2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Steganalysis in the Presence of Watermarked Images

Pham Hai Dang Le, Daniel Graf, and Matthias O. Franz
Institute for Optical Systems (IOS)
HTWG Konstanz University of Applied Sciences
Brauneggerstraße 55, 78462 Konstanz, Germany
{*dangle, daniel.graf, mfranz*}*@htwg-konstanz.de*

***Abstract*—Watermarked images are increasingly prevalent in the internet. Hence, any practical steganalyzer has to take the presence of watermarked images into account, particularly as potential source of false alarms due to similar embedding algorithms. In this study, we investigate the impact of watermarked images on the performance of a standard steganalyzer using two recent watermarking schemes: *JPEG Compression Resistance Watermarking* (JCRW) in the DCT domain [1] and *Controllable Secure Watermarking* (CSW) [2] in the pixel domain. Our findings show that JCRW and CSW do interfere with steganalysis. In particular, CSW-watermarked images were mostly classified as stego images thus rendering the investigated steganalyzer useless because of the excessively high false alarm rate. We propose a two-step classifier to handle this problem which achieves the same performance as the original steganalyzer, but in the presence of watermarking.**

***Keywords*-Steganalysis, Steganography, Watermark, SVM**

I. Introduction

The rising interest in the protection of digital content has lead to an increased use of watermarking. As a consequence, any steganalyzer applied to large assorted image datasets (e.g. from the internet) is likely to encounter watermarked images. In forensic steganalysis, where one is only interested in detecting steganography, not in watermarks, this poses a severe problem: Since both watermarking and steganography are concerned with hiding a message in other information it is quite plausible that both techniques interfere with each other, e.g. certain watermarks are erroneously detected as steganograms, or steganographic messages are embedded in addition to the watermarks in order to further obscure them.

Note that especially the first point of watermarked images as false alarms can render a steganalyzer completely useless in spite of the fact that currently, watermarked images account for only a few percent of all images available online. Typically, watermarked images will still greatly outnumber the few – if at all – real steganograms in an assorted online image database. If a significant proportion of the watermarked images is erroneously detected as steganograms, the few real hits will be swamped in their sheer number. In a realistic, highly unbalanced steganalysis scenario the number of clean images will exceed that of steganograms by many orders of magnitudes which makes keeping the false alarm rate low a crucial feature of any practical steganalyzer. Watermarking is likely to interfere with this goal.

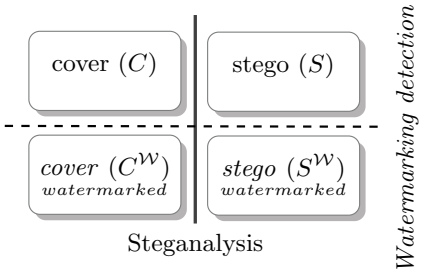

Figure 1: Steganalysis and watermark detection

We have to consider four image types that can arise in this context (Figure 1):

- non-watermarked cover images[1], denoted as C,
- non-watermarked stego images (S),
- watermarked cover images ($C^{\mathcal{W}}$),
- and watermarked stego images ($S^{\mathcal{W}}$).

The detection of both watermarks ($C^{\mathcal{W}}$ vs. C) and stego images (S vs. C) are well-studied fields (e.g., [3]–[5]). However, to our knowledge, the combined problem has not been studied in the literature so far. Here, we focus on *steganalysis in the presence of watermarking*, i. e., on the dichotomy $S \cup S^{\mathcal{W}}$ vs. $C \cup C^{\mathcal{W}}$. Our test scenario consists of the image databases *BOSSbase* [6], [7] and *BOWS-2* [8]. The images were represented in the formats JPEG and PNG. The JPEG images were watermarked in the DCT domain using *JPEG Compression Resistance Watermarking* (JCRW) [1], the PNG images in the pixel domain using *Controllable Secure Watermarking* (CSW) [2]. In the next step, we generated stego images for all four image types by LSB matching (also called ± 1 embedding) [9]. Our test steganalyzer is a standard Support Vector Machine (SVM, e.g. [10]) with *Subtractive Pixel Adjacency Matrix* (SPAM) features [11] which are known to give good results for LSB matching.

SVMs need a training set consisting of cover-stego image pairs to derive statistical regularities for predicting whether an unknown image is steganographically manipulated or not. We will show that such an SVM-based steganalyzer trained

[1]In the following, the terms *cover* and *stego image* always refer to the non-watermarked cases C and S, respectively.

978-0-7695-5120-3/13 $26.00 © 2013 IEEE
DOI 10.1109/IIH-MSP.2013.133

exclusively on non-watermarked images fails in the presence of certain types of watermarking. Here, a straightforward approach would be to augment the training set by additionally including watermarked cover-stego image pairs. We will see that this approach is indeed viable for steganalysis in the presence of watermarking, however at the price of a considerably reduced performance. It appears that watermarking and steganography mask each other in the training process. In order to handle watermarked images without decreasing performance, we propose a two-step steganalyzer with a preselection step that picks out the watermarked images before steganalysis. Our experiments show that this steganalyzer achieves the same performance in the presence of watermarking as the original test steganalyzer.

In the next section, we describe the various components of our test scenario in more detail. The experimental results are presented in Section III. Finally, we conclude our paper with a brief summary and discussion in Section IV.

II. Steganalyzer and watermarking technique

We start by introducing LSB matching as the steganographic algorithm, and CSW [2] and the JCRW [1] technique as the two applied watermarking techniques. In Subsection II-C, we describe our initial steganalyzer and extend the initial classifier by using a preselection step.

A. LSB matching

Assume that the image is denoted as a vector where the pixel is indexed by a single number, i. e. $\mathbf{x} = (x_k)_{k=1}^n \in \{0, ..., 255\}^n$. LSB matching modifies the pixel elements by incrementing and decrementing the pixel value. From this point of view, LSB matching is also called ± 1 embedding. If the message bit $m_i \in \{0, 1\}$ is contrary to the least significant bit of the image, i. e. $m_i \neq (\text{LSB}(x_i))$, LSB matching can be described by the following function:

$$Emb_{\pm 1}(x_i) = \begin{cases} x_i - 1, & \text{if} \quad x = 255, \\ x_i + 1, & \text{if} \quad x = 0, \\ x_i - 1, & \text{for} \quad \textbf{even} \text{ message bit number}, \\ x_i + 1, & \text{for} \quad \textbf{odd} \text{ message number}. \end{cases}$$

B. Watermarking techniques

The JCRW technique [1] is designed to be robust against JPEG compression. JCRW operates in the discrete cosine transformation (DCT) domain that is also used in the JPEG standard. The watermark is embedded in the DCT-coefficients. The process can be divided into two steps. The first step identifies the DCT-coefficients which are suitable to resist against high compression. To this end, JPEG compression is applied from a quality factor of 1% up to 100% to identify the likely DCT-coefficients for embedding. In the second step, the determined DCT-coefficients from the first step are selected again. Here, the method selects the DCT-coefficients with a minimal difference to the uncompressed DCT-coefficient.

The controllable secure watermarking (CSW) technique [2] controls the tradeoff between robustness and security. In CSW, the security means that the distribution of the host contents and the marked contents is the same, i. e. the $\mathrm{D}_{KL}(\mathbf{x}, \mathbf{y}) = 0$, where D_{KL} indicates the Kullback-Leibler divergence [12]. This security terminology is also used in the transportation natural watermarking (TNW) [12] which serves as basis for CSW. The CSW technique contains an additional matrix $\mathbf{V}$ compared to the TNW technique, which allows to alter the host signal in the orthogonal complement of the embedding subspace [2]. The invariant subspace is obtained from the column vectors of the matrix ($\mathbf{VU}$) where $\mathbf{U}$ is the key in the form of a matrix. The control of the tradeoff lies in the dimension of $\mathbf{V}$. The increase of the dimension of $\mathbf{V}$ will increase the security of the CSW whereas its robustness decreases. In the case the CSW technique disregards the matrix $\mathbf{V}$, the CSW is equal to the TNW technique. According to [12], TNW does not reduce the robustness of the classical natural watermarking (NW) scheme.

C. SPAM features type steganalyzer

Our initial steganalyzer uses SPAM (Subtractive Pixel Adjacency Matrix) features [11] as input features. The classification was done with a 1-norm soft margin non-linear C-SVM classifier using a Gaussian kernel. The choice of the parameter C of the SVM and the width σ of the Gaussian kernel was based on the paired cross-validation procedure described in [13].

The SPAM features [11] make use of the dependencies between neighboring pixels by regarding their transition probabilities. For the 8-neighborhood $\{\nwarrow, \swarrow, \nearrow, \searrow, \leftarrow, \rightarrow, \uparrow, \downarrow\}$ of a pixel, the model determines the probabilities of the eight transitions. Let $X = (X_{ij}) \in \{0, ..., 255\}^{m_1 \times m_2}$ be an image. In the first step [11], the model calculates the difference array $\mathbf{D}^{\bullet}$, for each direction $\bullet \in \{\nwarrow, \swarrow, \nearrow, \searrow, \leftarrow, \rightarrow, \uparrow, \downarrow\}$ separately. For instance, the horizontal left-to-right transition, $D_{ij}^{\rightarrow}$ is calculated as

$$D_{ij}^{\rightarrow} = X_{ij} - X_{ij+1} \ , \tag{1}$$

where $1 \leq i \leq m_1, 1 \leq j \leq m_2 - 1$. The next step permits two options, either the first-order Markov process calculated by

$$M_{uv}^{\rightarrow} = Pr(D_{ij+1}^{\rightarrow} = u | D_{ij}^{\rightarrow} = v) \ , \tag{2}$$

or the second-order Markov process described as

$$M_{uvw}^{\rightarrow} = Pr(D_{ij+2}^{\rightarrow} = u | D_{ij+1}^{\rightarrow} = v, D_{ij}^{\rightarrow} = w) \ , \tag{3}$$

where $u, v, w \in \{-T, ..., T\}$, $1 \leq T \leq 255$. The SPAM features are obtained by averaging the eight matrices $\mathbf{M}^{\bullet}$ as follows

$$\begin{aligned} \mathbf{F}_{1,...,k} &= \tfrac{1}{4} \left[\mathbf{M}_{\cdot}^{\leftarrow} + \mathbf{M}_{\cdot}^{\rightarrow} + \mathbf{M}_{\cdot}^{\uparrow} + \mathbf{M}_{\cdot}^{\downarrow}\right] , \\ \mathbf{F}_{k+1,...,2k} &= \tfrac{1}{4} \left[\mathbf{M}_{\cdot}^{\nwarrow} + \mathbf{M}_{\cdot}^{\swarrow} + \mathbf{M}_{\cdot}^{\nearrow} + \mathbf{M}_{\cdot}^{\searrow}\right] , \end{aligned} \tag{4}$$

where $k = (2T+1)^3$ for the second-order features and $k = (2T+1)^2$ for the first-order features. In our implementation, the SPAM features were computed using the SPAM software provided on the BOSS website [6]. Originally, the authors in [11] propose to apply $T = 4$ for the first-order Markov process (162 features) and $T = 3$ for the second-order Markov process. Here, we use the second order features resulting in 686 values for the support vector machine.

D. Two-step classifier

In this work, the extended steganalyzer is composed of three 2-SVM classifiers: 2-SVM^{Pre}, 2-SVM$^{\mathcal{N}}$, and 2-SVM$^{\mathcal{W}}$ (see Figure 2). The proposed steganalyzer can be divided into two steps: the first 2-SVM^{Pre} works as a preselection step whereby the focus is on the classification between non-watermarked images and watermarked images; in the second step, we focus on the detection of stego and cover images by using two different classifiers: the 2-SVM$^{\mathcal{N}}$ for detecting non-watermarked cover and stego images and the 2-SVM$^{\mathcal{W}}$ for watermarked cover and stego images respectively. We use the identical 2-SVM and the SPAM features as input features as described above for all three classifiers, i.e. SPAM features are used for watermarking detection and steganalysis.

III. Experiments

Our test scenario is based on the image databases *BOSSbase* 0.92 [6], [7] containing 9,074 images and *BOWS-2* [8] containing 10,000 images. We select 8,000 images from each database resulting in 16,000 cover images in PNG format. For each cover image, we also generated a corresponding watermarked image using CSW according to [2]. Here, the embedded watermarked message is $\{$-1, 1$\}^{512}$. We reduce the robustness in favor of security by setting the size of the additional matrix $\mathbf{V}$ to the size of the input image. From both image types, we generated stego images with an embedding rate of $E_R = 40\,\%$ using LSB matching [9]. As a result, we obtained 64,000 (16,000 x 4) PNG images (Figure 4). Because of the popularity of JPEG formats, we produced 64,000 JPEG images in the same manner but using JCRW [1] for watermarking. In JPEG images, the embedded watermarking logo is shown in Figure 3. Each of the eight image sets (containing 16,000 images) was separated into two disjoint sets: a training set containing 4,000 and a test set containing 12,000 images (see Figure 4). In the case of $S^{\mathcal{W}}$, the image sets was first watermarked and then steganographic manipulated.

We report steganalysis performance as the error probability P_E at the point of the minimum overall error point of the *receiver operating characteristic* (ROC) curve, and as well as the area under the ROC curve (AUC). AUC was determined according to [14].

Figure 3: Embedded binary watermark logo (24×7)

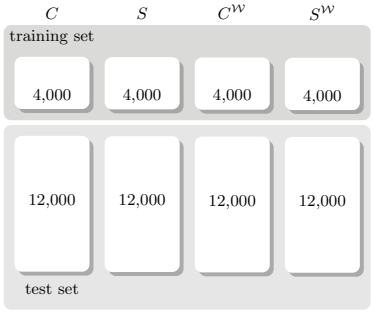

Figure 4: Training and test sets for PNG and JPEG images

A. Standard classifier trained on non-watermarked images

In the first experiment, we trained a standard 2-SVM with SPAM features (referred to as 2-SVM$^{\mathcal{N}}$) on 4,000 non-watermarked cover images and 4,000 non-watermarked stego images. Steganalysis performance was tested on the remaining 12,000 images of each image type. Table I shows that CSW-watermarked images $C^{\mathcal{W}}$ are typically misclassified as stego images $S^{\mathcal{W}}$. Only 1.70 % of the CSW-watermarked images $C^{\mathcal{W}}$ were correctly classified as cover images. In contrast, steganographically manipulated JCRW-watermarked images are mostly classified as cover images (8.88 % Table I) which means that LSB steganography is hidden by JCRW.

B. Training with watermarked images

In the second experiment, we augmented the training set with cover-stego pairs of watermarked images. The training set of the classifier consisted of the first 2,000 cover images and its corresponding 2,000 stego images and the last 2,000 watermarked cover images and its corresponding 2,000 watermarked stego images. We denote the steganalyzer with the augmented training set as 2-SVM$^{\mathcal{W}}$ (naïve approach). The naïve approach on the 2-SVM$^{\mathcal{W}}$ increases the detection performance both for watermarked cover images and watermarked stego images. But in the case of CSW, the detection performance of the non-watermarked images decreases from 94.31 % to 91.93 % in the case of C and from 95.72 % to 87.33 % in the case of S (see Table I). Apparently, steganography and watermarking mask each other during training.

C. Two-step steganalyzer

As a potential remedy for the decreased detection performance of the non-watermarked images for CSW, we tested

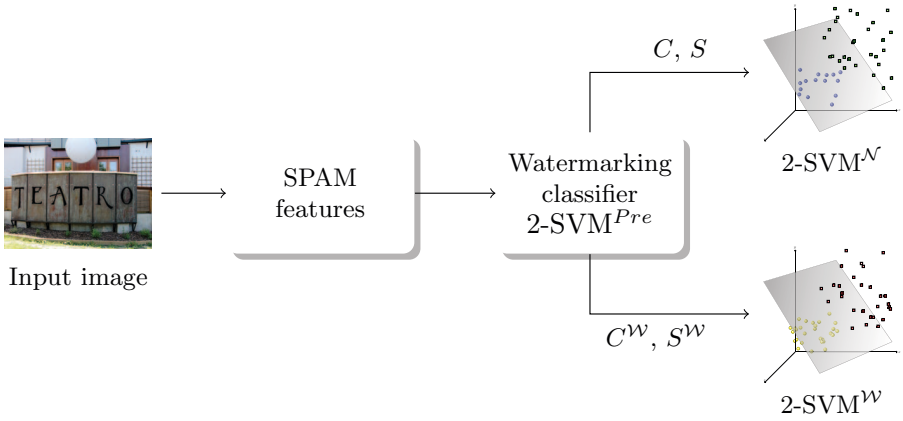

Figure 2: Two-step steganalyzer

Table I: Steganalysis performance of standard (2-SVM$^{\mathcal{N}}$) and naïve approach (2-SVM$^{\mathcal{W}}$). C, S, $C^{\mathcal{W}}$ and $S^{\mathcal{W}}$ denote the detection rates for the corresponding image classes.

		Non-watermarked test set				Watermarked test set			
Classifier	Watermark	AUC	P_E	C	S	$AUC^{\mathcal{W}}$	$P_E^{\mathcal{W}}$	$C^{\mathcal{W}}$	$S^{\mathcal{W}}$
2-SVM$^{\mathcal{N}}$	JCRW	99.99	9.14	98.79	99.73	72.06	49.68	91.76	8.88
	CSW	99.17	4.98	94.31	95.72	41.97	49.97	1.70	98.36
2-SVM$^{\mathcal{W}}$	JCRW	99.98	1.36	99.15	98.13	99.99	0.99	98.13	99.90
	CSW	95.80	10.37	91.93	87.33	99.99	1.23	97.82	99.72

the two-step steganalyzer in our third experiment. Note that we used only 4,000 images from each type of training set for all three classifiers according to Figure 4 without mixing training and test set. In a first preselection step, watermarks are detected by a 2-SVM^{Pre} classifier. Table II shows that the preselection step is highly accurate. In the second step, we use exactly the same two classifiers as in the first and second experiments: 2-SVM$^{\mathcal{N}}$ and 2-SVM$^{\mathcal{W}}$. We apply 2-SVM$^{\mathcal{W}}$ if the watermark detector 2-SVM^{Pre} classifies the input image as watermarked, otherwise we apply 2-SVM$^{\mathcal{N}}$. The final result of the two step classifier

Table II: Watermark detection performance of 2-SVM^{Pre} and 2-SVM$^{Pre}_{0.001}$

Classifier	C	$C^{\mathcal{W}}$	S	$S^{\mathcal{W}}$
2-SVM^{Pre}	99.29	96.17	99.00	98.93
2-SVM$^{Pre}_{0.001}$	99.91	87.37	99.86	95.20

(denoted as 2-Step-SVM^{Pre}) is shown in Table III. As intended, Table III suggests that the performance of the 2-Step-SVM^{Pre} is similar to the initial 2-SVM$^{\mathcal{N}}$ for discriminating between non-watermarked stego and cover images (P_E = 4.98 % vs. P_E = 5.11 %). If we further reduce the rate of missed watermarked images by an appropriate choice of the threshold of 2-SVM^{Pre} from 1 % to 0.1 % on the training set, we obtain a more accurate preselection step, denoted as SVM$^{Pre}_{0.001}$ in Table II. When the identical second step is applied to the output of SVM$^{Pre}_{0.001}$, we obtain the steganalyzer 2-Step-SVM$^{Pre}_{0.001}$ in Table III. The performance of the conservative 2-Step-SVM$^{Pre}_{0.001}$ classifier could almost reach the performance of the initial 2-SVM$^{\mathcal{N}}$ classifier in terms of P_E (P_E = 4.98 % vs. P_E = 4.99 %, see Table III). However, the performance on watermarked images decreases since the input of 2-SVM$^{\mathcal{W}}$ is more contaminated by non-watermarked images on which this classifier was not trained.

Table III: Comparison of steganalysis performance for the four investigated approaches (in %) for PNG (left) and CSW images (right)

Classifier	P_E	C	S	P_E	$C^{\mathcal{W}}$	$S^{\mathcal{W}}$
2-SVM$^{\mathcal{N}}$	4.98	94.31	95.72	49.97	1.70	98.36
2-SVM$^{\mathcal{W}}$	10.37	91.93	87.33	1.23	97.82	99.72
2-Step-SVM^{Pre}	5.11	94.65	95.13	10.29	91.73	87.69
2-Step-SVM$^{Pre}_{0.001}$	4.99	94.39	95.63	9.93	91.82	88.32

IV. Conclusion

In this paper, we investigated the influence of watermarked images on steganalysis. To our knowledge, the combined problem has not been studied in the literature so far. We found that both watermarking schemes disturb steganalysis. In particular, the standard steganalyzer based on SPAM features classified most of the CSW images as stego images. This shows that – as initially suspected – steganalysis can be highly perturbed or even made impossible in a realistic unbalanced image dataset scenario where watermarked images are likely to outnumber steganograms by far. The results we found in the pursuit of a possible answer to this problem can be summarized as follows:

- The naïve approach of training a standard 2-SVM$^{\mathcal{W}}$ on an augmented training set consisting of both watermarked and non- watermarked images reaches high performance in detecting watermarked stego and cover images, however at the price of a considerably reduced performance on non-watermarked images. This precludes this approach from practical application since watermarked images currently account only for a small percentage of all available images.
- A SPAM-feature-based classifier is capable of detecting CSW- and JCRW-watermarked images with high accuracy and thus can be used as a preselection step.
- Our proposed two-step steganalyzer reaches the performance of the initial 2-SVM$^{\mathcal{N}}$ classifier on non-watermarked images, with reduced performance on watermarked images. Currently, this poses no practical problem. However, if the proportion of watermarked images further increases, the design of the steganalyzer can be adapted, e.g. by tuning the preselection step to a different false alarm rate, and by training the secondary classifiers on the output of the preselection step.

According to our results, it seems that watermarked images are not well-suited as cover images for steganographic manipulation ($P_E^{\mathcal{W}}$ = 0.99 for JCRW and $P_E^{\mathcal{W}}$ = 1.23 for CSW, see Table I). Therefore, we think that the need for the detection of watermarked stego images is not a likely scenario.

In future work, we want to extend our investigation to different steganalyzers and watermarking schemes. Of special interest will be the question of how to combine a suite of specialized steganalyzers and watermarked detectors into one system. Here, the proposed two-step approach has the advantage that the two problems of detecting steganography and watermarks are decoupled and thus can be addressed separately.

References

[1] C.-H. Fan, H.-Y. Huang, and W.-H. Hsu, "A robust watermarking technique resistant jpeg compression," *J. Inf. Sci. Eng.*, vol. 27, no. 1, pp. 163–180, 2011.

[2] J. Cao and J. Huang, "Controllable secure watermarking technique for tradeoff between robustness and security," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 821–826, 2012.

[3] R. Chandramouli and N. D. Memon, "On sequential watermark detection," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1034–1044, 2003.

[4] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.

[5] M. Arnold, P. G. Baum, and X.-M. Chen, "Robust detection of audio watermarks after acoustic path transmission," in *Proceedings of the 12th ACM workshop on Multimedia and security*, ser. MM&Sec '10. New York, NY, USA: ACM, 2010, pp. 117–126. [Online]. Available: http://doi.acm.org/10.1145/1854229.1854253

[6] T. Filler, T. Pevný, and P. Bas, "BOSS (Break Our Steganography System)," 2010, software available at http://www.agents.cz/boss/.

[7] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system": The ins and outs of organizing BOSS," in *Information Hiding, 13th International Conference*, ser. Lecture Notes in Computer Science, T. Filler, T. Pevný, S. Craver, and A. D. Ker, Eds. Springer-Verlag, May 18–20 2011, pp. 59–70.

[8] European Network of Excellence ECRYPT, "BOWS-2 (Break Our Watermarking System)," 2008, software available at http://www.agents.cz/boss/ (accessed 2010).

[9] A. D. Ker and I. Lubenko, "Feature reduction and payload location with WAM steganalysis," in *Media Forensics and Security*, 2009, p. 72540.

[10] B. Schölkopf and A. J. Smola, *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[11] T. Pevný, P. Bas, and J. J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[12] B. Mathon, P. Bas, F. Cayre, and B. Macq, "Optimization of natural watermarking using transportation theory," in *Proceedings of the 11th ACM workshop on Multimedia and security*. ACM, 2009, pp. 33–38.

[13] V. Schwamberger and M. O. Franz, "Simple algorithmic modifications for improving blind steganalysis performance," in *Proceedings of the 12th ACM workshop on Multimedia and security*, ser. MM&Sec '10. New York, NY, USA: ACM, 2010, pp. 225–230. [Online]. Available: http://doi.acm.org/10.1145/1854229.1854268

[14] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The binormal assumption on precision-recall curves," in *Proceedings of the 20th International Conference on Pattern Recognition*. IEEE Computer Society, 2010, pp. 4263–4266.