4-2007

Steganography Anomaly Detection Using Simple One Class Classification

Benjamin M. Rodriguez
Air Force Institute of Technology

Gilbert L. Peterson
Air Force Institute of Technology

Sos S. Agaian
University of Texas at San Antonio

Recommended Citation

Benjamin M. Rodriguez, Gilbert L. Peterson, and Sos S. Agaian "Steganography anomaly detection using simple one-class classification", Proc. SPIE 6579, Mobile Multimedia/Image Processing for Military and Security Applications 2007, 65790E (2 May 2007); https://doi.org/10.1117/12.717979

Steganography Anomaly Detection Using Simple One-Class Classification

Benjamin M. Rodriguez*[a], Gilbert L. Peterson[a], Sos S. Agaian[b]
[a]Department of Electrical and Computer Engineering
Graduate School of Engineering and Management, Air Force Institute of Technology;
[b]Multimedia and Mobile Signal Processing Laboratory
The University of Texas at San Antonio, Department of Electrical and Computer Engineering

ABSTRACT

There are several security issues tied to multimedia when implementing the various applications in the cellular phone and wireless industry. One primary concern is the potential ease of implementing a steganography system. Traditionally, the only mechanism to embed information into a media file has been with a desktop computer. However, as the cellular phone and wireless industry matures, it becomes much simpler for the same techniques to be performed using a cell phone. In this paper, two methods are compared that classify cell phone images as either an anomaly or clean, where a clean image is one in which no alterations have been made and an anomalous image is one in which information has been hidden within the image. An image in which information has been hidden is known as a stego image. The main concern in detecting steganographic content with machine learning using cell phone images is in training specific embedding procedures to determine if the method has been used to generate a stego image. This leads to a possible flaw in the system when the learned model of stego is faced with a new stego method which doesn't match the existing model. The proposed solution to this problem is to develop systems that detect steganography as anomalies, making the embedding method irrelevant in detection. Two applicable classification methods for solving the anomaly detection of steganographic content problem are single class support vector machines (SVM) and Parzen-window. Empirical comparison of the two approaches shows that Parzen-window outperforms the single class SVM most likely due to the fact that Parzen-window generalizes less.

Keywords: Classification, Steganography, Steganalysis, Anomaly Detection

1. INTRODUCTION

Several steganography methods exist for various forms of embedding within digital images which are shared between a sender and a receiver. As technology progresses, new practical forms of communication are developed, each providing malicious users opportunities to further exploit the transmission of digital data. In Wireless Steganography, Agaian *et al*. [1] shows techniques for sending secure communication over mobile devices using digital images. This type of application can be developed without the consent or knowledge of the manufacturer and service provider. Additionally the ability to apply steganography within mobile devices greatly increases the availability of this particular type of covert communication. While this has a significant impact on the transmission of classified data the same methods can be used for nefarious reasons.

Several forensics tools have been developed for digital forensics analysis of mobile devices. While there is no single solution for all the diverse requirements present in a mobile computing forensics investigation, steps can be taken to incorporate parts of various tools to help the investigator. For example, imaging (making an exact copy) of a suspect's mobile hard/flash drive is critical in the forensics process, and must ensure that every bit on the suspect's drive be copied exactly. Along with imaging tools, forensic analysis tools must be used based on the media and file types encountered. One common file type is digital image files (e.g., bmp, gif, jpeg, png, tiff, etc.). Most image types fall into one of three categories based on their embedding procedure, spatial (bmp, gif, png, raw), transform or lossy compressed (jpeg, jpeg2k), or vector (divides the space into discrete features). With these image types, there is the opportunity for an individual to conceal data in the image that under normal circumstances goes unnoticed. Each embedding procedure has its own unique steganography fingerprint, and presents different challenges. With this said, this paper focuses on the

development of a new multi pixel comparison steganography detection method used for digital images transmitted over mobile channels. The presented technique uses multiple masks to generate features and weights them to be sensitive enough to discriminate between clean and stego images. In this article, we show that for lossy compressed images one-class classification techniques reliably separate clean images from images that contain stego. The reason for using anomaly detection (AD) instead of traditional classification is that with over 250 stego tools available [2] and more being generated each day, assuming that we could not encounter something novel is not realistic.

The next section presents related work in the area of steganalysis as well as several embedding methods. This is followed in section 3 with a discussion on the feature selection for both spatial and transforms domain image types. In section 4 the anomaly detection algorithms using the image features are described. The results of our analysis are discussed in section 5.

2. RELATED WORK

Several digital cameras create lossy compressed images which are popular due to their small transmission size, and as a result several different embedding techniques are available:

- Discrete Cosine Transform (DCT) least significant bit (LSB) embedding [3] modifies the LSB of the DCT instead of the pixel value allowing the algorithms to encode in series or via a random walk technique.
- F5 [4] which was developed as a challenge to the steganalysis community makes use of matrix embedding by taking n coefficients and hashes to k bits with XOR hash into k message bits to determine which coefficient to change by decrementing absolute value.
- JP Hide and Seek [5] embeds data files with the use of an encryption algorithm requiring a password. The stego data is encoded with an encrypted file by modifying the LSBs of the DCT coefficients in a pseudo-random fashion.
- JPEG-JSteg [2] embedding encodes a converted data file into a JPEG image even if the stego data is a text document. JSteg has the unique property of accepting eight cover images and performs the lossy part of the JPEG encoding to include the blocking, DCT, and quantization.
- Model based [6] fits the coefficient histogram to an exponential model via maximum likelihood.
- OutGuess [7] modifies the LSB of the DCT by statistically checking the original image heuristics against the embedded image and manipulates nearby DCT blocks to maintain DCT histogram.
- Steghide [8] converts the secret data with the use of compression and encryption embedding the stego data within the cover files using a pseudo-random number generator.

The steganography methods identified are used for the analysis since cell phone steganography methods are platform dependent and not easily available for analysis. For the forensics practitioner, several steganalysis tools exist:

- ILook Investigator © toolsets
- Inforenz Forager® (Identifies Embedding Tools)
- SecureStego
- StegDetect
- WetStone Stego Suite™ (Identifies JP Hide n Seek, F5, JSteg, and Camouflage)

These tools currently assist the digital forensics examiner; however, there are improvements that can be made, specifically in the area of detecting stego without targeting specific embedding methods, as has been conducted in recent research.

In [9], Farid proposed a higher-order statistical model constructed from multi-scale wavelet decomposition of an image. This approach relies on building higher-order statistical models for natural images and looking for deviations from these models. Farid uses the statistical regularities which are obtained from the decomposition of images using wavelets in order to differentiate clean “natural” images from stego-images. The wavelet based method uses a large data base of over 40,000 images to develop feature vectors for classification. The basic motivation of this method is that Stego-images are perceptually identical to cover images, but they exhibit statistical irregularities. Farid argues that most steganalysis attacks look at only first order statistics. But new techniques try to keep the first order statistics intact. Farid [9] uses optimal linear predictor for wavelet coefficients and calculates the first four moments (i.e., mean, variance, kurtosis, and skewness) of the distribution of the prediction error. Various classifier types are then used to separate stego-message from cover-images. Agaian *et al*. [10] presented a local universal steganalysis technique which combines the advantages of wavelet coefficient, higher-order statistics based and DCT based localization for detecting embedding information. There are three basic components for this method: (1) novel DCT multilevel decomposition with wavelet structure, (2) a

new set of feature vectors and (3) a modified kernel function in the Kernel Fisher Discriminant. Inherently, the presented method captures stego information in small blocks while only using a small training set (a few hundred images), it can localize the hidden information and it can detect a low percentage of stego information in small blocks even in gray scale images. It is also shown that the new method has increased detection accuracy. In general, detection of as little as 2% embedding in a single 8 x 8 block of a jpeg-formatted image can be accomplished. However, these procedures have not been tailored for detecting steganography in cell phone images.

Detection of hidden information in apparently innocuous digital media is generally more challenging when nothing is known about the embedding method. Approaches which tackle this problem are known as blind or anomaly detecting algorithms. Several articles throughout the past few years have been proposed which address various methods for detecting steganographic information within digital images. McBride and Peterson [11] proposed a blind detection method for anomaly detection using a hyper-convex polytope to create a self class model, and a modified *k*-means using spherical and elliptical representations. In [8], Lyn and Farid exploited color statistics to show how a one-class support vector machine (SVM) greatly simplifies the training stage of the classifier by eliminating the need for training from stego images which makes for a blind classifier of common and future developed stego programs. Jackson [12] also used wavelet statistics in his work on blind steganalysis. Wavelet analysis offers more flexibility because it provides long time windows for low frequency analysis and a short time window for high frequency analysis. In the next section a feature extraction method is presented which is trained with a set of clean cell phone images to create a one-class system which identifies cell phone images.

3. FEATURE EXTRACTION

The first step in performing any type of stego detection requires the generation of a set of features that represent the statistical nature of the image. This section introduces the feature extraction method that is used to generate the set of features to train the classifiers for identifying clean images.

3.1 DCT Multilevel Energy Bands

The features focus on the energy band of the DCT coefficients. To target the vertical, horizontal, and diagonal energy bands in a compressed jpeg image, a modification to the discrete cosine transform is used which extracts the energy from the 8 x 8 blocks within the image set. To counter the effects caused by embedding within the edges of the blocks the DCT is also taken from adjacent pixel shifts, i.e., shifting the block down one pixel and right one pixel. Zigzag and peano scans are performed in each 8 x 8 DCT matrix to separate the vertical, horizontal and diagonal edge effects from the analyzed block. Higher-order statistics and predicted log errors are then calculated from the decomposition and used as the features for classification.

The standard DCT used in jpeg compression does not generate the multilevel energy bands which wavelet decomposition creates. In order to extract the various energy bands, the DCT transforms are rearranged in a wavelet decomposition structure. This structure is created by using the 8 x 8 pixel blocks from the jpeg compression technique. Rearranging the coefficients of the DCT splits the frequency spectrum into uniform spaced bands containing vertical, horizontal and diagonal energy. The presented structure captures the energy better than the normal DCT as well as some commonly used wavelet decompositions used in image processing. The result is significantly better energy concentration and similar properties of jpeg compression.

The features are then generated by calculating the difference between a target coefficient in the energy band separation with those of its vertical, horizontal and diagonal coefficient neighbors, shown in Figure 1. These coefficients result in an unstructured (non-Gaussian) distribution.

$$c = \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}, \; Q = \begin{bmatrix} q_{1,1} & \cdots & q_{1,n} \\ \vdots & \ddots & \vdots \\ q_{k,1} & \cdots & q_{k,n} \end{bmatrix}$$

where Q is a matrix of neighboring coefficients selected using the predictor pattern for the direction under consideration and c_i, $i = 1,\ldots,k$, is a vector of the coefficients to be predicted within an analyzed block.

In a similar manner the linear predictors are concatenated as well.

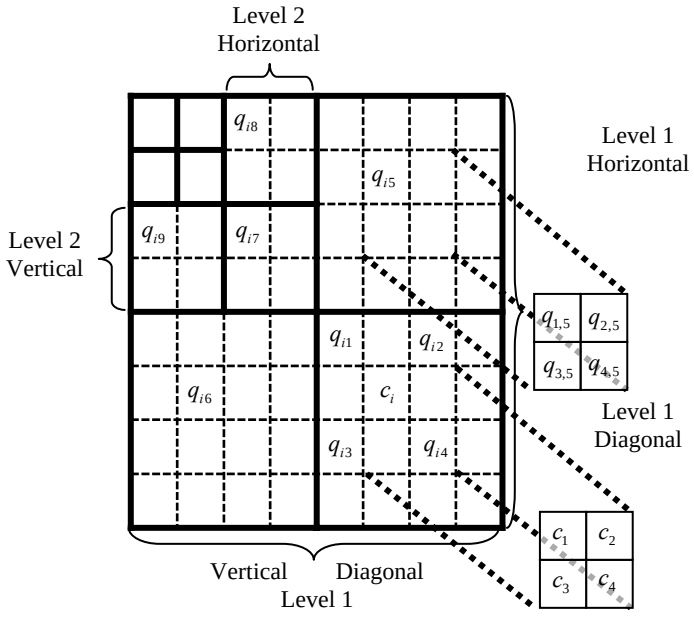


Figure 1: 9-Point Predictor Pattern

The pattern for the diagonal linear predictor is shown in Figure 1. The predictor DCT decomposition coefficients are created by weighting the neighbors, represented by q, such that the sum squared difference between their sums and coefficient to be predicted, c, is minimized. Rows and columns that lie on the edges of each direction and scale are compared to there adjacent 8 x 8 blocks when the error statistics are generated to preserve the predictor pattern from the properties of the DCT decomposition. The linear relationship in a matrix format can be represented as:

$$W = (Q^T Q)^{-1} Q^T c \tag{1}$$

where $W=(w_1,\ldots,w_9)^T$ and w is a scalar. The W contains the coefficients magnitudes of c_i represented as a column vector, and the matrix Q contains the neighboring coefficient magnitudes.

The predicted weight calculations of the coefficients and the coefficients of interest c_i are used to generate a measure of error. The log error for the linear predictor is defined in the following equation.

$$E = \log_2(c) - \log_2(|Wc|) \tag{2}$$

The features are generated with the separation of the vertical, horizontal and diagonal coefficients and predicted neighboring coefficients. These coefficients result in an unstructured (non-Gaussian) distribution. Using the measure E of the coefficients' and the coefficients c, higher order statistics features are calculated:

$$P(i,j) = \frac{C(i,j)}{\sum_{i,j} C(i,j)} = \frac{\text{number of non-zero coefficients}}{\text{total number of coefficients in the region}} \tag{3}$$

A three-level DCT analysis is performed on a suspect image, and statistics are calculated from the resulting coefficients. For each direction of the first two-levels of the DCT decomposition shown above, the ten statistics of the DCT coefficients are calculated. The higher order statistics are applied to the coefficients, c, and errors, E. The statistics of the error vectors for each direction and scale are calculated. For the first levels 30 features are generated; 10 from each of

the statistic measuring the vertical coefficients, 10 from each of the statistic measuring the diagonal coefficients and 10 from each of the statistic measuring the horizontal coefficients. Another 30 features are generated from each of the 10 statistics measuring the predictors from the first level and the second level for the three coefficient representations, vertical, horizontal and diagonal. This produces 60 features which represent changes made to the DCT coefficients from the embedding methods. Using the second and third level DCT decomposition coefficients produces another 60 features in a similar manner as the first and second levels. The features are combined with the three-level DCT decomposition coefficient statistics generated to form a 120 feature set which are used to identify various jpeg embedding methods. These feature sets generated from training sets of clean and stego images are used to train with various classification methods.

4. CLASSIFICATION METHODS

Machine learning for a classification task involves training and testing over a set of instances. Each instance in the training set contains one "target value" (class label) and several "attributes" (features). The learning objective is to sort data into groups or classes so that the degree of association is strong between the instances of the same class and weak between members of different classes. Once these relationships are generated using the features from Section 3, a new image can then be classified as containing steganographic content (belonging to the class in which data contains stego) or as clean (no steganographic content). In this section two blind-classification methods are presented, Parzen-window [13] and single class Support Vector Machine [14-18].

4.1 Parzen-Window

Parzen estimation is a refinement of histogramming. The Parzen-window density estimator depends on a user defined window width and the number of data samples. The basic idea behind Parzen-window estimation is that the knowledge gained by each training sample **x** of the input space, $\mathbb{R}^n$, is represented by a function centered at **x** in the feature space. The functions themselves are represented with the use of a distance measure or a *kernel* estimator. The final class estimation is derived by summing the results from the kernel functions of each training sample:

$$p_k = \frac{1}{\ell_k} \sum_{i=1}^{\ell_k} K(\mathbf{x}, \mathrm{x}_i) \tag{4}$$

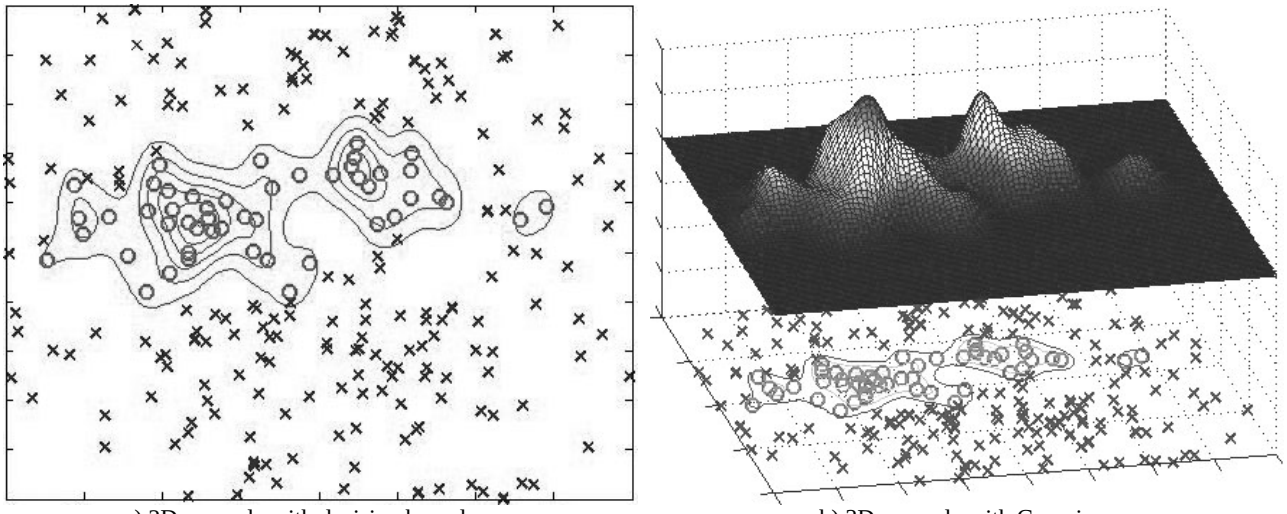
For example, the Parzen-window density model is optimized by maximizing the likelihood of the training data with the use of a Gaussian window surrounding each input data point. The Gaussian window can be represented with the use of a kernel function $K(\mathbf{x},\mathrm{x}_i)$ as an interpolation function which defines an inner product between the individual training sample. The Radial Basis kernel function uses a window width parameter,σ, which is also known as the spread of the function:

$$p_k = \frac{1}{\ell_k} \sum_{i=1}^{\ell_k} \frac{1}{\sqrt{(2\pi)^{\ell_k}}\sigma^{\ell_k}} \exp\left(-\frac{\|\mathbf{x}-\mathrm{x}_i\|^2}{2\sigma^2}\right) \tag{5}$$

This results in a sum of small multivariate Gaussian probability distributions centered at each training sample **x,** an example is shown in Figure 2. As the density of the training samples and their respective Gaussian distributions increase the estimation of the probabilities approach the true probability distribution function (PDF) of the training samples. The estimation for classification for a data cluster is then based on a threshold set for the combined posterior probability from all samples. The classification decision assigns the samples to the class with maximal posterior probability according the inequality:

$$\frac{1}{\ell_k} \sum_{i=1}^{\ell_k} \frac{1}{\sqrt{(2\pi)^{\ell_k}}\sigma^{\ell_k}} \exp\left(-\frac{\|\mathbf{x}-\mathrm{x}_i\|^2}{2\sigma^2}\right) > \frac{1}{\ell_j} \sum_{i=1}^{\ell_j} \frac{1}{\sqrt{(2\pi)^{\ell_j}}\sigma^{\ell_j}} \exp\left(-\frac{\|\mathbf{x}-\mathrm{x}_i\|^2}{2\sigma^2}\right), \ \forall\ k \neq j \tag{6}$$

This method requires a reasonably large training data set and is computationally inexpensive during training but is computationally expensive for testing, as the kernel function must be computed comparing the new sample with all of the existing training samples.



a) 2D example with decision boundary b) 3D example with Gaussian curve

Figure 2: Parzen-window decision boundary

4.2 Support Vector Machine (SVM)

SVM is a classification algorithm that provides state-of-the-art performance in a wide variety of application domains [14]. The goal of the SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes. The support vector classifier distinguishes between two or more classes and does not consider outliers not belonging to any of the classes. In this paper a one-class SVM is used for blind classification which obtains a boundary around a one-class data set. In the simplest case a hyper sphere, similar to hyper ellipsoid in Figure 3, is used to enclose all target objects. This model gives a closed boundary around the data set, often referred to as a hyperplane.

A SVM performs pattern recognition for two-class problems by determining the separating hyperplane that has maximum distance to the closest points of the training set [17,18]. These closest points are called support vectors. In order to do this, the SVM performs a nonlinear separation in the input space by using a nonlinear transformation $\phi(\cdot)$ that maps the data points $\mathbf{x}$ of the input space, $\mathbb{R}^n$, into a higher dimensional space, called kernel space $\mathbb{R}^p$ ($p > n$). The mapping $\phi(\cdot)$ is represented in the SVM classifier by a kernel function $K(\mathbf{x},\mathrm{x}_j)$ which defines an inner product in $\mathbb{R}^p$. Given ℓ samples $\left\{\left(\mathrm{x}_i, y_i\right)\right\}_{i=1}^{\ell}$, the decision function of the SVM is linear in the feature space and can be written as:

$$f\left(\mathbf{x}\right) = \mathbf{w}\phi\left(\mathbf{x}\right) + b = \sum_{i=1}^{\ell} \alpha_i^0 y_i K\left(\mathbf{x}, \mathrm{x}_i\right) + b \tag{7}$$

The optimal hyperplane is the one with the maximal distance (in space $\mathbb{R}^p$) to the closest points $\phi(\mathrm{x}_i)$ of the training data. Determining the hyperplane requires maximizing the following function with respect to α

$$W\left(\boldsymbol{\alpha}\right) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j K\left(\mathrm{x}_i, \mathrm{x}_j\right) \tag{8}$$

where under constraints $\sum_{i=1}^{\ell} \alpha_i y_i = 0$, $C \geq \alpha_i \geq 0$, $i = 1,\ldots,\ell$ and $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_\ell)$ is the non-negative Lagrange multipliers. The indexes of α_i which have nonzero values when a solution as described below is found correspond to the support vectors. Writing $W(\boldsymbol{\alpha})$ in matrix notation, incorporating non-negativity of $\boldsymbol{\alpha}$ and constraint, the following dual quadratic program is defined:

$$\text{Maximize} \qquad W\left(\boldsymbol{\alpha}\right) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j K\left(\mathrm{x}_i, \mathrm{x}_j\right) = \boldsymbol{\alpha}\cdot 1 - \frac{1}{2}\boldsymbol{\alpha} H \boldsymbol{\alpha} \tag{9}$$

Subject to:

$$\boldsymbol{\alpha} \cdot \boldsymbol{y}^T = 0$$
$$\boldsymbol{\alpha} \geq 0$$

where $\boldsymbol{y} = (y_1,\ldots,y_\ell)$ and H is a symmetric $\ell \times \ell$ matrix with elements $H_{ij} = y_i y_j K(\mathrm{x}_i,\mathrm{x}_j)$, which is a Hessian.

An upper bound on the expected error probability P_{error} of a SVM classifier is given by

$$f(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x}) + b = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathrm{x}_i) + b \tag{10}$$

where $\mathbf{w} = (\mathrm{w}_1,\ldots,\mathrm{w}_p)$. The contribution of a feature x_i to the decision function in equation 10 depends on α_i. In [15,16], $f(\mathbf{x})$ is used to denote the current hypothesis which is determined by the values of the dual variable and the bias, b, at a particular stage of learning. The error, E, is then calculated as the difference between the function output ant the target classification on the training points x_i and the support vectors x_j as follows:

$$E_i = f(\mathbf{x}_i) - y_i = \left(\sum_{j=1}^{\ell} \alpha_j K(\mathrm{x}_i, \mathrm{x}_j) + b \right) - y_i \tag{11}$$

It should be noted that the value of E_i may be large for a correctly classified input feature x_i.

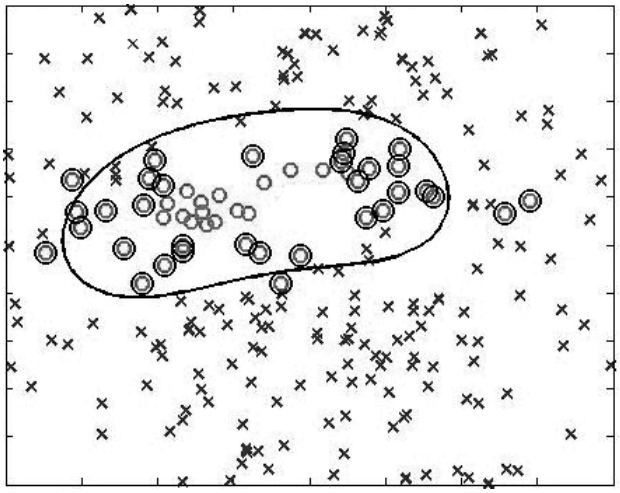


Figure 3: SVM optimal hyperplane

To minimize the chance of accepting outliers, the volume of this hyper sphere is minimized [18]. This model does not always create a closed boundary around the entire targeted data set. To create a bendable boundary the inner product used to calculate the hyper sphere is replaced by a Radial Basis kernel function, $K(\mathbf{x},\mathrm{x}_i)$, which implicitly maps the data set into another feature space. An ideal kernel function would map the target data onto a bounded area in the feature space and outlier objects are outside this bounded area. This creates a one-class SVM classifier, for performing anomaly detection [18].

5. EXPERIMENTAL RESULTS

This section presents the results of performing anomaly detection using each of the previously discussed classification algorithms, Parzen-window and Support Vector Machine, for mobile jpeg embedding methods. Testing is performed using 5-fold cross validation in which the partitions consist of 80% clean data used for training and the untrained upon 20% clean and 20% of the stego image data is used for testing, repeated five times each time with a different 20% of the data being used for testing.

Two sets of test were conducted in this section. The first test uses 150 256 x 256 grayscale clean high-quality jpeg images taken with a Nikon D100 and steganography images consist of six sets of 150 images each for each embedding method (F5 [4], JP Hide [5], JSteg [2], Model Base [6], OutGuess [7] and StegHide [8]). The amount of hidden information embedded within each of the files was 4000 characters which is equivalent to one page of text. The percentage of altered coefficients varies based on the embedding method, but is in the range of 5% to 25%. The classification with this method is based on the probability of an exemplar belonging to a clean class with a probability of 15%.

Table 1. 5-fold cross validation for the one-class classification of jpeg domain images using Parzen-window and SVM

SVM (Radial Basis Function)

ACTUAL	PREDICTED		
		Anomaly	Clean
	Anomaly	94.21±1.7%	24.0±4.3%
	Clean	5.79±1.7%	76.0±4.3%

Parzen-window

ACTUAL	PREDICTED		
		Anomaly	Clean
	Anomaly	97.65±1.5%	10.66±11%
	Clean	2.34±1.5%	89.33±11%

The results shown in Table 1 indicate that the classifiers perform well when classifying an anomaly (i.e., true positive) using Parzen-window and SVM methods. In the case of classifying anomalous data, Parzen-window outperform SVM by approximately 4%. For the embedding conditions, jpeg images, the detection algorithms perform well, resulting in an ability to determine if an image has steganography. One reason for this is that the generalization which occurs using the Parzen-window classifier is limited by the window width parameter and the threshold value, where as with the SVM, the generalization is bound by the classes distance from the origin of the space which can result in a larger bounding area.

In the second test the images used for analysis are from a Nokia 6620 (Symbian operating system) camera phone. The images are of size 640 x 480 with 100 clean mid-quality Symbian jpeg images and steganography images consist of six sets of 50 images each for every embedding method (F5, JP Hide, JSteg, Model Base, OutGuess and StegHide). The images are downloaded onto a Windows PC and converted to steganography images using the six embedding methods. These images are then uploaded into the Nokia 6620 for transmission. The amount of hidden information embedded within each of the files was 4000 characters which is equivalent to one page of text. The percentage of altered coefficients varies based on the embedding method and is in the range of 4% to 30%. The classification with this method is based on the probability of an exemplar belonging to a clean class with a probability of 25%.

Table 2. 5-fold cross validation for the one-class classification of Nokia 6620 jpeg images using Parzen-window and SVM

SVM (Radial Basis Function)

ACTUAL	PREDICTED		
		Anomaly	Clean
	Anomaly	88.0±2.1%	10.0±2.5%
	Clean	12.0±2.1%	90.0±2.5%

Parzen-window

ACTUAL	PREDICTED		
		Anomaly	Clean
	Anomaly	91.5±1.2%	8.9±1.3%
	Clean	8.5±1.2%	91.1±1.3%

The results shown in Table 2 indicate that the classifiers begin to identify the differences between an anomaly (i.e., true positive) and a clean Nokia jpeg images (i.e., true positive) using Parzen-window and SVM methods. In the case of classifying anomalous data, Parzen outperform SVM by approximately 3%.

6. CONCLUSION

In this article analysis was conducted with two types of jpeg compressed images, images taken with a Nikon D100 digital camera and the second set of images taken with the Nokia 6620 camera phone. The steganography methods used in this analysis consisted of six popular methods available over the Internet. The jpeg images taken with the Nikon D100 were identified by the classifier when alterations were made to the original images. The image set taken by the Nokia 6620 camera phone show a true separation between the clean and anomalous images, as depicted in Table 2. This is an indication that the steganalysis system being trained for cell phone images is beginning to reach a point of reliability. In

future work the steganalysis system is to be trained with a larger set of images at various jpeg compression qualities in order to improve the overall classification accuracy.

ACKNOWLEDGEMENTS

The work on this paper was 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. We would additionally like to express our appreciation to June Rodriguez for the contribution of a multitude of digital images for analytical support.

REFERENCES

1. S. Agaian, G. Peterson and B. Rodriguez , "Multiple Masks Based Pixel Comparison Steganalysis Method for Mobile Imaging", IS&T/SPIE, Defense and Security Symposium, Mobile Multimedia/Image Processing For Military And Security Applications, 17-21 April 2006
2. StegoArchive.com, http://www.stegoarchive.com/
3. T. Sharp, "An Implementation of Key-Based Digital Signal Steganography", Lecture Notes in Computer Science, Volume 2137/2001, Information Hiding: 4th International Workshop, IHW 2001, Pittsburgh, PA, USA, April 25-27, 2001. Proceedings
4. A. Westfeld, "High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm)", In: Moskowitz, I.S. (eds.): Information Hiding. 4th International Workshop. Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg New York (2001) 289–302
5. N. Provos, and P. Honeyman, "Hide and Seek: An Introduction to Steganography", *IEEE Security & Privacy Magazine*, May/June 2003
6. P. Sallee, "Model-based steganography," International Workshop on Digital Watermarking, Seoul, Korea, 2003
7. N. Provos, *OutGuess*. http://www.outguess.org/
8. S. Hetzel, StegHide, http://steghide.sourceforge.net/
9. S. Lyu and H. Farid, "Steganalysis Using Color Wavelet Statistics and One-Class Support Vector Machines, SPIE Symposium on Electronic Imaging, San Jose, CA, 2004. spie04
10. S.Agaian, "Color Wavelet Based Universal Blind Steganalysis", The 2004 International Workshop on Spectral Methods and Multirate Signal Processing, SMMSP 2004
11. B. McBride, and G. Peterson, "Blind Data Classification using Hyper-Dimensional Convex Polytopes", *Proceedings of the 17th International FLAIRS Conference 200*4. 520-527
12. J. T. Jackson, *TARGETING COVERT MESSAGES: A UNIQUE APPROACH FOR DETECTING NOVEL STEGANOGRAPHY*, MS thesis, AFIT/GCE/ENG/03-02 Air Force Institute of Technology Wright-Patterson AFB OH, March 2003
13. R. Duda, P. Hart and D. Stork, *Pattern Classification*, Second Edition, Wiley 2001
14. C. Burgers, "A tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery"
15. N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press 2000
16. C.-W. Hsu, C.-C. Chang, C.-J. Lin. "A practical guide to support vector classification", http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
17. B. Scholkopf, A. J. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", The MIT Press, 2002
18. D.M.J. Tax and R.P.W. Duin, "Support Vector Data Description", *Machine Learning*, vol. 54, no. 1, 2004, 45-66

*benjamin.rodriguez@afit.edu; phone 1 (937) 255-3636 ext. 7543; fax 1 (937) 656-4055