Improving Steganalysis by Fusion Techniques: A Case Study with Image Steganography

Mehdi Kharrazi[a], Husrev T. Sencar[b] Nasir Memon[b]

[a]Department of Electrical and Computer Engineering
[b]Department of Computer and Information Science
Polytechnic University, Brooklyn, NY 11201, USA

ABSTRACT

In the past few years, we have witnessed a number of powerful steganalysis technique proposed in the literature. These technique could be categorized as either specific or universal. Each category of techniques has a set of advantages and disadvantages. A steganalysis technique specific to a steganographic embedding technique would perform well when tested only on that method and might fail on all others. On the other hand, universal steganalysis methods perform less accurately overall but provide acceptable performance in many cases.

In practice, since the steganalyst will not be able to know what steganographic technique is used, it has to deploy a number of techniques on suspected stego objects. In such a setting the most important question that needs to be answered is: What should the steganalyst do when the decisions produced by different steganalysis techniques are in contradiction? In this work, we propose and investigate information fusion techniques, that combine a number of steganalysis techniques. We start by reviewing possible fusion techniques which are applicable to steganalysis. Then we illustrate, through a number of case studies, how one is able to obtain performance improvements as well as scalability by employing suitable fusion techniques.

Keywords: Steganography, steganalysis, fusion

1. INTRODUCTION

Steganography refers to the science of "invisible" communication. Unlike cryptography, where the goal is to secure communications from an eavesdropper, steganographic techniques strive to hide the very presence of the message itself from an observer. On the other hand, *steganalysis* techniques are used to detect the presence of hidden messages in an object. The reader is referred to[1] for a review of the field. Essentially there are two approaches to the problem of steganalysis, one is to come up with a steganalysis method specific to a particular steganographic technique. The other is developing universal techniques that are effective for all steganographic techniques.

Specific steganalysis attacks concentrate on image features which are directly modified by the embedding algorithm. For example, F5[2] embedding algorithm suffers from shrinkage, in which the number of zero DCT coefficients increases after the embedding operation. To exploit this, the specific attack proposed in,[3] examines the differences between the histogram of the stego image and it's estimated original. As another example, in the model based embedding technique[4] the crux of the embedding operation lies in fitting a parametric model to the DCT histograms and preserving those models after embedding. The weakness of this approach is that DCT histograms of the cover images do not follow the model precisely. The specific attack proposed in ,[5] looks at how well the image's DCT histograms match the fitted model for that image, which indicates whether the images at hand is carrying any hidden messages or not. Although a steganalysis technique specific to an embedding method would give very good results when tested only on that embedding method, it might fail on all other steganographic methods.

Universal steganalysis techniques work by designing a classifier based on a training set of cover objects and stego objects obtained from a variety of different embedding algorithms. Classification is done based on some inherent "features" of typical cover images which can get modified when an image undergoes embedding process. There have been a number of universal steganalysis techniques proposed in the literature, these techniques differ in the feature sets they propose for capturing the natural image statistics. For example, Avcibas et al.[6] calculate

several binary similarity measures between the seventh and eighth bit planes of an image. Farid et al.,[7,8] obtain a number of statistics from the wavelet decomposed images. On the other hand, Fridrich,[9] utilizes at statistics of DCT coefficients as well as spatial domain statistics. As observed in[6,10] universal steganalysis technique do not perform equally over all embedding techniques. Nor are they able to distinguish perfectly between cover and stego images. In addition, it should be noted that training steganalyzers is a computationally expensive task.

With the availability of different steganalyzers (specific and universal) a number of questions would arise:

- What is performance penalty due to use of a universal (or a specific) steganalysis method in the practical setting of the problem?
- When multiple steganalyzers are used together, how do we deal with contradictory decisions?
- How does detection performance change when all embedding techniques are deployed in training the steganalyzer, and what is the computational cost for repeating the training process (to include new steganographic methods)?
- What is the most efficient strategy to *combine* different steganalyzers?

The questions above all point to one problem, the availability of many different technique, each making mistakes independently of the rest. As a solution to this problem, we investigate how steganalyzers, specific or universal, could be incorporated together with the help of *information fusion* techniques. In what follows, we first review fusion techniques which are applicable to our work in 2. In section 3, we study how the steganalyst can create a new steganalyzer by fusing a number of steganalysis techniques and whether the resulting performance results are superior to the performances of techniques being fused. In section 4, we study how incorporation of a number of steganalysis techniques could be made scalable with the help of fusion. We conclude in section 5 by discussing the obtained results.

2. FUSION TECHNIQUES

At the heart of every steganalyzer is a general classifier, which given an image feature value, in the case of specific steganalyzers, or an image feature vector, in case of universal steganalyzers, decides whether the image at hand contains any secret messages. Jain et al.[11] provide a break up of potential stages in the classification process, at which fusion could be applied. Below we will go over these stages, and discuss how they could be applied to steganalysis techniques.

2.1. Pre-classification

In essence, given an image I, the steganalyst first calculates the feature vector $X_I = [x_1, x_2, x_3, ...]$ from the image. The feature vector is then used by a classifier, trained on previous observations of X, to output a decision as to the nature of the image I (i.e., cover or stego). Fusion at this stage could be done by concatenating the feature vectors obtained for each steganalysis techniques, and re-training of the classifier with feature vector Y_I, which is equal to:

$$Y_I = [X_{I1}|X_{I2}|X_{I3}...] \tag{1}$$

Fusion at this stage would be best in an information theoretical sense, since the features are incorporated without any processing. But in practice a number of problems arise with such an approach. These are:

- With large set of features one should be careful with the curse of dimensionality.
- Correlated and redundant features need to be excluded.
- Steganalyzer needs to be re-designed every time a new component is added to the feature vector.
- Different feature sets could require different designing approaches.

To elaborate on the last point, in our experiments we have observed that some feature vectors gain much improvement with more computationally expensive non-linear classifiers, whereas others gain very little improvements. Thus one need to take into consideration such factors when designing the classifier.

2.2. Post-classification

The classifier is trained using a set of stego and cover feature vectors, calculating the location of the decision hyper-plane in the feature space. Therefore, the trained classifier could be thought of as a function, f_{class}. Given the feature vector of a test image, I, this function provides the perpendicular distance of I to the decision hyperplane in the feature space. This distance, also called decision value, is used to categorize the image I to either cover or stego. Therefore we have:

$$DecisionValue_{X_I} = f_{class}(X_I) = f_{class}(x_1, x_2, ..., x_n) \tag{2}$$

2.2.1. Measurement Level

The obtained decision values need to be normalized in order to make them comparable among a set of classifiers. This could done by converting the decision values to a conditional distribution, $P(stego|X_I)$, i.e., the posterior probability of image I represented by feature vector X_I carrying a secret message. Therefore we have:

$$P(stego|X_I) = f_{norm}(DecisionValue_{XI}) = f_{norm}(f_{class}(X_I)) \tag{3}$$

Since there are only two classes available (i.e. cover or stego), we have:

$$P(cover|X_I) = 1 - P(stego|X_I) \tag{4}$$

This is the most widely used stage for fusion. Here, the measurement info obtained from a set of steganalyzers could be either input into a second stage classifier for a final decision, or could be combined using schemes such as the sum rule, or max rule.

- *Sum Rule*

  $C = argmax_j \sum_{i=1}^{N} P(c_j|X_{Ii})$

  With this rule, the class c_j assigned to input image I, is the class with which the sum of the conditional probabilities for that class is maximized.

- *Max Rule*

  $C = argmax_j max_i P(c_j|X_{Ii})$

  Here the class c_j is assigned to input image I with which the maximum conditional probability is obtained.

A theoretical definition for the above mentioned rules is provided by Kittler et al. in ,[12] where the authors show that the sum rule is least susceptible to estimation errors in the conditional probability distributions.

2.2.2. Abstract Level

Fusion could also be applied at the last stage of classification, in which conditional class distributions are thresholded (or alternatively the decision values are thresholded directly), and a decision is made as to the class of the image I:

$$P(stego|X_I) > .5 \Rightarrow I \in stego \tag{5}$$

$$P(stego|X_I) < .5 \Rightarrow I \in cover \tag{6}$$

In this case, technique such as voting could be used to obtain a collective decision from a set of steganalyzers. But since this stage is obtained by thresholding the conditional probability distribution values, yielding a binary value, it will provide the minimal usable information for fusion.

Other that the classification stages we discussed, fusion could be applied in a number alternate scenarios as well. For example, among a set of classifiers all designed using one feature vector but with different settings. (I.e., given a feature vector, we could design a linear as well as non-linear classifier and fuse the results together.) On the other hand, fusion could also be applied to a set of classifiers each designed with a separate feature vector. Furthermore the two approaches could be combined into a hybrid approach.

3. FUSION BASED STEGANALYSIS

In a practical setting, the steganalyst will be unsure of the embedding technique being used, if any. Therefore the best approach would be for her to employ a universal steganalyser which could detect, although not perfectly, stego images. But the steganalyst could also have a set of specific steganalyzers at her disposal, which perform more accurately than the universal technique in some cases. In such scenario, the steganalyst could create a new steganalyzer and improve her detection performance by fusing the decision obtained from a set of universal and specific steganalysis techniques.

For example, take the scenario in which the embedder uses two types of embedding techniques, LSB and LSB +/-. The steganalyst has available at her disposal the BSM universal steganalyzer ,[6] which performs well over both the LSB and LSB +/- embeddings, and the Pair steganalysis ,[13] which is a specific attack on the LSB embedding technique. In what follows we illustrate through experimentation, how a new steganalyzer could be built by fusing the results from the BSM and Pair steganalysis techniques and obtain superior performance to either of the two technique when distinguishing between stego and cover images.

From the large data set of gray scale images obtained for our benchmarking study in,[10] we obtained about 13000 images with a quality factor of 85 and above and with a minimum width of 1000 pixels. These images were then down-sized to a size of 640x480 and saved as BMP. This was done to minimize the JPEG compression artifacts. We should note that the data set only consists of images that had their aspect ratio preserved after the down sampling operations. Two BSM steganalyzers were trained independently, using the non-linear SVM ,[14] with the LSB and LSB +/- stego images. Furthermore, in order to obtain the fusion result at different false positive rates (i.e. ROC curve), we opted to choose the output of the pair analysis attack as a feature vector in the classifier. This would also allow us to obtain classification confidence values which we will use when fusing the steganalysis techniques. Thus we have four steganalyzers available, including the fused steganalyzer.

The 4 steganalyzers are tested against the cover dataset and 3 different stego datasets, an LSB dataset, an LSB +/- dataset, and a dataset consisting of equal number of unique LSB and LSB +/- stego images. The obtained ROC curves for each dataset could be seen in figure 1, and the AUR (area under ROC curves) could be seen in table 1. From the results we observe that the specific attack works perfectly and has an accuracy of 100% when distinguishing between cover and LSB stego images, and when tested against the LSB +/- dataset the technique fails as expected. But when the pair analysis is tested against the mixed dataset, it has an average performance since it only detects the LSB images and not the LSB +/- images.

The two BSM steganalzyers, one trained with cover and LSB stego images the other with cover and LSB +/- stego images, perform around the same level with all three datasets. But when the output of these 3 steganalyzers, tested against the mixed dataset, is fused, we observe a 15.3%, 5.55%, and 5.17% performance improvement from the results obtained if we had used only the specific attack, BSM trained with LSB stego images, and BSM trained with LSB +/- stego images, respectively.

Table 1. AUR obtained from the ROC curves, when fusing universal and specific steganalysis techniques.

	Specific Attack	BSM (LSB)	BSM(LSB +/-)	Fusion
LSB (60%)	99.96	93.72	88.04	99.06
LSB +/- (60%)	53.84	79.49	85.72	84.15
Mixed Set	76.94	86.52	86.90	92.07

It should be noted that we also tried a decision tree approach in which at the root we had placed the pair steganalysis technique. But the results of such fusion technique were poor, due to the inaccuracy of the pair steganalysis technique in identifying LSB +/- stego images.

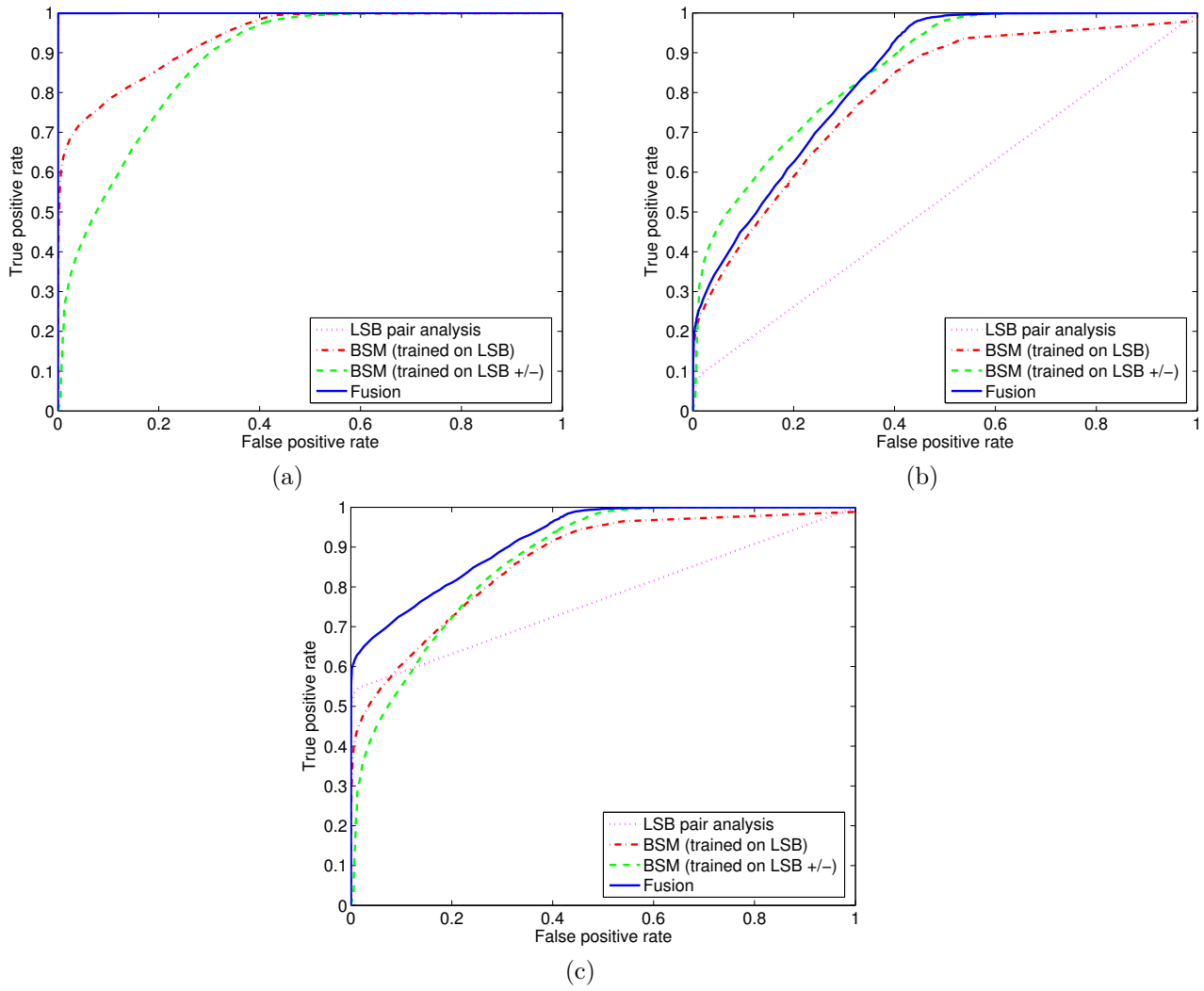

Figure 1. Obtained ROC curves for the 4 steganalyzers when employed against the same cover but different stego datasets. (a) The stego test images consist of only LSB images. (b) The stego test images consist of only LSB +/- images. (c) The stego test images consist of both LSB and LSB +/- images. Here the number of LSB and LSB +/- stego images is the same, and we have avoided using the same image from both sets.

4. FUSION BASED ADAPTIVE STEGANALYSIS

Although in theory universal steganalysis techniques are meant to detect any stego embedding technique, even ones unseen to it at the training stage, in our experiments (as will be discussed later in this section) we have observed otherwise. That is, a trained steganalyzer using embedding technique A, which also performs well when tested on stego images of type A, performs quite inaccurately if it is asked to classify stego image obtained from embedding technique B. This is best illustrated in figure 2, were we show two stego sets denoted as *stego1* and *stego2*.

If the training dataset only consists of *cover* and *stego1* images then the classifiers will have a decision plane following line A, with which most of *stego2* images will be classified correctly. But if the training dataset consists of *cover* and *stego2* images then the classifiers decision plane will follow line B, with which half of the the *stego1* images will be misclassified as cover images. In order to avoid such a scenario, the training set needs to include both *stego1* and *stego2* images with which the classifiers decision plane will follow line C, and will correctly classify both *stego1* and *stego2* images.

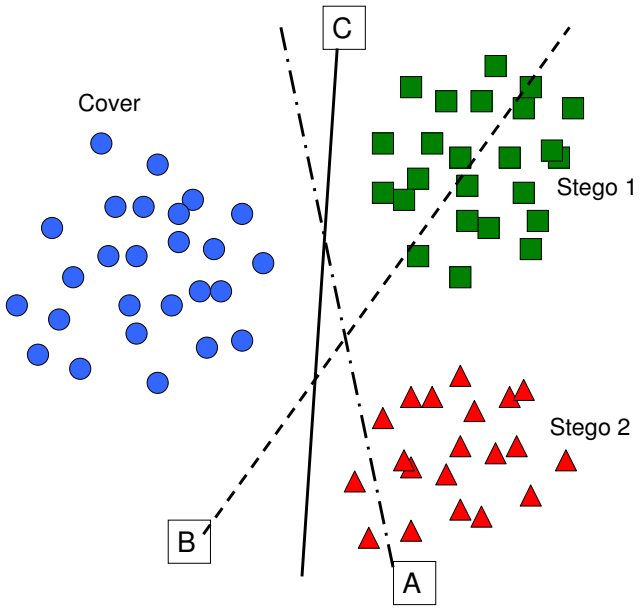

Figure 2. Effects of training set on the performance of universal steganalysis techniques.

We should note that the problem could potentially be avoided using one-class SVMs, but that approach has it's own downsides. One-class SVMs are designed by creating a hyper-sphere in the feature space. In the context of our discussion, the design parameters are chosen so that all (or most), cover images will fall inside the hyper-sphere. Any image which fall outside of the hyper-sphere is categorized as non-cover, or stego image. Thus as the name suggests, one-class SVM are designed with only one class of images (i.e. cover images). Therefore the accuracy of such classifiers greatly depends on how well the cover images, represented by a set of extracted features, could be enclosed in a hyper-sphere. That is why the binary SVM classifiers, which have access to both cover and stego images at the design stage, outperform the one-class SVM.

As the number of stego technique represented in the training dataset increases the size of the training dataset needs to grow as well. This is so that a minimal number of stego images from each technique could be represented in the dataset. The result is an increase in the training dataset size, which increases the classifier's training cost, thus making such approach unscalable. In fact, as observed in[10] the classifier training time versus training dataset size follows a polynomial curve. With fusion we could make this process scalable by designing a separate classifier for each available stego technique and then fusing the results obtained by testing an image against all available classifiers. But the question will be whether, with fusion, we will be able to obtain accuracy results as well as those obtained from a steganalyzer trained with a training set containing all available stego techniques.

To investigate the above question, we obtained a set of cover images, as in section 3, and used 3 embedding techniques, Outguess +, F5, and model based technique to create 3 stego datasets. The message lengths used are respectively .04, .06, and .08 bits per image pixel. As for the steganalyzer, we employed the FBS technique. The message lengths were chosen so that the steganalyzer when tested against each of the three stego sets have performance in the same range. A steganalyzer was trained independently (i.e., only using cover and stego images from that technique), we further designed a steganalyzer using a training set which consists of cover images and a stego dataset compromised of equal number of stego images from all 3 embedding techniques.

To show the importance of the training set on the performance of the universal steganalyzers, we tested each trained steganalyzer against the same cover but three different stego datasets. The obtained ROC curves could be seen in figure 3, and the calculated AUR is presented in table 2. We observe from these results that the steganalyzer trained solely on the Outguess + stego images, when asked to distinguish between cover and Outguess + images it obtains an accuracy of 94.90%. Similarly, its accuracy for distinguishing cover images from F5 and model based images is 77.61% and 51%, respectively. On the other hand the steganalyzer trained using all 3 available stego images, performs on all 3 datasets with accuracy values ranging from 85% to 90%.

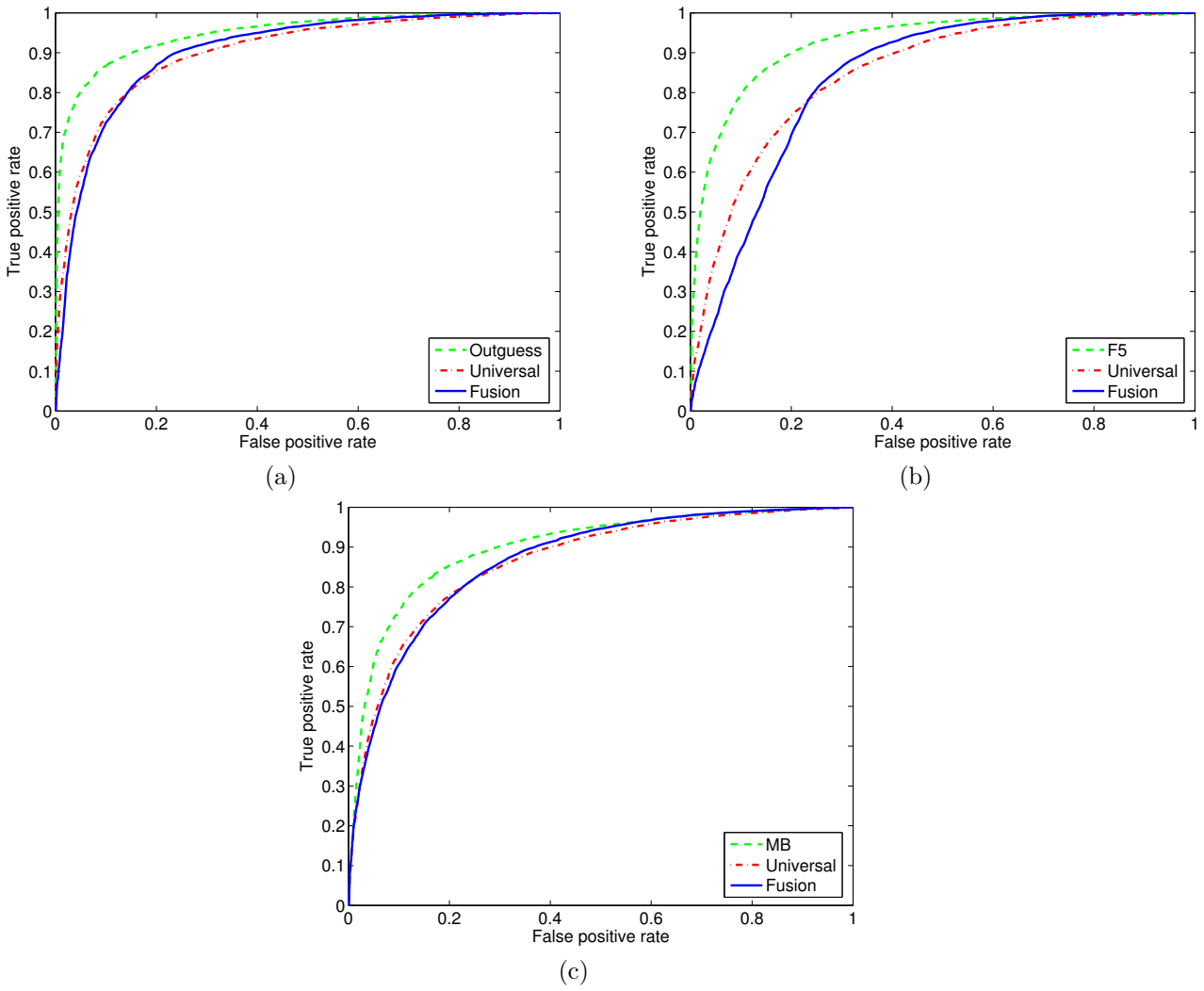

Figure 3. Obtained ROC curves for the designed steganalyzers when employed against the same cover but different stego datasets. (a) Tested against the Outguess stego set. (b) Tested against the F5 stego set. (c) Tested against the MB stego set.

Using the *sum* rule to fuse the output of the 3 steganalyzers, each trained for one of the 3 embedding techniques, we test it on the 3 datasets. As evident from the results the fused steganalyzer has performance values very close to the steganalyzer trained with all the three techniques. Thus, fusion allows us to train steganalyzers only using one embedding technique and then fuse the outputs together which is a scalable solution.

Given the results, we make two observations, first of all, as argued earlier the composition of training set plays an important role on the performance of the steganalyzer. Secondly, we see a performance degradation ranging roughly from 3% to 7% when we test our datasets against the steganalyzer trained with all three stego techniques.

5. DISCUSSION

With the availability of large number of steganalysis techniques proposed in the literature, one might feel that the steganalyst has a good chance of distinguishing between cover and stego images. But in practice, the steganalyst will have to select one or more techniques which she will employ on a set of suspected stego objects. However, the question of what should she do when the results produced by the techniques are in contradiction was not

Table 2. AUR obtained from the ROC curves, when fusing steganalyzers to obtain scalability.

	FBS (Out+)	FBS (F5)	FBS (MB)	FBS (Universal)	FBS (Fusion)
Out+	94.90	44.59	83.55	90.20	90.42
F5	77.61	92.81	63.44	85.08	83.73
MB	51.00	65.70	89.93	86.38	86.60

answered. In this work, we investigated how fusion techniques could be applied in steganalysis to resolve such questions. As the first application, we illustrated how a new steganalyzer could be create by fusing a number of steganalysis techniques, at the same time improving detection accuracy.

As a second application of fusion, we discussed the importance of the training set for universal steganalysis techniques and argued that incorporation of the new embedding technique into the an already designed steganalyzer is a costly and unscalable procedure. As an alternative we proposed fusing decisions from a set of steganalysis technique trained independently using only one embedding technique. We illustrated through experimentation that the obtained accuracy results matches that of training steganalyzers individually using stego images from a number of embedding techniques while at the same time providing a scalable solution to the problem.

We believe that the applications of fusion techniques are not limited to the examples we have studied in this work. For example, although the goal of steganalysis is the detection of stego objects, fusion techniques could also be employed for post-steganalysis operations which might help determine the specific embedding technique used or the length of the embedded message. This form of information would be quite valuable in any forensic analysis of the stego-object that intends to recover the secret message.

REFERENCES

1. M. Kharrazi, H. T. Sencar, and N. Memon, "Image steganography: Concepts and practice," *to appear in Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore* , 2004.
2. A. Westfeld, "F5 steganographic algorithm: High capacity despite better steganalysis," *4th International Workshop on Information Hiding.* , 2001.
3. J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitive steganalysis of digital images: Estimating the secret message lenght," *ACM Multimedia Systems Journal, Special issue on Multimedia Security* , 2003.
4. P. Sallee, "Model-based steganography," *International Workshop on Digital Watermarking, Seoul, Korea.* , 2003.
5. R. Bhme and A. Westfeld, "Breaking cauchy model-based jpeg steganography with first order statistics," *9th European Symposium on Research in Computer Security, Sophia Antipolis, France, September* , 2004.
6. I. Avcibas, M. Kharrazi, N. Memon, and B. sankur, "Image steganalysis with binary similarity measures," *To appear in EURASIP Journal on Applied Signal Processing* , 2005.
7. S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," *5th International Workshop on Information Hiding.* , 2002.
8. 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.
9. J. Fridrich, "Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes," *Proc. 6th Information Hiding Workshop, Toronto, Canada, May 23-25* , 2004.
10. M. Kharrazi, T. H. Sencar, and N. Memon, *Benchmarking steganographic and steganalysis techniques.*
11. A. K. Jain, K. Nandakumar, and A. Ros, "Score normalization in multimodal biometric systems," *to appear in Pattern Recognition* , 2005.
12. J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3), pp. 226–239, 1998.
13. S. Dumitrescu, X. Wu, and N. Memon, "On steganalysis of random lsb embedding in continuous-tone images," *IEEE International Conference on Image Processing, Rochester, New York.* , September 2002.
14. C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.