs best due to it's capability of accurately estimating a convex polytope (Melnik, 2002) while retaining generality due to it's simpler bias. This indicates that in learning models

of normal that the investigator must examine the learning technique being used; ensuring that the normal space closely fits normal and that the technique used does not have an overly complex bias, still providing generality in order to best detect and prevent previously unseen security attacks.

Acknowledgements. The work on this paper was supported (or partially supported) by the Digital Data Embedding Technologies group of the Air Force Research Laboratory, Information Directorate. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Air Force Research Laboratory, or the U.S. Government.

References

Avcibas, I., Memon, N., and Sankur, B. Image Steganalysis With Binary Similarity Measures. *International Conference on Image Processing*, Rochester, NY, September 2002.

Barber, C.B.,Dobkin, D.P., and Huhdanpaa, H.T. The Quickhull Algorithm For Convex Hulls. *ACM Trans. on Mathematical Software*. 22, 469-483, 1997.

Barber, C.B and Huhdanpaa, H.T. Qhull, Version 2002.1. 283k. Computer Software, 2002. http://www.thesa.com/software/qhull/.

Barron, A.R. Approximation and Estimation Bounds for Artificial Neural Networks. *Proc. of the Fourth Ann. Workshop on Computational Learning Theory*, Morgan Kaufmann, Palo Alto, Calif., 243-249, 1991.

Baum, E.B., and D. Haussler. What Size Net Gives Valid Generalization?. *Proceedings of Neural Information Processing Systems*. New York, NY, 81-90, 1988.

Blake, C.L. and Merz,C.J. *UCI Repository of Machine Learning Databases*. University of California, Department of Information and Computer Science. Irvine, CA, 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html

Brotherton, T., and Johnson, T. Anomaly detection for advanced military aircraft using neural networks. *IEEE Aerospace Conference* , Big Sky, MT, 2001.

Chang, C.I., and Chiang, S.S. Anomaly detection and classificiation for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*. 40(6):1314-1325. 2002.

Cho, S.B., and Park H.J., Efficient anomaly detection by modeling privilege flows using hidden Markov model. *Computers & Security*. 22(1):45-55, 2003.

Cohen, W.W. Generalizing Number and Learning from Multiple Examples in Explanation Based Learning. *Machine Learning*, 256-269, 1988.

Coxeter, H.S.M. *Regular Polytopes, 3rd ed.*. New York.: Dover, 1973.

Dasgupta, D., and Gonzales, F. An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Trans. on Evolutionary Computation*. Vol 6, June 2002.

Delany, S.J., and Cunningham, P. ECUE: A Spam Filter that Uses Machine Learning to Track Concept Drift. Technical Report TCD-CS-2006-05, Trinity College Dublin, Computer Science Department, 2006.

Denning, D.E. An intrusion detection model. *IEEE Transactions on Software Engineering*. SE-13:222-232, 1987.

Drummond, C., and Holte, R. Learning to Live with False Alarms. *KDD-2005 Workshop on Data Mining Methods for Anomaly Detection*. August 21-25, 2005, Chicago, IL, pp. 21-24.

Duda R,O., Hart, P.E., and Stork, D.G. *Pattern Classification, 2nd Edition*. John Wiley & Sons, Inc., 2001.

Eskin, E. Anomaly detection over noisy data using learned probability distributions. *Proceedings of the International Conference on Machine Learning*. Stanford University, 2000.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L, and Stolfo, S. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Data Mining for Security Applications*. Kluwer, 2002.

Fan, W., Miller, M., Stolfo, S., Lee, W., and Chan, P., Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, Springer, 6(5):507-527.

Faird, H. and Lyu, S. Higher-order Wavelet Statistics and their Application to Digital Forensics. *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI, June 2003.

Fridrich, J., Goljan, M., and Du, R. Detecting LSB Steganography in Color and Gray-Scale Images. *Magazine of IEEE Multimedia Special Issue on Security*. October 2001, pp. 22-28.

Gupta, A., and Sekar, R. An Approach for Detecting Self-Propagating Email Using Anomaly Detection. *Lecture Notes in Computer Science: Recent Advances in Intrusion Detection: 6Th International Symposium, RAID 2003*. Pittsburgh, PA, USA, September 8-10, 2003.

Haines, J., Lippmann, R. , Fried, D., Tran, E., Boswell, S., and Zissman, M. 1999 DARPA Intrusion Detection System Evaluation: Design and Procedures. MIT Lincoln Laboratory Technical Report.

Hammerly, G., and Elkan, C. Learning the k in k-means. *Advances in Neural Information Processing Systems 15 (NIPS)*, 2003.

Inous, H., and Forrest, S. Anomaly Intrusion Detection in Dynamic Execution Environments. *New Security Paradigms Workshops*. 2002.

Jackson, J. *Targeting Covert Messages: A Unique Approach For Detecting Novel Steganography*. Masters Thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, 2003.

Kharrazi, M., Sencar, T., and Memon, N. Benchmarking Steganographic And Steganalysis Techniques. *IEEE SPIE* San Jose, CA, January 16-20, 2005.

Kubler, T.L. *Ant Clustering with Locally Weighting Ant Perception and Diversified Memory*. Masters Thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, 2006.

Lambert, T. Convex Hull Algorithms applet, UNSW School of Computer Science and Engineering, 1998. http://www.cse.unsw.edu.au/ lambert/java/3d/hull.html

Lane, T., and Brodley, C. An Empirical Study of Two Approaches to Sequence Learning for Anomaly Detection. Machine Learning. 51(1):73-107, 2003.

Lazarevic, A., Ertoz, L., Ozgur, A., Srivastava, J., Kumar, V. Evaluation of Outlier Detection Schemes for Detecting Network Intrusions. *Proc. Third SIAM International Conference on Data Mining*. San Francisco, CA, May 2003.

Lyu, S., and Farid, H. Detecting Hidden Messages Using Higher-Order Statistics And Support Vector Machines. *Information Hiding: 5th International Workshop, IH 2002*, Noordwijkerhout, The Netherlands, October 7-9, 2002.

Lyu, S., and Farid, H. Steganalysis Using Color Wavelet Statistics And One-Class Support Vector Machines. *SPIE Symposium on Electronic Imaging*, San Jose, CA, 2004.

MacQueen, J. Some methods for classification and analysis of multivariate observations. *Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297, 1967.

Mahoney, M., and Chan, P., An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. *Proceedings of the Recent Advances in Intrusion Detection, RAID 2003*. Pittsburgh, PA, USA, September 8-10, 2003.

McBride, B., and Peterson, G., Blind Data Classification using Hyper-Dimensional Convex Polytopes. *Proc. of the 17th International FLAIRS Conference*, Miami, FL, 520-526, 2004.

McBride, B. T., Peterson, G. L., and Gustafson, S. C. A New Blind Method for Detecting Novel Steganography. *Digital Investigation*. 2:50-70, 2005.

Melnik, O. Decision Region Connectivity Analysis: A method for analyzing high-dimensional classifiers. *Machine Learning*. 48:(1/2/3), 2002.

Mitchell, T.M. Generalization As Search. *Artificial Intelligence*. vol. 18. 203-226, 1982.

Mitchell, T.M., R.M. Keller, and S.T. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*. vol. 1 no. 1, 47-80, 1986.

Nguyen, H., O. Melnik, and K. Nissim. Explaining High-Dimensional Data. unpublished presentation. http://dimax.rutgers.edu/ hnguyen/GOAL.ppt. Accessed 4 Aug 2003.

O'Rourke, K. *Computation Geometry in C, 2nd ed.* Cambridge, England: Cambridge University Press, 1998.

Pelleg, D., and Moore, A. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 2000.

Peterson, G.L., Mills, R.F., McBride, B.T., and Alred, W.C. A Comparison of Generalizability for Anomaly Detection. *KDD-2005 Workshop on Data Mining Methods for Anomaly Detection*. August 21-25, 2005, Chicago, IL, pp. 53-57.

Thrun, S. Lifelong learning: A case study. Technical Report CMU-CS-95-208, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, 1995.

Wah, B.W. Generalization and Generalizability Measures. *IEEE Transactions on Knowledge and Data Engineering*. vol. 11, No. 1, 175-186, 1999.

Widmer, G., and Kubat, M. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*. vol. 23, No. 1, 68-101, 1996.

Wong C., C. Chen, and S. Yeh. K-Means-Based Fuzzy Classifier Design. *The Ninth IEEE International Conference on Fuzzy Systems*. vol. 1, pp. 48-52, 2000.