# On the Performance of Wavelet Decomposition Steganalysis with JSteg Steganography

Ainuddin Wahid Abdul Wahab, Johann A Briffa and Hans Georg Schaathun

Department of Computing, University of Surrey

**Abstract.** In this paper, we study the wavelet decomposition based steganalysis technique due to Lyu and Farid. Specifically we focus on its performance with JSteg steganograpy. It has been claimed that the Lyu-Farid technique can defeat JSteg; we confirm this using different images for the training and test sets of the SVM classifier. We also show that the technique heavily depends on the characteristics of training and test set. This is a problem for real-world implementation since the image source cannot necessarily be determined. With a wide range of image sources, training the classifier becomes problematic. By focusing only on different camera makes we show that steganalysis performances significantly less effective for cover images from certain sources.

## 1 Introduction

Steganography allows a user to hide a secret message in such a way that an adversary cannot even detect the existence of the message. We are concerned with image steganography, where the secret message is represented by imperceptible modification in a cover image.

Over the last decade a wide variety of steganography techniques have appeared in the literature. In response, there have also been a wide variety of steganalysis techniques, intended to let an adversary determine whether an intercepted image contains a secret message or not. In particular, a number of techniques based on machine learning have emerged. Such techniques tend to be blind, in the sense that they do not assume any particular steganography algorithm and can usually break a variety of algorithms.

In this paper we consider a steganalysis technique due to Lyu and Farid [12]. This technique, claimed to break a variety of steganography systems including Jsteg and Outguess was published in [12].

The idea in machine learning is that the steganalysis algorithm, during a training phase, is given large sets of steganograms and natural images. These images, where the classification (steganogram or natural image) is

known are used to tune the classification parameters. When an unknown image is subsequently presented to the system, it is classified according to these pre-tuned parameters.

Typically a set of professional photographs from an online database is used. However, if the algorithm is later used to analyze images from a different source, the training data may not be relevant. In particular, sensor noise from cheap cameras may have characteristics similar to an embedded message. The same could be the case for images with low quality due to poor lighting or photography skills.

In this paper we confirm this hypothesis by simulating embedding and analysis of images from different sources. We conclude that further research is needed to make the Lyu-Farid algorithm reliable for real-world applications where the image source cannot necessarily be determined. This is due to the fact that the Lyu-Farid algorithm requires detailed information about cover source.

## 2 The Lyu-Farid Algorithm

In this section we explain how to implement the Lyu-Farid algorithm. The algorithm uses a Support Vector Machine (SVM) classification. SVM does not operate on the images themselves. Instead a *feature vector* (i.e. a series of statistics) are extracted from the image to be used by the SVM.

The features used in the steganalysis are extracted from the wavelet domain, so we will first present the wavelet transform and then the exact features used. Subsequently we will introduce the SVM.

### 2.1 The Wavelet Transform

A wavelet is a waveform of limited duration with an average value of zero. One dimensional wavelet analysis decomposes a signal into basis functions which are shifted and scaled versions of a original wavelet [16]. Continuous Wavelet Transform (CWT) is usually used for time continuous signals while Discrete Wavelet Transform (DWT) is used when the signals is sampled, as in digital image processing.

Besides its usefulness in image processing, the decomposition also exhibits statistical regularities that can be exploited for certain purposes such as steganalysis. The decomposition technique used in our experiment is based on quadrature mirror filter (QMFs) [16]. The decomposition process splits the frequency space into multiple scales and orientations (vertical, horizontal, diagonal and lowpass subband). This is achieved by applying separable lowpass and highpass filter along image axes.

### 2.2 Wavelet Decomposition Steganalysis

Two set of statistics (basic coefficient distribution and errors in an optimal linear predictor of coefficient magnitude)are then collected. Those composed of mean, variance, skewness and kurtosis of the subband coefficient for different orientation and scales($s$). The total of 12($s$-1) for each set of statistics collected. From [13], $s$ value is four. Based on that, 72 individual statistics are generated. The collected statistics are then used as a feature vector to discriminate between clean (image without payload encoded into them) and stego (image with payload encoded into them) images using Support Vector Machine (SVM).

### 2.3 Support Vector Machine

SVM performs the classification by creating a hyperplane that separates the data into two categories in the most optimal way.

In our experiment, the SVM is employed to classify each image as either clean or stego, based on its feature vector. For our experiment, the feature vector are constructed from the wavelet decomposition process discussed previously where each image is represented with the 72 individual statistics. SVM has been shown to provide good results in [12]. Classification requires training and testing data sets. In the training set, each individual instance has a class label value and several attributes or features (feature vector). SVM will produce a model based on training set data, and then using that model to predict the class of testing set based only on their attributes.

### 2.4 SVM Kernel

Lyu and Farid in [12] have been using Radial Basis Function (RBF) kernel for SVM. RBF kernel function:

$$K(X_i, X_j) = \exp\left(-\gamma||X_i - X_j||^2\right), \gamma > 0$$

where $\gamma$ is a kernel parameter.

According to [8], there are four basic kernel in SVM namely Linear, Polynomial, RBF and Sigmoid. The first three basic functions has been tested with the input from this experiment for clarification on why RBF used as SVM kernel in [12]. The details of the image sets or input for this experiment are discussed in section 4. Basically, the training images is the combination of images from all three devices (Table 2) while the test set is from 'Canon' and 'Sony' set of images. Both the training and test set

are combination of clean and corresponding JSteg stego images (image with payload encoded using JSteg). From the result in Table 1, it can be seen that RBF really provide the best results for both set of test images. Besides that, the suggestion on how to select the best kernel in [8] also indicate that RBF is the best kernel to be used for wavelet decomposition type of data.

Table 1: SVM Kernel Test

| | Canon | | | Sony | | |
|---|---|---|---|---|---|---|
| SVM Kernel | Linear | Polynomial | RBF | Linear | Polynomial | RBF |
| False Negative | 2.0% | 2.0% | 5.0% | 7.0% | 19.0% | 6.0% |
| False Positive | 18.0% | 19% | 3.0% | 1.0% | 0% | 0% |
| Detection Rate | 80.0% | 79.0% | 92.0% | 92.2% | 81.0 % | 94.0% |

## 3 JSteg

JSteg [17] is a steganographic method for JPEG images that can be viewed as LSB steganography. It works by embedding message bits as the LSBs of the quantized DCT (Discrete Cosine Transform) coefficients. All coefficients values of '0' and '1' will be skipped during JSteg embedding process which can be perform either in sequential or random location. Quite a number of attacks have been used to defeat JSteg such as chi-square attack [1] and generalize chi-square attack [15]. Futhermore, Lyu and Farid in [12] have shown that their wavelet decomposition steganalysis technique is able to defeat JSteg.

## 4 Detection Experiment

To evaluate the performance of the steganalysis algorithm on JSteg, we use sample images captured using Canon digital camera, Nokia N70 mobile phone and Sony video camera. The details of devices used to capture image given in Table 2.

As in [5], cover grayscale images are used due to the fact that steganalysis are harder for grayscale images. All images were crop to the center, resulting in image sizes of 640x480 pixels and saved in JPEG image format with quality factor of 75. This can help to ensure that the image dimensions is not correlated with spatial characteristics, such as noise or local energy as what mentioned in [3].

The images keep at fixed size of 640x480 to ensure that the collected statistics will be at the same amount for each image since it has been found in [10] and [3] that detection performance is likely to suffer for smaller images.

Following the image preparation process, all the images went through the JSteg encoding process [17] to produce a set of stego images. The secret message is an image with size of 128x128. With the cover and stego images ready, the wavelet decomposition steganalysis technique conducted by using SVM [14] as classifier.

For SVM, the soft-margin parameter used with its default value of 0. The only parameter tuned to get the best result is the kernel's parameter, $\gamma$ where

$$\gamma \in \left\{2^i\right\}, i \in \{0, 1, 2, 3\}$$

The total number of images used for training and testing is 400 for each set of images from different camera. Besides the three main set of images (Canon, Nokia and Sony), there is another two sets of images (Combination200 and Combination600) which contains a combination of images from each type of camera. These two sets, 'Combination200' and 'Combination600', have a total number of 400 and 1200 images for training and testing accordingly.

For reference purpose, the sixth set of images included in our experiment. It is a set of images from Greenspun's database [7] which consist of a balanced combination of indoor and outdoor images. The total number of images for training and testing for this set is 400. This database is the source of images used by Lyu in [12].

Table 2: Device Specification

| Device | Model | Resolution | Additional Info |
|---|---|---|---|
| Sony | DCR-SR42 | 1 MP | Digital video camera |
| Nokia | N70 | 2MP | Build in phone camera |
| Canon | Powershot A550 | 7.1MP | Canon Digital Camera |

## 5 Discussion

Classification accuracy used in [4] and [12] to measure the performance of their proposed method while in [6] the performance were evaluated using

Table 3: False Negative | False Positive (False Alarm)

| Test Set | Training Set | | | | | |
|---|---|---|---|---|---|---|
| | Canon | Nokia | Sony | Comb.(200) | Comb.(600) | Greenspun |
| Canon | 7.6% \| 6.7% | 28.0% \| 2.7% | 1.6% \| 60.3% | 2.0% \| 9.0% | 1.6% \| 6.3% | 31.0% \| 2.0% |
| Nokia | 42.3% \| 3.0% | 20.3% \| 4.6% | 15.6% \| 36.6% | 7.0% \| 19.0% | 7.3% \| 11.0% | 61.0% \| 3.0% |
| Sony | 64.0% \| 6.7% | 53.6% \| 3.3% | 3.3% \| 4.6% | 13.0% \| 3.6% | 8.6% \| 1.3% | 71.0% \| 1.0% |
| Combination(200) | 40.3% \| 1.0% | 29.6% \| 4.3% | 8.0% \| 30.6% | 8.0% \| 14.3% | 9.0% \| 7.6% | 62.0% \| 0% |
| Combination(600) | 38.0% \| 0.7% | 26.3% \| 4.5% | 7.3% \| 31.3% | 7.0% \| 13.7% | 8.6% \| 3.0% | 50.3% \| 2.3% |
| Greenspun | 66.0% \| 9.0% | 55.0% \| 30.0% | 6.0% \| 86.0% | 10.0% \| 64.0% | 10.0% \| 61.0% | 27.0% \| 7.0% |

Table 4: Precision

$$TruePositive / (TruePositive + FalsePositive)$$

| Test Set | Training Set | | | | | |
|---|---|---|---|---|---|---|
| | Canon | Nokia | Sony | Combination(200) | Combination(600) | Greenspun |
| Canon | 93.2% | 96.4% | 62.0% | 91.6% | 94.0% | 97.2% |
| Nokia | 95.0% | 94.5% | 69.8% | 83.0% | 89.4% | 92.9% |
| Sony | 84.3% | 93.4% | 95.5% | 96.0% | 98.6% | 96.7% |
| Combination(200) | 98.4% | 94.2% | 75.0% | 86.5% | 92.3% | 100.0% |
| Combination(600) | 98.9% | 94.2% | 74.8% | 87.2% | 96.8% | 95.6% |
| Greenspun | 87.7% | 74.0% | 54.4% | 67.6% | 67.8% | 91.3% |

Table 5: Detection Rate (Accuracy)

$$(TruePositive + TrueNegative) / (TotalPositive + TotalNegative)$$

| Test Set | Training Set | | | | | |
|---|---|---|---|---|---|---|
| | Canon | Nokia | Sony | Combination(200) | Combination(600) | Greenspun |
| Canon | 92.9% | 84.7% | 69.1% | 94.5% | 96.1% | 83.5% |
| Nokia | 77.4% | 87.6% | 73.9% | 87.0% | 90.9% | 68.0% |
| Sony | 64.7% | 71.2% | 96.1% | 91.7% | 95.1% | 64.0% |
| Combination(200) | 79.4% | 83.1% | 80.7% | 88.9% | 91.7% | 69.0% |
| Combination(600) | 80.7% | 84.6% | 80.7% | 89.7% | 94.2% | 73.7% |
| Greenspun | 78.0% | 76.0% | 58.0% | 74.5% | 75.5% | 83.0% |

'detection reliability' $\rho$ defined as

$$\rho = 2A - 1,$$

where $A$ is the area under the Receiver Operating Characteristics (ROC) curve, also called an accuracy.

For our experiment, the results in Tables 3, 4 and 5 showing the performance of SVM using false negative and false positive rate (Table 3) together with classification precision (Table 4) and classification accuracy (Table 5). Those results confirm existing claims that te Lyu-Farid technique can be used to defeat JSteg. The detection rate for using the same source of images for training and test sets match with claims in [12]. While showing the success of Lyu-Farid technique, these results also shows that the accuracy of the technique is seriously affected by the training set used.

By using confidence interval estimation technique [2], from the results, we have computed 95.4% confidence intervals for the false negative rate for Canon $(3.2\%, 10.2\%)$, Nokia $(14.6\%, 26.0\%)$, and Greenspun $(20.7\%, 33.3\%)$. Thus, we can confidently conclude that the steganalysis algorithm is significantly less effective for cover images from certain sources.

In [10] it has been found that JPEG image quality factor affects the steganalyzer performance where cover and stego images with high quality factors are less distinguishable than cover and stego image with lower quality. Furthermore, Böhme in [3] also found that images with noisy texture yield the least accurate stego detection. Related to those results, in our experiment, while having the same quality factor and using the same steganography technique, it has been shown that images captured using high resolution devices (Canon) are more distinguishable than cover and stego image from a low resolution device (Sony).

From [9], it has been observed that a trained steganalyzer using specific embedding technique performs well when tested on stego images from that embedding technique, but it performs quite inaccurately if it is asked to classify stego image obtained from another embedding technique. In their experiment, when steganalyzer trained solely on the Outguess (+) stego images, and asked to distinguish between cover and Outguess (+) images, it obtains an accuracy of 98.49%. But, its accuracy for distinguishing cover images from F5 and Model Based images is 54.37% and 66.45%, respectively.

Having the same pattern, in our experiment, by covering all types of image sources while training the SVM, the technique can be seen to have a good detection rate. However, the performance decrease when the test image type is not previously in its training set. The clearest example is when SVM trained with 'Sony' images and then tested with 'Nokia' and 'Sony'. With detection rate of 96.1% for the 'Sony' test images the detection rate are lower for 'Canon' and 'Nokia' images with rates of 69.1% and 73.9% accordingly.

The Lyu-Farid technique also seems not to perform well when trained with images from higher resolution camera and then tested with lower resolution camera. For example, it can be seen when the SVM trained using images from 'Canon' and tested with images from 'Nokia' and 'Sony'. While having an accuracy of 92.9% for 'Canon' test set, the accuracy decreased to 77.4% and 64.7% respectively with 'Nokia' and 'Sony' test sets.

The number of images in training set also plays an important role to ensure that the SVM are well trained. This can be seen clearly at the differences of accuracy rate when the SVM trained using images from 'Combination200' and 'Combination600'. With assumption that SVM is not well trained using smaller number of images ('Combination200' training set), the accuracy rate can be seen increased when the SVM trained with bigger number of images ('Combination600' training set).

The problem with the above situations is the practicality for the real-world implementation of the technique. There is a huge diversity of image sources in the world today. Kharrazi in [9] has demonstrated how the computational time increases rapidly as the training set size increases. In his experiment, by having training set consists of 110 000 images, it would take more than 11 hours to design or train the non-linear SVM classifier. In our case for example, if there is new source of image found or designed, then the SVM classifier has to be retrained. If we try to cover all possible image sources, we can imagine how long it would take to retrain the SVM classifier.

Also related to image source diversity, is the attempt to train the classifier with all possible type of images in the public domain. Some researchers are trying this using a million images in the training set [9].

## 6 Conclusion

In our experiments we investigated the performance of Lyu-Farid steganalysis on JSteg using cover images from different sources. Our exper-

iments show that performance claims previously made have to assume knowledge about the cover source. If the steganalyst is not able to train the SVM using cover images from a very similar source, significant error rates must be expected.

In the case of Jsteg, Lyu-Farid seems to get reasonable performance if a large and mixed training set is used (as in the Combined 600-set). A training set to enable reliable detection of any steganogram using any cover source would probably have to be enormous.

Even when we have been able to train on images from the correct source, we observe that some sources make steganalysis difficult. Images from the Nokia phone and from the Greenspun database have significantly higher false negative rates than other sources. An interesting open question is to identify optimal cover sources from the steganographer's point of view.

It may very well be possible to design steganographic algorithms whose statistical artifacts are insignificant compared to the artifacts of a particular camera. This is to some extent the case for SSIS [11] which aims to mimic sensor noise which is present (more or less) in any camera.

On the steganalytic case, quantitative results would be more interesting if real world implementation is considered. How large does the training set have to be to handle any source? Are there other important characteristics of the image which must be addressed, such as the ISO setting of the camera, lighting conditions and also indoor versus outdoor?

## References


1. A.Westfeld and A. Pfitzmann. Attacks on steganographic systems. *IHW 99*, 1999.
2. G. K. Bhattacharyya and R. A. Johnson. *Statistical Concepts and Methods*. Wiley, 1977.
3. R. Böhme. Assessment of steganalytic methods using multiple regression models. In *Information Hiding*, pages 278–295, 2005.
4. H. Farid. Detecting hidden messages using higher-order statistical models. In *International Conference on Image Processing*, Rochester, NY, 2002.
5. J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In *MM&Sec '07: Proceedings of the 9th workshop on Multimedia & security*, pages 3–14, New York, NY, USA, 2007. ACM.
6. J. J. Fridrich. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Information Hiding*, pages 67–81, 2004.
7. P. Greenspun. Philip greenspun's home page, 2008. `http://philip.greenspun.com/`.
8. C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, Taipei, March, 2008. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

9. M. Kharrazi, H. T. Sencar, and N. Memon. Improving steganalysis by fusion techniques: A case study with image steganography. *LNCS Transactions on Data Hiding and Multimedia Security I*, 4300:123–137, 2006.
10. M. Kharrazi, H. T. Sencar, and N. Memon. Performance study of common image steganography and steganalysis techniques. *Journal of Electronic Imaging*, 15(4), 2006.
11. C. G. B. L. M. Marvel and C. T. Retter. Spread spectrum image steganography. *IEEE Transactions On Image Processing*, 8(1):1075–1083, August 1999.
12. S. Lyu and H. Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.
13. S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.
14. W. S. Noble and P. Pavlidis. Gist:support vector machine. `http://svm.nbcr.net/cgi-bin/nph-SVMsubmit.cgi`.
15. N. Provos and P. Honeyman. Detecting steganographic content on the internet. *CITI Technical Report*, pages 01–11, 2001.
16. G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
17. D. Upham. Jpeg-jsteg-v4. `http://www.nic.funet.fi/pub/crypt/steganography`.