Chapter 27

FUSION OF STEGANALYSIS SYSTEMS USING BAYESIAN MODEL AVERAGING

Benjamin Rodriguez, Gilbert Peterson and Kenneth Bauer

Abstract The increasing use of steganography requires digital forensic examiners to consider the extraction of hidden information from digital images encountered during investigations. The first step in extraction is to identify the embedding method. Several steganalysis systems have been developed for this purpose, but each system only identifies a subset of the available embedding methods and with varying degrees of accuracy. This paper applies Bayesian model averaging to fuse multiple steganalysis systems and identify the embedding used to create a stego JPEG image. Experimental results indicate that the steganalysis fusion system has an accuracy of 90% compared with 80% accuracy for the individual steganalysis systems.

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. However, it is not possible to extract the hidden information without first identifying the method used to create the steganographic image. With more 250 steganography tools available on the Internet it is important to develop multi-class steganalysis systems that can label a suspect image as containing a specific type of steganography.

Several steganography detection systems are available, including research prototypes [4, 9, 11, 14, 18, 21] and commercially-available tools (e.g., ILook Investigator, Inforenz Forager, SecureStego, StegDetect [12] and WetStone Stego Suite). Each system has its own advantages and disadvantages. But with so many detection systems available to the steganalyst, a problem arises in deciding which system is best to use. A

Please use the following format when citing this chapter:

Rodriguez, B., Peterson, G. and Bauer, K., 2008, in IFIP International Federation for Information Processing, Volume 285; *Advances in Digital Forensics IV*; Indrajit Ray, Sujeet Shenoi; (Boston: Springer), pp. 345–355.

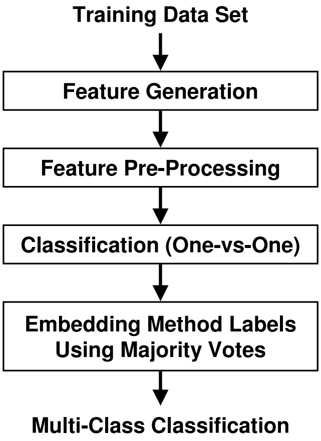
Training Data Set
Feature Generation
Feature Pre-Processing
Classification (One-vs-One)
Embedding Method Labels Using Majority Votes
Multi-Class Classification

Figure 1. Multi-class detection system.

solution to this problem is to fuse the results from the various detection systems to more accurately identify the embedding method.

This paper focuses on the detection of six steganography methods: F5 [22], JP Hide and Seek [8], JSteg [20], Model Based [16], OutGuess [13] and StegHide [5]. Bayesian model averaging [6] is used to combine four multi-class steganalysis detection systems. The first steganalysis system is StegDetect [12], which is capable of detecting F5, JP Hide and Seek, JSteg and OutGuess. The remaining three systems are one-vs-one multi-class classifiers [4, 9, 15] that use a two-class support vector machine (SVM) for classification. Test results show that the steganalysis fusion system has an accuracy of 90% compared with 80% accuracy for the individual multi-class steganalysis systems.

2. Related Work

Commercially available steganography detection tools are designed to give the analyst an initial indication if a set of images contains hidden information. These tools include ILook Investigator, Inforenz Forager, SecureStego, StegDetect [12] and WetStone Stego Suite. However, no tool targets all the common embedding methods. For example, StegDetect detects four (F5, JP Hide and Seek, JSteg and OutGuess) of the six common embedding methods.

The steps involved in multi-class steganalysis detection are illustrated in Figure 1. A data set containing clean and stego images is used to train a multi-class detection system.

The first step involves the generation of features from the input images; feature generation significantly reduces the amount of information sent to the classifier. The feature generation techniques used in our work are the wavelet-based method of Lyu and Farid [9], a DCT-based feature generation method [11], and a method that generates features from DCT decomposed coefficients [15].

The next step, feature pre-processing, employs two procedures. The first procedure normalizes the set of input features; this reduces the likelihood that features with large values would have a greater influence on the cost function than features with small values. The second procedure eliminates the less important features while retaining satisfactory class discrimination capability.

Many multi-class classifiers for steganalysis [11, 14] use a two-class SVM classification method in conjunction with a one-vs-one approach to combine individual classifiers. Multiple SVM classifiers are trained to distinguish clean images and images created with specific embedding methods. The overall multi-class classification system counts the votes from each SVM classifier; the final classification (identification of the embedding method used) is determined as the classification with the most number of votes.

The next section describes our steganalysis fusion system. It incorporates four systems discussed in this section: StegDetect [12], wavelet feature generation [9], DCT-based feature generation [11], and DCT decomposition feature generation [15].

3. Steganalysis Fusion System

The fused multi-class steganalysis detection system uses multi-class classifiers with Bayesian model averaging. The steganography techniques targeted by the detection system include F5 [22], JP Hide and Seek [8], JSteg [20], Model Based [16], OutGuess [13] and StegHide[5]. All these embedding methods hide data by manipulating the quantized discrete cosine transform (DCT) values generated during the JPEG image compression process. This section provides details of the feature generation, classification and labeling steps involved in multi-class detection (Figure 1).

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 to create a multi-class steganalysis classification system.

Wavelet feature generation first performs a multi-scale Haar wavelet decomposition of an image [9]. Next, higher-order statistics are calculated over each pixel in the wavelet and the pixel's relationship to its neighbors in the current and higher scales. 36 coefficient statistics and 36 error statistics are computed to yield a total 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 using several functions applied to the stego JPEG image. These functions include the global DCT coefficient histogram, co-occurrence matrix, spatial blockiness and others [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. An approximation of the hidden information is generated by applying the same functions to the cropped image. This feature generation technique produces 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 in the vertical, diagonal and horizontal directions. This is referred to as DCT decomposition. In addition, the coefficients are categorized into raw, shifted and predicted coefficients. The shifted coefficients are used to identify embedding blockiness between neighboring 8×8 blocks. The predicted coefficients estimate the coefficients altered by an embedding method. The features are generated by calculating several higher-order statistics (first, second, third and fourth moments; second, third and fourth central moments; and entropy) for 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 variety of application domains [1, 17]. In particular, the SVM produces a model that predicts the class of data instances in a testing set given only the attributes. 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. This is accomplished by performing a nonlinear separation of the input space using a nonlinear transformation $\phi(\cdot)$ that maps data instances x (with features x_i) from the input space into a higher-dimensional space called kernel space. The mapping, $\phi(\cdot) \rightarrow \phi(x_i)$, is performed by the SVM classifier using a kernel function $K(\cdot,\cdot)$. The SVM decision function is linear in the kernel space, albeit not in the feature space. We use LibSVM [2] in our work. This implementation employs sequential minimal optimization for a binary SVM with an L1-soft margin [3].

3.3 One-vs-One Methodology

Two-class classifiers are combined using a one-vs-one methodology [19]. This technique 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. The method is fairly reliable when the feature space is separable for the various classes.

Seven classes (6 stego + 1 clean, i.e., $k = 7$) are targeted by the steganalysis fusion system; this requires 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

Multi-class detection requires a training set for which the number of classes have been assigned. In our work, we attempt to detect stego images created using six embedding methods (F5, JP Hide and Seek, JSteg, Model Based, OutGuess and StegHide). Consequently, the training set consisted of seven classes of images (6 stego and 1 clean). Multi-class detection based on the training set involves the following steps:

1. Feature Generation: This step generates features from each JPEG test image. Three feature generation methods [9, 11, 15] are used to develop three distinct multi-class systems.

2. Feature Pre-Processing: This step normalizes the feature values and selects a subset of features based on the Fisher's discriminant ratio ranking. Other pre-processing methods could be applied for outlier removal, data normalization, feature selection and feature extraction [7].

3. Classification: This step uses an SVM to train each one-vs-one classifier based on the training data set.

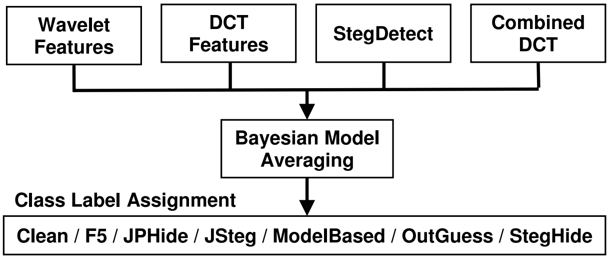

Figure 2. Bayesian model averaging structure.

4. Majority Vote Assignment: This step assigns a class label based on a majority vote from each classifier.

3.5 Bayesian Model Averaging

Bayesian model averaging merges several multi-class classifiers by combining the probability density estimation of each classifier's classification accuracy as a mixture of Gaussians [6, 10]. The Bayes Net Toolbox for Mathlab [10] was used to perform the model averaging computations. The probability density estimation specifies the local conditional probability distribution (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 $p(M_k|T)$, which represents the probability that a classification model will classify a target instance T. For example, given a target image that contains data hidden using JP Hide and Seek, $p(M_k|T = \textit{JP Hide and Seek})$ 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 classifications for a multi-class classifier, provides the probability density estimation for each classifier.

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

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

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

Figure 2 illustrates an example Bayesian model averaging system. 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.

The following seven steps are involved in using Bayesian model averaging for steganalysis:

1. Generate features.

2. Select relevant features.

3. Create the classification model based on one-vs-one training.

4. Use the majority vote strategy to populate the confusion matrix containing actual and predicted classified values for the clean, F5, JP Hide and Seek, JSteg, Model Based, OutGuess and StegHide training sets.

5. Repeat Steps 1 through 4 for each of the three 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 technique.

This seven-step procedure produces a multi-class model that receives four inputs (three from each of the trained detection systems and one from StegDetect) in order to classify a suspect image. The resulting steganalysis fusion system is shown in Figure 2.

4. Results

The results presented in this section are based on a data set containing 1,000 512×512 RGB JPEG (stego and clean) images. The training set consisted of 200 clean images and 100 images for each of the six embedding methods (F5, JP Hide and Seek, JSteg, Model Based, OutGuess and StegHide). The test set contained 50 clean images and 25 images for each embedding method. The clean images in the test set did not overlap with the stego images, 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. Approximately one page of text (4,000 characters) was hidden in each stego image.

The following were the percentages of altered coefficients for the six embedding methods:

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 (JP)	22.2 ± 0.5	21.8 ± 0.8	17.4 ± 1.1	20.0 ± 1.0
JSteg (JS)	20.8 ± 0.8	22.0 ± 1.6	20.0 ± 2.1	22.8 ± 0.8
Model Based (MB)	13.2 ± 1.3	16.4 ± 0.5	0.0 ± 0.0	17.8 ± 0.5
Outguess (OG)	17.0 ± 0.7	13.8 ± 0.5	17.4 ± 2.1	18.4 ± 0.5
StegHide (SH)	17.6 ± 1.1	16.4 ± 0.5	0.0 ± 0.0	18.0 ± 0.7

- F5 had an average of 0.3% of the coefficients altered.
- JP Hide and Seek had an average of 2.8% of the coefficients altered.
- JSteg had an average of 6.7% of the coefficients altered.
- Model Based had an average of 7.8% of the coefficients altered.
- OutGuess and StegHide had an average of 1% of the coefficients altered.

The testing was performed using five-fold cross validation. Note that the results are not intended to benchmark one system against the others. Rather, they are used to show that the steganalysis fusion system takes advantage of the strengths of the individual systems and improves the overall accuracy.

Table 1 presents the results for the individual steganalysis systems. The results reveal that no multi-class classification algorithm outperforms the others. For example, StegDetect detects all the F5 images; wavelet feature generation (Wavelet) labels the fewest clean images as stego; DCT based feature generation (DCT) identifies the largest number of JP Hide and Seek images; and DCT decomposition feature generation (Combined DCT) identifies the most Model Based and OutGuess images.

Table 2 presents the results obtained for the steganalysis fusion system. It is clear that the fusion system consistently outperforms the individual systems. The only exception is for the F5 embedding, where the fusion system and StegDetect detect all the images.

5. Conclusions

The steganalysis fusion system uses Bayesian model averaging to combine three multi-class SVM classifiers, each of which uses a different

Table 2. Confusion matrix obtained by Bayesian model averaging.

Actual		**Predicted**						
		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

feature extraction method. This strategy improves the overall accuracy with which steganography embedding algorithms are identified.

Future research will involve the addition of new steganalysis systems to the fused multi-class system, and the creation of richer JPEG data sets with images of various sizes and compression ratios. This work, which will utilize embedding signatures based on image size and compression changes, will further enhance the detection and identification of steganography embedding methods.

Acknowledgements

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

References

[1] C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, vol. 2(2), pp. 121–167, 1998.

[2] C. Chang and C. Lin, LIBSVM: A Library for Support Vector Machines (www.csie.ntu.edu.tw/∼cjlin/libsvm).

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

[4] J. Fridrich, Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes, *Proceedings of the Sixth International Information Hiding Workshop*, pp. 67–81, 2004.

[5] S. Hetzl, StegHide (steghide.sourceforge.net).

[6] J. Hoeting, D. Madigan, A. Raftery and C. Volinsky, Bayesian model averaging: A tutorial (with discussion), *Statistical Science*, vol. 14(4), pp. 382–417, 1999.

[7] A. Jain, R. Duin and J. Mau, Statistical pattern recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(1), pp. 4–37, 2000.

[8] A. Latham, Steganography – JP Hide and Seek (linux01.gwdg.de /∼alatham/stego.html).

[9] S. Lyu and H. Farid, Steganalysis using color wavelet statistics and one-class support vector machines, *Proceedings of the SPIE Electronic Imaging Symposium*, 2004.

[10] K. Murphy, Bayes Net Toolbox for Matlab (www.cs.ubc.ca/∼mur phyk/Software/BNT/bnt.html), 2007.

[11] T. Pevny and J. Fridrich, Merging Markov and DCT features for multi-class JPEG steganalysis, *Proceedings of the SPIE Electronic Imaging Symposium*, 2007.

[12] N. Provos, OutGuess (www.outguess.org).

[13] N. Provos and P. Honeyman, Hide and seek: An introduction to steganography, *IEEE Security & Privacy*, vol. 1(3), pp. 32–44, 2003.

[14] B. Rodriguez and G. Peterson, Steganography detection using multi-class classification, in *Advances in Digital Forensics III*, P. Craiger and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 193–204, 2007.

[15] B. Rodriguez, G. Peterson and R. Neher, DCT combined directional and frequency band distance measure features, submitted to *IEEE Transactions on Information Forensics and Security.*

[16] P. Sallee, Model-based steganography, *Proceedings of the Second International Workshop on Digital Watermarking*, pp. 154–167, 2003.

[17] B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, Massachusetts, 2001.

[18] Y. 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, prediction-error images and neural networks, *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 269–272, 2005.

[19] D. Tax and R. Duin, Using two-class classifiers for multi-class classification, *Proceedings of the Sixteenth International Conference on Pattern Recognition*, pp. 124–127, 2002.

[20] D. Upham, JPEG-JSteg (ftp.funet.fu/pub/crypt/steganography).

[21] Y. Wang and P. Moulin, Optimized feature extraction for learning-based image steganalysis, *IEEE Transactions on Information Forensics and Security*, vol. 2(1), pp. 31–45, 2007.

[22] A. Westfeld, F5 – A steganographic algorithm, *Proceedings of the Fourth International Workshop on Information Hiding*, pp. 289–302, 2001.

[23] G. Xuan, Y. 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, *Proceedings of the Seventh International Workshop on Information Hiding*, pp. 262–277, 2005.