Chapter 1

FUSION OF MULTI-CLASS STEGANALYSIS SYSTEMS USING BAYESIAN MODEL AVERAGING

Benjamin M. Rodriguez, Gilbert L. Peterson, and Kenneth W. Bauer

Abstract Several steganography methods are available over the Internet for hiding information within digital images. A digital forensics examiner must be able to extract a hidden message from a digital image. Extraction requires first identifying the embedding method used. Several steganalysis systems exist that identify a subset of the available embedding methods. Each steganalysis systems has its own particular identification strengths and weaknesses. This paper applies Bayesian model averaging to fuse these multiple systems and identify the embedding method used to create a stego JPEG image. The fusion of the multi-class detection systems results in better classification than the individual systems, with an accuracy improved from 80% for the individual multi-class steganalysis systems to 90% with the steganalysis fusion system.

Keywords: Steganalysis, Multi-class Fusion, Bayesian Model Averaging

1. Introduction

The problem of steganalysis has moved from simply determining if an image contains hidden information to extracting the hidden message. Extraction requires the intermediate step of identifying the method used to create the steganography image. With over 250 steganography methods available over the Internet it is important to develop multi-class steganalysis systems capable of properly labeling an image as containing a specific type of steganography.

Several detection systems are available including research tools [4, 9, 11, 14, 18, 22] and commercially available systems (ILook Investigator© toolsets, Inforenz Forager®, SecureStego, StegDetect [12], WetStone Stego Suite^{TM}). Each systems has its own advantages and disadvan-

tages. With so many detection systems available to the steganalyst, a problem arises in deciding which detection system is best to use. A solution to this problem is to fuse the results from each of the systems to more accurately identify the embedding method.

This paper uses Bayesian model averaging [6] to combine several multi-class steganalysis systems to achieve an increase in detection performance. The dataset under consideration is a seven class steganalysis problem with samples consisting of clean images (one class) and steganography images (six classes). The steganography methods tested against in this paper are F5 [21], JP Hide and Seek [8], JSteg [20], Model Base [16], OutGuess [13] and StegHide [5]. This steganalysis fusion system fuses four multi-class steganalysis methods. The first steganalysis system is a prepackaged detection system named StegDetect [12] capable of detecting F5, JP Hide and Seek, JSteg, and OutGuess. The other three systems are one-vs-one multi-class classifiers that use a two-class support vector maching (SVM) as the classifier. These three individual multi-class method differ in the features used for classification [9, 4, 15].

The following section presents related commercially available and research based multi-class classification systems. Following this is a description of the steganalysis fusion system. Testing results from the system include individual multi-class system results and results from the stegnalaysis fusion system.

2. Related Work

This section briefly discusses available detection systems consisting of complete packages available as commercial products and research systems.

The commercial detection systems available as automated tools are designed to give the analyst an initial indication if a set of images contains hidden information. For the forensics practitioner, several steganalysis tools exist that can be used such as: ILook Investigator© toolsets, Inforenz Forager®, SecureStego, StegDetect [12], and WetStone Stego Suite™.

These tools currently assist the digital forensics examiner; however, not all embedding methods are targetted by all of the tools. Specifically, StegDetect, which is one of the detection systems in our steganalysis fusion system, detects four (F5, JP Hide and Seek, JSteg and OutGuess) of the six targeted embedding methods.

Multi-class research steganalysis detection systems tend to follow a five step process, shown in Figure 1. The first step creates a data set

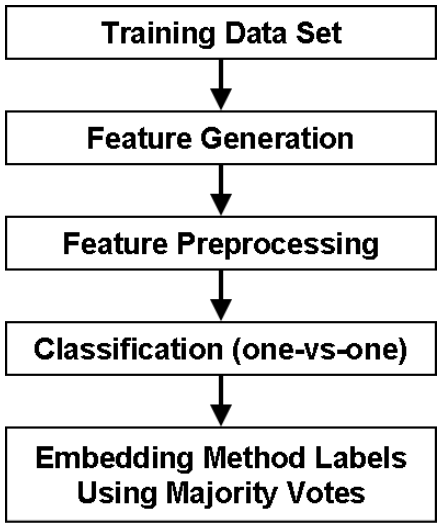

Figure 1. Block Diagram of Multi-Class Detection System.

containing both clean and stego images used in training the multi-class detection system.

In the second step, features are generated from an input image. By generating features from an image, the information sent the classifier is significantly reduced.Several feature generation techniques exist, those focused on in this work consist of Lyu and Farid's [9] wavelet based method, a DCT based feature generation method [11], and a method that generates features from DCT decomposed coefficients [15].

In the feature preprocessing stage, step three, two procedures are used. In the first procedure, the set of input features are normalized. This reduces the chance that features with large values would have a larger influence in the cost function than features with small values. The second procedure in the feature preprocessing stage selects the most important features to reduce the number of features and retain as much class discrimination capability as possible.

Many of the multi-class classifiers for steganalysis [11, 14], use a two-class SVM classification method in conjunction with a one-vs-one approach to merge the individual classifiers to generate a multi-class classification. Multiple SVM classifiers are trained to distinguish each embedding method from each of the others and clean images. The multi-class classification counts the votes from each SVM classifier and the final classification (i.e. identification of the embedding method) is determined by the classification with the most votes.

In the next section, the implementation of the steganalysis fusion system is described in detail. Four of the systems mentioned in this section: StegDetect [12], wavelet feature generation [9], DCT based feature generation [11], and DCT decomposition feature generation [15] are used in the steganalysis fusion system.

3. Method

Before discussing the steganalysis fusion system, this section presents details on the feature generation, classification, and labeling steps of the multi-class detection system, Figure 1. The steganalysis fusion system, uses the mutli-class classifiers generated using this process with Bayesian model to create creating the fused multi-class steganalysis detection system.

The steganography techniques targeted in this paper, include F5 [21], JP Hide and Seek [8], JSteg [20], Model Base [16], OutGuess [13], and StegHide[5]. All of these embedding methods, hide data by manipulating the quantized Discrete Cosine Transform (DCT) values generated during the JPEG image compression process.

3.1 Feature Generation

Three feature generation methods, wavelet feature generation [9], DCT based feature generation [11], and DCT decomposition feature generation [15] are used in the creation of three different multi-class steganalysis classification tools.

Wavelet feature generation first performs a multi-scale Haar wavelet decomposition of an image [9]. This is followed by the extraction of higher-order statistics calculated over each of the pixels in the wavelet and the pixels relationship to its neighbors in both the current and higher scales. In this manner, 36 coefficient statistics, and 36 error statistics are calculated to yield a total number of 72 statistics. These statistics form the feature vectors used to discriminate between clean and stego images.

DCT based feature generation calculates first and second order features over the DCT values and pixel values (spatial domain) of an image [11]. The features in the DCT and spatial domains are calculated from a number of functions applied to the stego JPEG image. These functions consist of the global DCT coefficient histogram, a co-occurrence matrix, spatial blockiness, and several other functions [11]. The stego image is decompressed to the spatial domain, cropped by four pixels in each direction and recompressed with the same quantization table used in decompression. By applying the same functions to the cropped image,

an approximation of the hidden information is generated. This feature generation technique results in 274 features.

DCT decomposition feature generation divides a processed DCT block into directional and frequency bands [15]. The DCT coefficients are separated into low, medium and high frequencies as well as vertical, diagonal and horizontal directions. This is referred to as DCT decomposition. In addition to the decomposition, the coefficients are categorized into raw, shifted and predicted coefficients. The shifted coefficients are used to identify embedding blockiness between neighboring 8 by 8 blocks. The predicted coefficients estimate the coefficients altered by an embedding method. The features are generated by calculating the higher order statistics: 1st, 2nd, 3rd, and 4th moments, 2nd, 3rd, and 4th central moments, and entropy, over the sets of selected coefficients. This produces 234 total features consisting of 72 shifted coefficients, 72 raw coefficients, 72 predictors and 18 histogramming features.

3.2 Support Vector Machine

The support vector machine (SVM) is a classification algorithm that provides state-of-the-art performance in a wide variety of application domains [1, 17]. The goal of the SVM is to produce a model that predicts the class of data instances in the testing set given only the attributes. A SVM performs pattern recognition for two-class problems by determining the separating hyperplane that has maximum distance between the closest points of each class in the training set. The closest points to the hyperplane are called support vectors. In order to do this, the SVM performs a nonlinear separation of the input space by using a nonlinear transformation $\phi(\cdot)$ that maps the data instances x (with features denoted x_i) from the input space into a higher dimensional space, called kernel space. The mapping, $\phi(\cdot) \rightarrow \phi(x_i)$, is performed in the SVM classifier by a kernel function $K(\cdot, \cdot)$. The decision function of the SVM is linear in the kernel space although not in the feature space. In this paper, the SVM method used is LibSVM [2] which uses a Sequential Minimal Optimization (SMO) for binary SVM with an L1-soft margin [3].

3.3 One-vs-One

In multi-class classification methods, two-class classifiers are combined using a one-vs-one methodology [19]. This method trains several classifiers, each individual classifier compares one class against one of the other classes. For k classes, this produces $k(k-1)/2$ classifiers that each vote on the class assignment for a data instance. The algorithm then

identifies the final classification as the class with the highest vote. The goal is to train the multi-class rule based on the majority vote strategy. This is a fairly reliable method assuming that the feature space is separable between the various classes.

In the steganalysis fusion system, seven classes (1 clean + 6 stego), $k = 7$, are targeted, requiring 21 classifiers to be trained. The output of each SVM is a vote that is tallied. The classification with the majority of votes for a class wins.

3.4 Multi-class Detection System

In developing the multi-class detection system there are five steps (refer to Figure 1) as follows:

1. Training Data Set: Define a training data set in which the number of classes have been assigned. In this paper the data set consists of clean and stego images. The stego images are created using six embedding methods (F5, JP Hide and Seek, JSteg, Model Base, OutGuess and StegHide).

2. Feature Generation: Generated features are created from each JPEG image. This step uses three feature generation methods [9, 11, 15] to develop three distinct multi-class systems.

3. Feature Preprocessing: Feature preprocessing normalizes the feature values and a subset of the features are selected based on the Fisher's disriminant ratio ranking. Other methods of preprocessing for outlier removal, data normalization, feature selection, and feature extraction could also be used [7].

4. Classification: The classification step trains each one-vs-one classifier from the training data set using a SVM, Section 1.3.3.

5. Majority Vote: In this step a class label is assigned based upon a majority vote from each classifier.

3.5 Bayesian Model Averaging

Bayesian model averaging merges together several multi-class classifiers by combining the probabilistic density estimation of each classifiers classification accuracy as a mixture of Gaussians [6, 10]. Murphy's [10] Bayes Net Toolbox (BNT) for Matlab was used in the analysis to facilitate the computations in the model averaging. The proba-

bilistic density estimation specifies the local *conditional probability distributions* (CPD) for a classification model, M_k, where k is one of K classifiers, and M is the set of all classifiers. The CPD of each model M_k is $p(M_k|T)$. This represents for each class, the probability of what a classification model will classify a target instance T as. For example, given a target image that contains data hidden using JP Hide and Seek, $p(M_k|T =' JPHideandSeek')$ represents the probability distribution over all of the possible classifications M_k could make, i.e. F5, JP Hide and Seek, etc. In our implementation, the confusion matrix, which represents the correct and incorrect classification for a multi-class classifier, provides the probabilistic density estimation for each classifier.

The fusion process then uses the classifications from the classification models, M, and calculates the joint probability distribution over each target classification, $T = c$:

$$p(T = c|M) = \eta \prod_{k=1}^{K} p(M_k|T = c)p(T = c). \quad (1)$$

The final classification is then the target classification, $T = c$, with the highest probability. The prior probabilities of $p(T)$ are calculated from the number of clean, and each type of embedding method images used in testing.

An example Bayesian model averaging system is shown in Figure 2. The four nodes at the top represent the classifiers and CPDs for each M_k. The Bayesian Model Averaging node contains the $p(T)$ CPD that merges the results of the four models and makes the final classification.

3.6 Bayesian Model Averaging Structure for Steganalysis

The process for using Bayesian model averaging for steganalysis consists of 7 steps. The general process is:

1. Generate Features.
2. Select Relevant Features.
3. Create Classification Model based on one-vs.-one training.
4. Using majority vote strategy to populate the confusion matrix containing actual and predicted classified values for clean, F5, JP Hide and Seek, JSteg, Model Base, OutGuess and StegHide trained image sets.

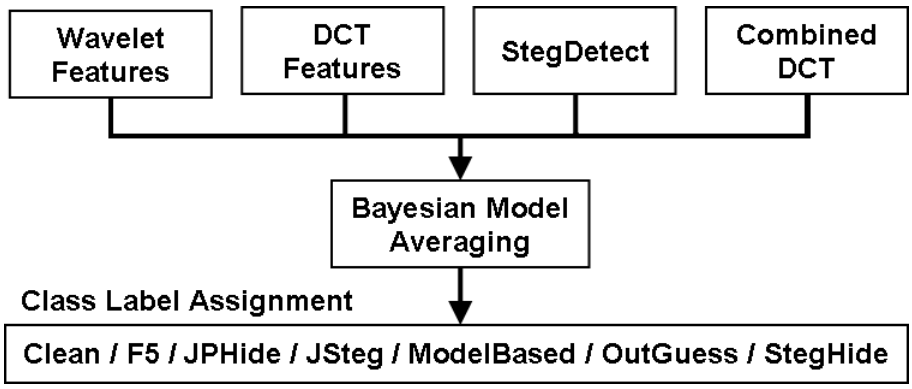

Figure 2. Block Diagram of Bayesian Model Averaging Structure.

5. Repeat Steps 1 thru 4 for each of the Feature Generation methods, [9, 11, 15].
6. Create a confusion matrix for StegDetect [12].
7. Use the four confusion matrices as classifier models for the Bayesian averaging model.

This results in a multi-class model that receives four inputs, three from each of the trained detection system and one from StegDetect, for an individual image to be classified. The resultant steganalysis fusion system is shown in Figure 2.

4. Results

The results presented are based on a 1000 image data set of 512 by 512 RGB JPEGs consisting of clean and stego images. The training set of images consisted of 200 clean images and 100 each for each embedding method (F5, JP Hide and Seek, JSteg, Model Based, OutGuess and StegHide). The test set contains 50 clean and 25 for each embedding method. The clean images in this test set did not overlap with images in the stego image sets, nor did any of the images from one stego type overlap with another; for example, none of the F5 images were the same as the JSteg images. The amount of hidden information embedded within each of the files is 4000 characters, which is equivalent to one page of text. The percentage of altered coefficients based on each embedding method is:

- F5 has an average of 0.3% of the coefficients altered.

Table 1. Test Set Classification Accuracy for Individual Detection Systems.

Image Type	Wavelet Features	DCT Features	StegDetect	Combined DCT Features
Clean	45.4 ± 1.1	42.6 ± 2.1	40.6 ± 1.1	42.8 ± 0.8
F5	21.4 ± 0.8	24.2 ± 1.8	25.0 ± 0.0	18.0 ± 0.7
JP Hide	22.2 ± 0.5	21.8 ± 0.8	17.4 ± 1.1	20.0 ± 1.0
JSteg	20.8 ± 0.8	22.0 ± 1.6	20.0 ± 2.1	22.8 ± 0.8
Model Based	13.2 ± 1.3	16.4 ± 0.5	0.0 ± 0.0	17.8 ± 0.5
Outguess	17.0 ± 0.7	13.8 ± 0.5	17.4 ± 2.1	18.4 ± 0.5
StegHide	17.6 ± 1.1	16.4 ± 0.5	0.0 ± 0.0	18.0 ± 0.7

- JP Hide and Seek has an average of 2.8% of the coefficients altered
- JSteg has an average of 6.7% of the coefficients altered
- Model Base has an average of 7.8% of the coefficients altered
- OutGuess and StegHide have an average of 1% of the coefficients altered

The testing is performed using 5-fold cross validation. It should be noted that the presented results are not benchmarking any individual system against the others. Rather the results are used to show how the steganalysis fusion systems takes advantage of the strengths of each of the individual systems and improves accuracy.

The results for the individual systems are shown in Table 1 and the steganalysis fusion system results are shown in Table 2. In Table 1, by examining each of the confusion matrices, no single multi-class classification algorithm outperforms the others. For example, StegDetect detects all of the F5 images, wavelet feature generation (Wavelet) [9] incorrectly labels the fewest clean images as stego, DCT based feature generation (DCT) [11] identifies the largest number of JP Hide and Seek images, and the DCT decomposition feature generation (Combined DCT) [15] identifies the most OutGuess and Model Based images.

The results of the steganalysis fusion system are shown in Table 2. The steganalysis fusion system consistently out performs the individual systems. The only case in which the steganalysis fusion system does not outperform the individual systems is in the case of F5 embedding, for which both the fusion system and StegDetect detect all of the images.

5. Conclusion

The steganalysis fusion systems improves the accuracy of multi-class steganalysis systems. The novel approach presented builds a steganalysis fusion system using three multi-class SVM classifiers each using

Table 2. Bayesian Model Averaging Confusion Matrix Results.

		Predicted						
Actual		Clean	F5	JP	JS	MB	OG	SH
Clean	Ave.	46.8±	0.8±	0.2±	0.2±	0.2±	1.6±	0.2±
	σ^2	0.8	0.5	0.5	0.5	0.5	0.9	0.5
F5	Ave.	0.0±	25.0±	0.0±	0.0±	0.0±	0.0±	0.0±
	σ^2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JP	Ave.	0.0±	0.0±	23.6±	1.4±	0.0±	0.0±	0.0±
	σ^2	0.0	0.0	0.6	0.6	0.0	0.0	0.0
JS	Ave.	0.0±	0.0±	1.4±	23.2±	0.0±	0.4±	0.0±
	σ^2	0.0	0.0	0.6	0.8	0.0	0.6	0.0
MB	Ave.	4.6±	1.6±	0.0±	0.0±	18.0±	0.2±	0.6±
	σ^2	0.6	0.6	0.0	0.0	0.7	0.5	0.6
OG	Ave.	1.8±	0.4±	0.0±	0.0±	0.0±	18.8±	4.0±
	σ^2	0.5	0.6	0.0	0.0	0.0	0.5	0.7
SH	Ave.	1.2±	0.0±	0.0±	0.0±	0.0±	2.6±	21.2±
	σ^2	0.5	0.0	0.0	0.0	0.0	0.6	0.8

one of three different feature extraction methods. The steganalysis fusion system uses Bayesian model averaging to correctly determine the embedding algorithm. The combination of several steganalysis systems improves the overall detection accuracy of the multi-class system.

In future work, the steganalyst can add new detection systems to the presented fusion system as they are developed. In addition, the data sets should be expanded to various image sizes and various JPEG compression ratios. These suggestions should improve an investigators chance of correctly identifying the steganography embedding method as embedding signatures, image sizes, and compression change.

6. Acknowledgments

This research was partially funded by the US Air Force Research Laboratory, Information Directorate/Multi-Sensor Exploitation Branch, Rome New York. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Air Force, Department of Defense or the U.S. Government.

References

[1] C. Burgers, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 2(2), pp. 121-167, 1998.

[2] C.-C. Chang, and C.-J. Lin, *LIBSVM: a library for support vector machines*, Retrieved February 2007,

http://www.csie.nte.edu.tw/ cjlin/libsvm.

[3] N. Cristiani, and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[4] J. Fridrich, Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes, *LNCS 6th Information Hiding Workshop*, eds. J. Fridrich, Springer-Verlag, pp. 67-81, 2004.

[5] S. Hetzl, StegHide. Retrieved July 2005, http://steghide.sourceforge.net/, 2003.

[6] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky, Bayesian Model Averaging: A Tutorial (with discussion), *Statistical Science*, 14(4), pp 382-417, 1999.

[7] A.K. Jain, R.P.W. Duin, and J. Mau, Statistical Pattern Recognition: A Review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), pp. 4-37, 2000.

[8] A. Latham, Steganography, Retrieved July 2005, http://linux01.gwdg.de/ alatham/stego.html

[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.

[10] K. Murphy, The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, Volume 33,2001.

[11] T. Pevny and J. Fridrich, Merging Markov and DCT Features for Multi-Class JPEG Steganalysis, *Proceedings of SPIE Electronic Imaging, Photonics West*, pp. 03-04, 2007.

[12] N. Provos, OutGuess, Retrieved July 2006, http://www.outguess.org/, 2004.

[13] N. Provos, and P. Honeyman, Hide and Seek: An Introduction to Steganography, *IEEE Security & Privacy Magazine*, May/June 2003.

[14] B.M. Rodriguez, and G.L. Peterson, Steganography Detection Using Multi-Class Classification, *Advances in Digital Forensics III*, eds. Craiger and Shenoi, Springer Science+Business Media, New York, NY, pp. 193-204, 2007.

[15] B.M. Rodriguez, G.L. Peterson, and R. Neher, DCT Combined Directional and Frequency Band Distance Measure Features, *IEEE Transactions on Information Forensics and Security*, submitted.

[16] P. Sallee, Model-based steganography, *International Workshop on Digital Watermarking*, Seoul, Korea, 2003.

[17] B. Scholkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, The MIT Press, 2002.

[18] Y.Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, Image Steganalysis Based on Moments of Characteristic Functions Using Wavelet Decomposition, Predicition-Error Image, and Neural Network, *IEEE International Conference on Multimedia and Expo*, 2005.

[19] D. M. J. Tax and R. P. W. Duin, Using Two-Class Classifiers for Multiclass Classification, *International Conference on Pattern Recognition*, Quebec City, Canada, pp. 124-127, 2002.

[20] D. Upham, JPEFG-Jsteg, Retrieved July 2005, ftp://ftp.funet.fu/pub/crypt/steganography, 1993.

[21] A. Westfeld, F5 - A Steganogrpahy Algorithm: High Capacity Despite Better Steganalysis, *Information Hiding, 4th International Workshop. Lecture Notes in Computer Science*, eds. Moskowitz, I.S. Vol.2137. Springer-Verlag, Berlin Heidelberg New York (2001) 289-302

[22] Y. Wang and P. Moulin, Optimized Feature Extraction for Learning-Based Image Steganalysis, *IEEE Transactions on Information Forensics and Security*, 2(1), pp. 31-45, 2007

[23] G. Xuan, Y.Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen, Steganalysis Based on multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions, *information Hiding Workshop (IHW05)*, Barni, Joancomarti, Katzenbeisser, and Perez-Gonzalez, eds., pp. 262-278, 2005.