Novelty Detection in Blind Steganalysis

Tomáš Pevný *
INPG - Gipsa-Lab
46 avenue Félix Viallet
Grenoble cedex 38031
France
pevnak@gmail.com

Jessica Fridrich
SUNY Binghamton
Department of ECE
Binghamton, NY, 13902-6000
001 607 777 6177
fridrich@binghamton.edu

ABSTRACT

It is generally believed that a blind steganalyzer trained on sufficiently many diverse steganographic algorithms will become universal in the sense that it will generalize to previously unseen (novel) stego methods. While this is a partially correct statement if the embedding mechanism of the novel method resembles some of the methods on which the classifier was trained, we demonstrate that if the classifier is presented with stego images produced by a completely different embedding mechanism, it may fail to detect the images as stego even for an otherwise fairly easily detectable method. Motivated by this observation, we explore two approaches for construction of universal steganalyzers—one-class and one-against-all classifiers. Their advantages and disadvantages are discussed and performance compared on a wide variety of steganographic algorithms. One-against-all classifiers have generally better performance than approaches based on characterizing just the class of covers but they may fail catastrophically on previously unseen stego algorithms. One-class methods are less likely to fail to detect unknown stego algorithms but have lower overall detection accuracy on known stego methods. The suitability of each approach thus depends on the application.

Keywords

Steganalysis, JPEG Images, Novelty detection

1. INTRODUCTION

The goal of blind steganalysis is to detect any steganographic method irrespective of its embedding mechanism. Construction of blind schemes starts with a few assumptions that we now review. As in Cachin's approach to steganographic security [1], it is assumed that there exists a random variable c on the space of all theoretically possible covers $\mathcal{C}$, with probability density function (pdf) P_c. The value $\int_{\mathcal{B}} P_c(x)dx$ is the probability of selecting $c \in \mathcal{B} \subset \mathcal{C}$ for hiding a message. Any practical blind steganalysis scheme, however, cannot work with the full representation of covers due to its large dimensionality. Instead, blind schemes work with a simplified model of $\mathcal{C}$ obtained by mapping $\mathcal{C}$ to a low-dimensional feature space $\mathcal{X}$, $\mathbf{x} : \mathcal{C} \mapsto \mathcal{X}$, (typically, $\mathcal{X} = \mathbb{R}^d$ and $\mathbf{x}(c) = (x_1(c), \dots, x_d(c)) \in \mathbb{R}^d$) inducing there a random variable $\mathbf{x}(\mathsf{c})$ with pdf $p_c(\mathbf{x})$. Blind steganalysis amounts to solving the following composite hypothesis testing problem

$$\begin{aligned} \mathrm{H}_0 &: \mathbf{x}(\mathsf{c}) \sim p_c \\ \mathrm{H}_1 &: \mathbf{x}(\mathsf{c}) \nsim p_c. \end{aligned} \tag{1}$$

For blind steganalysis to work, we need one more condition, which is surprisingly rarely discussed in the literature on blind steganalysis. The feature space needs to be complete in the sense

$$D(P_s||P_c) > 0 \Rightarrow D(p_s||p_c) > 0,$$

where D is the Kullback-Leibler divergence and P_s is the pdf of stego images. Finding a complete feature set, however, is a difficult problem and one we do not intend to study in this paper. Here, we will simply assume that we have a feature set that is approximately complete in the sense that it is hard to practically construct a stego scheme with $D(p_s||p_c) = 0$.

Applying classical detection theory to (1) would require estimating the pdf p_c, which is infeasible because the feature spaces that aspire to be complete in the above practical sense are still relatively high-dimensional. To obtain accurate parametric or non-parametric models of p_c, in practice the problem of estimating a pdf is replaced with yet a simpler problem of *classification*. A classifier can be trained on features $\mathbf{x}(c)$ for c drawn from a sufficiently large database.

Several avenues towards construction of blind steganalyzers can now be taken. One possibility is to characterize the cover features in the feature space by training a one-class detector capable of recognizing the covers. The potential problem with this approach is that the database has to be very large in the sense of number of images and *diverse*. The adjective 'diverse' should be emphasized because we certainly do not wish to misidentify processed covers (e.g., sharpened images) as containing stego. The second approach is to train a cover vs. all-stego binary classifier on two classes: cover images and stego images produced by a sufficiently large number of stego algorithms. The hope is that if the classifier is trained on all possible archetypes of embedding oper-

*The author has done this work while staying at Binghamton University.

ations, it should be able to generalize to previously unseen schemes. The third option is to train a multi-class detector capable of classifying images to known steganographic programs. Again, the hope is that a multi-classifier trained on sufficiently many algorithms will be able to correctly detect a stego image embedded using a novel stego method as stego (i.e., we desire the multi-classifier to recognize that the image is not a cover image).

In Section 2, it is show that a good multi-class detector does not necessarily have to be universal. On the example of a state-of-the-art multi-class detector for JPEG images [11], we demonstrate that while it is capable of detecting some novel stego methods, it may badly fail when presented with stego objects produced by schemes with a completely different and previously unseen embedding method. And this is despite the fact that the very same embedding method may be otherwise easily detectable using the same feature set by training a separate binary classifier with pairs of cover and stego images embedded using the unseen method. In Section 3, we identify sources of failure of the multi-class detector and present several approaches to solve this problem. The solutions are experimentally compared in Section 4. Finally, the paper is concluded in Section 5.

2. MULTI-CLASSIFIER FOR GENERAL STEGANOGRAPHY DETECTION?

In the past, some authors proposed steganalyzers that can classify stego images according to the stego method. An example of such a multi-class detector for JPEG images is [11]. We review only the most essential facts about this classifier in this paper, referring the reader to the original publication for more details. This classifier was selected intentionally because of its excellent performance (see the comparisons in [11, 4, 18, 9]).

2.1 The multi-class detector

The classifier was built around the Merged feature set consisting of 193 extended DCT features and a reduced set of 81 Markov features [11]. The DCT features capture inter-block dependencies among DCT coefficients and among pixels (blockiness), while the role of Markov features [17] is to describe intra-block dependencies. All features were calibrated [2] to make them less sensitive to image content and more sensitive to steganographic changes.

Support vector machines (SVM) were used to build a set of $\binom{k+1}{2}$ binary classifiers distinguishing between every two classes among k stego algorithms and a set of covers (total $k+1$ classes). Each classifier was a soft-margin SVM with Gaussian kernel. The kernel width and the penalty parameter were determined on a multiplicative grid using five-fold cross-validation on the training data. The threshold for each binary classifier was adjusted to produce less than 1% of false alarms on the training set.

The training set contained 3500 single-compressed JPEG 75% quality images of a wide variety of scenes acquired with 22 different digital cameras[1]. The results of all tests were generated from experiments on 2504 JPEG images of the same quality never seen by the classifier during training.

The multi-class detector was trained to recognize stego images from $k = 6$ popular JPEG steganographic algorithms: F5 [21], Model Based steganography with and without deblocking [14] (MBS1 and MBS2), JP Hide&Seek[2], OutGuess [13], and Steghide [3]. Even though this multi-class detector was not originally designed as a universal steganography detector, we may hope that by training it on a large number of diverse embedding techniques, it will become universal in the sense that it will assign a stego image produced by an unknown stego method to one of the 6 stego classes and not to the cover class. After all, there are not that many different ways how to slightly modify quantized DCT coefficients in a JPEG file.

2.2 Multi-class detector for novelty detection

The ability of the multi-class detector to generalize to previously unseen stego methods was assessed by first presenting it with stego images created by two methods on which the classifier was not trained: Jsteg[3] and the recently proposed MMx [5]. Jsteg uses simple LSB embedding in quantized DCT coefficients (coefficients 0 and 1 are skipped) along a pseudo-random path generated from a secret key. The MMx method is a more sophisticated algorithm that requires side information in the form of the uncompressed image. The algorithm minimizes the combined distortion due to quantization and embedding with modified matrix embedding using Hamming codes.

For Jsteg, 2504 images never seen by the classifier were embedded with messages of relative length 100%, 50%, and 25% of the embedding capacity. The stego images for MMx were embedded with random messages of relative length 2/3, 3/7, 4/15 bpac (bits per non-zero DCT coefficient). These payloads were selected to match the capacities determined by the co-dimension of the Hamming codes used for matrix embedding in MMx. The abbreviations MM2 and MM3 stand for the versions of the MMx algorithm that allow up to two or three modifications per embedding block, respectively. The algorithm security improves with the number of allowed changes. The quality factor for all stego images was again set to 75. As can be seen from Table 1, the multi-class detector reliably recognized stego images produced by Jsteg as containing secret messages even though Jsteg embedded images were not used for training the classifier. Note that Jsteg was mostly detected as F5 and OutGuess. Images embedded with the MMx algorithm were also reliably detected as stego and were assigned mostly to Model Based Steganography and Steghide (see Table 2). It is interesting to point out that the missed detection rate for MMx quickly increases with decreasing message length due to the decreased number of embedding changes arranged by matrix embedding. Here, we remark that both Jsteg and MMx can be detected more reliably using a targeted detector constructed from the same feature set (e.g., see the results in [6]).

The multi-class detector seems to be able to generalize well to both Jsteg and MMx. This is mainly due to their embedding mechanisms, which are similar to the schemes on which the multi-classifier was trained. For example, Jsteg, as well as OutGuess, use LSB embedding, which is probably why Jsteg was often classified as OutGuess.

Next, we decided to test the multi-class detector on stego images produced by a stego method with an entirely different embedding mechanism. To this end, we used the steganographic algorithm –F5 [6]. It embeds message bits into quantized DCT coefficients by changing their parity

[1]The image size ranged from 1.4Mpix to 6Mpix.

[2]`http://linux01.gwdg.de/~alatham/stego.html`

[3]`http://zooid.org/~paul/crypto/jsteg/`

	Cover	F5	JP H&S	MBS1	MBS2	OG	Steghide
Jsteg 100%	0.20%	57.91%	0.00%	0.00%	0.04%	41.81%	0.04%
Jsteg 50%	0.20%	57.59%	0.00%	0.04%	2.40%	39.58%	0.20%
Jsteg 25%	1.04%	57.63%	0.00%	5.47%	3.67%	30.99%	1.20%

Table 1: Confusion table of the multi-classifier on images from the testing set embedded by Jsteg. The multi-class detector was not trained to detect Jsteg images.

	Cover	F5	JP H&S	MBS1	MBS2	OG	Steghide
MM2-(1,3,2)	0.56%	0.92%	0.00%	0.80%	91.29%	1.92%	4.51%
MM2-(1,7,3)	0.92%	0.20%	0.00%	14.18%	27.08%	1.68%	55.95%
MM2-(1,15,4)	10.34%	0.44%	0.00%	27.52%	1.24%	0.68%	59.78%
MM3-(1,3,2)	0.44%	1.04%	0.00%	0.84%	91.61%	1.84%	4.23%
MM3-(1,7,3)	1.08%	0.20%	0.00%	15.06%	26.40%	1.88%	55.39%
MM3-(1,15,4)	17.05%	0.44%	0.04%	27.84%	1.16%	0.56%	52.92%

Table 2: Confusion table of the multi-class detector on images from the testing set embedded by MM2 and MM3. The multi-class detector was not trained to detect MM2 or MM3 images.

(LSB) along a pseudo-random path. If a parity of a DCT coefficient needs to be changed, instead of decreasing the absolute value of the coefficient as in F5, it is *increased*. This has a nice side effect of eliminating shrinkage from F5 (a situation when a non-zero DCT coefficient is changed to zero), which complicates the embedding mechanism of F5 and decreases its embedding efficiency (number of bits embedded per embedding change). Similar to F5, we used Hamming codes for matrix embedding to decrease the number of embedding changes. The embedding mechanism of none of the above 8 tested steganographic techniques is similar to –F5.

We performed the same type of experiment as with Jsteg and MMx. Total of 2504 images were embedded with a range of relative payloads and then presented to the classifier. The confusion matrix is displayed in Table 3. Quite surprisingly, –F5 is the most detectable for medium embedding rates and the least detectable at low embedding rates (as expected) but also at very high embedding rates! In other words, the multi-class detector completely failed to recognize images fully embedded with –F5 as containing stego content and instead classified them as covers. This is likely because the 6 individual binary classifiers distinguishing between covers and a stego method were all adjusted for low false positive rate, which is a necessity for any practical steganalytic tool. Consequently, because stego images fully embedded with –F5 did not resemble any of the stego images on which the classifier was trained, most of those 6 binary classifiers (cover vs. stego) conservatively assigned the image to the cover class.[4] Even though the decisions of the remaining $\binom{6}{2}$ binary stego vs. stego classifiers were biased towards F5 and OutGuess, stego classes usually did not get enough votes.

Despite the fact that the multi-classifier failed to detect –F5 embedded images as stego images, the –F5 is a poor steganographic method. We trained a targeted binary SVM classifier for –F5 on 3400 cover images and an even mixture of 3400 stego images embedded by –F5 with relative message lengths 5%, 10%, 20%, 25%, 50%, 75%, and 100%. The accuracy of this classifier (see Table 4) estimated from

[4]Additionally, the classifier was programmed to resolve ties in the number of votes for the cover, F5, and OutGuess classes by assigning the image to the cover class.

Target	Cover	–F5
–F5 100%	0.00%	100%
–F5 75%	0.00%	100%
–F5 50%	0.00%	100%
–F5 25%	0.04%	99.96%
–F5 20%	0.08%	99.92%
–F5 10%	0.32%	98.68%
–F5 5%	72.04%	27.95%
Cover	99.52%	0.48%

Table 4: Accuracy of a binary SVM classifier targeted to –F5 on 2504 test images.

2504 images from the testing set shows that –F5 is easily detectable even for relative payloads as small as 10% (recall that –F5 uses matrix embedding to further reduce the number of embedding changes). In fact, there are good reasons why –F5 is a bad choice for the steganographer. It can be shown that it introduces the largest combined distortion due to embedding and quantization [6] out of any embedding operation that changes a fraction of $\delta \geq 0$ coefficients towards zero and $1-\delta$ away from zero (–F5 is obtained for $\delta = 0$ and F5 corresponds to $\delta = 1$).

3. NOVELTY DETECTION IN STEGANALYSIS

Multi-class detection is a multiple hypothesis testing problem where the pdf of features for each hypothesis (stego method), $p_s(\mathbf{x})$, is sampled from stego images. On the other hand, a universal steganography detector is supposed to detect *all known and unknown* steganography algorithms, which leads to a much harder composite hypothesis testing problem. As the above experiment confirmed, a state-of-the-art multi-class steganography detector may not be a good general steganography detector (it is not universal).

Because a general steganography detector may never have enough information about the distribution of stego images $p_s(\mathbf{x})$, we can construct steganalysis by classifying *everything* that does not resemble a cover image to the stego class. Such a detector can be described by the decision function

Target	Cover	F5	JP HS	MBS1	MBS2	OutGuess	Steghide
–F5 100%	87.70%	7.23%	0.00%	0.04%	1.28%	3.71%	0.04%
–F5 75%	33.55%	23.12%	0.00%	0.20%	10.50%	30.47%	2.16%
–F5 50%	0.44%	0.08%	0.00%	2.52%	70.41%	1.36%	25.20%
–F5 25%	0.16%	0.00%	0.00%	12.86%	2.76%	0.28%	83.95%
–F5 20%	0.40%	0.00%	0.00%	12.86%	1.32%	0.32%	85.10%
–F5 10%	56.07%	0.36%	0.00%	3.99%	0.40%	1.12%	38.06%
–F5 5%	97.60%	0.76%	0.32%	0.16%	0.16%	0.60%	0.40%

Table 3: Confusion table of the multi-class detector on images from the testing set embedded by –F5. The classifier was not trained to detect –F5 images.

$h : \mathcal{X} \mapsto \{0, 1\}$

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } p_c(\mathbf{x}) > \lambda \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The free parameter λ in the decision function $h(\mathbf{x})$ is called the density level. It forms the threshold, above which the image is recognized as cover, and thus controls the trade-off between the probability of false alarm (cover image classified as stego), $\alpha = 1 - \int_{\mathcal{X}} h(\mathbf{x}) p_c(\mathbf{x}) \mathrm{d}\mathbf{x}$, and the probability of missed detection (stego image classified as cover), $\beta = \int_{\mathcal{X}} h(\mathbf{x}) p_s(\mathbf{x}) \mathrm{d}\mathbf{x}$.

The region of acceptance in (2) depends on the pdf $p_c(\mathbf{x})$ as well as on the density level λ. Without any prior information about h, the density and the parameter need to be estimated from samples, which makes the design of the detector inherently difficult.

The problem of designing the decision function (2) only from examples of one class (in our case the cover class) is known in the field of machine learning as the novelty / anomaly / density level detection problem. In the rest of this section, we describe some solutions [15, 19, 20, 8] to this problem that we believe are the most suitable for steganography. The approaches are experimentally compared in Section 4.

3.1 One-Class Support Vector Machines (OC-SVM)

For a fixed false positive rate α, the problem (2) can be approached by finding the minimum volume set $\mathcal{C}_\alpha$ (the decision region), so that the probability $p_c(\mathcal{C}_\alpha) \equiv \int_{\mathcal{C}_\alpha} p_c(\mathbf{x}) \mathrm{d}\mathbf{x} \geq 1 - \alpha$. Denoting the volume of $\mathcal{C} \subset \mathcal{X}$ as $\mu(\mathcal{C})$ for some $\mu : 2^{\mathcal{X}} \mapsto \mathbb{R}$ (where $2^{\mathcal{X}}$ is the power set of $\mathcal{X}$), we can write

$$\mathcal{C}_\alpha = \arg\min_{\mathcal{C} \subset \mathcal{X}} \{ \mu(\mathcal{C}) | \, p_c(\mathcal{C}) \geq 1 - \alpha \}. \quad (3)$$

In fact, we want to find the $1 - \alpha$ quantile of p_c assuming it exists. Estimators of this form are called minimum volume estimators.

To make the optimization problem (3) tractable, in OC-SVMs the minimum is taken over a restricted set $\mathcal{C} \in \mathcal{A} \subset 2^{\mathcal{X}}$ consisting of pre-images of all half-spaces in $\mathcal{F}$ under a mapping $\phi : \mathcal{X} \mapsto \mathcal{F}$ for some suitably chosen mapping ϕ and a space $\mathcal{F}$ equipped with the dot product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$

$$\mathcal{A} = \left\{ \mathcal{C} \subset \mathcal{X} | \, \exists w \in \mathcal{F}, (\mathbf{x} \in \mathcal{C}) \Leftrightarrow \langle w, \phi(\mathbf{x}) \rangle_{\mathcal{F}} > 0 \right\}.$$

Additionally, instead of minimizing the volume of $\mathcal{C}$, which might be difficult to calculate and the volume may not be finite, OC-SVMs minimize an SVM-style regularizer $\mu(\mathcal{C}) = \|w\|^2_{\mathcal{F}}$ controlling the length of the weight vector w (and consequently the complexity of the solution) in $\mathcal{F}$. The mapping ϕ as well as the space $\mathcal{F}$ are typically determined from a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ as $\phi(\mathbf{x}) = k(\cdot, \mathbf{x})$. The space $\mathcal{F}$, obtained by completing the space of all finite linear combinations $\sum a_i \phi(\mathbf{x}_i)$, $a_i \in \mathbb{R}$, is a space of functions $\mathcal{X} \mapsto \mathbb{R}$ ans is called the Reproducing Kernel Hilbert Space (RKHS) [16].

Denoting the training set $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$, the training of a OC-SVM leads to a quadratic programming problem on convex sets [15]:

$$\min_{w \in \mathcal{F}, \rho, \xi_i \in \mathbb{R}} \frac{1}{2} \|w\|^2_{\mathcal{F}} + \frac{1}{\nu l} \sum_{i=1}^{l} \xi_i - \rho \quad (4)$$

subject to

$$\begin{aligned} w \cdot \phi(\mathbf{x}_i) - \rho &\geq -\xi_i, \; i \in \{1, \ldots, l\} \\ \xi_i &\geq 0, \; i \in \{1, \ldots, l\}. \end{aligned}$$

The solution determines a hyperplane in $\mathcal{F}$ with w as its normal vector, $\{y \in \mathcal{F} | w \cdot y - \rho = 0\}$, and the decision function of the OC-SVM

$$h(\mathbf{x}) = \frac{1}{2}(1 + \mathrm{sgn}(w \cdot \phi(\mathbf{x}) - \rho)).$$

The optimization problem does not require all training samples to lie on the correct side of the hyperplane (w, ρ). Notice that if the slack variable $\xi_i > 0$, then the corresponding training sample $\mathbf{x}_i$ is classified as novelty $(w \cdot \phi(\mathbf{x}_i) - \rho < 0)$. The parameter ν controls the trade-off between the complexity of the solution and the number of misclassified points from the training set. Its role is to prevent the optimization reach a degenerate solution because there always exists a combination of (w, ρ) correctly classifying all training samples. If ν is set to desired false positive rate α, the OC-SVM asymptotically converges to the optimal solution of (3) [15].

The major issue with the use of OC-SVM is setting the parameters of the kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and the parameter ν controlling the false positive rate. In binary SVMs, this is usually done by estimating the performance by cross-validation on a finite grid of possible parameter values. Since the missed detection rate of a OC-SVM cannot be estimated (we have examples only from one class), this approach cannot be used. To illustrate this issue, one can imagine that it is always possible to choose the kernel wide enough to guarantee a zero false positive rate on testing set. However, the missed detection of this classifier would likely be very high and the lack of stego training examples prevents us to estimate it. The setting of the parameters of OC-SVM thus relies on experience of the user and heuristics. One

general heuristics[5] is to use the Gaussian kernel $k(\mathbf{x},\mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2)$ with $\gamma = \frac{1}{\eta^2}$, where η is the median of L_2 distances between samples in the feature space, and setting ν to the desired false positive rate α.

Minimum enclosing balls [15] employed by Farid and Lyu [7] for blind steganalysis with wavelet features, can also be casted as a OC-SVM.

3.2 One Class Neighbor Machine (OC-NM)

For the description of One Class Neighbor Machine [8], we need the notion of sparsity measure M. Let $S_l = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$ be a set of iid samples drawn according to pdf p_c. The function $M : \mathcal{X} \times S_l \mapsto \mathbb{R}$, defined for all $l \in \mathbb{N}$, is a sparsity measure if and only if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$, $p_c(\mathbf{x}) > p_c(\mathbf{y}) \Rightarrow M(\mathbf{x}, S_l) < M(\mathbf{y}, S_l)$. A sparsity measure characterizes closeness of the sample $\mathbf{x}$ to the set of training examples S_l. The rationale behind OC-NMs is to find a threshold ρ so that all samples $\mathbf{x}$ with $M(\mathbf{x}, S_l) > \rho$ are classified as anomalies, i.e., $h(\mathbf{x}) = \mathrm{sgn}(\rho - M(\mathbf{x}, S_l))$.

The training of an OC-NM is simple because we only need to find the threshold ρ. It starts with calculating the sparsity of all training samples $m_i = M(\mathbf{x}_i, S_l)$, $i \in \{1, \ldots, l\}$ and ordering them so that $m_1 \leq m_2 \leq \ldots \leq m_l$. By setting $\rho = m_{\lfloor(1-\alpha)l\rfloor+1}$, we ensure that at most α fraction of training samples are classified as anomalies. As in the case of OC-SVMs, it has been shown that OC-NMs converge to optimal solution with increasing number of training samples l [8].

Note that there is a key difference between utilizing samples S_l in OC-NM and in OC-SVM. While OC-SVMs only use a fraction of them during classification (support vectors defining the hyperplane), OC-NMs use all samples, which shows the relation to classifiers of the nearest neighbor type.

The original publication [8] presents several types of sparsity measures. The one we adopted here is based on the Hilbert kernel density estimator

$$M(\mathbf{x}, S_l) = \log\left(\frac{1}{\sum_{i=1}^{l} \frac{1}{\|\mathbf{x}-\mathbf{x}_i\|_2^{hd}}}\right). \tag{5}$$

Note the free parameter h in (5) controlling the smoothness of the measure.

3.3 Density Level Detection by Support Vector Machines (DLD-SVM)

Intuitively, if we had some information about the distribution of features of stego images, the performance of the steganography detector should improve. Steinwart et al. [19, 20] introduced an approach to anomaly detection problem that assumes that we have available samples from the pdf of stego images, μ. This converts the composite hypothesis testing problem (2) to a simple hypothesis test

$$\begin{aligned} \mathrm{H}_0 &: \quad \mathbf{x}(\mathsf{c}) \sim p_c \\ \mathrm{H}_1 &: \quad \mathbf{x}(\mathsf{c}) \sim \mu. \end{aligned} \tag{6}$$

The pdf μ expresses prior information about the possible location of novelties in the feature space. If no prior information is available, we can choose μ to be the least informative, e.g., uniform on $\mathcal{X}$.

The H_0 acceptance region, $\mathcal{R}_0$, is determined by the decision function $f(\mathbf{x}) : \mathcal{X} \mapsto \{-1, +1\}$, $\mathcal{R}_0 = \{\mathbf{x} \in \mathcal{X} | f(\mathbf{x}) = +1\}$ to be learned from available samples (the training set)

$$\{(\mathbf{x}_1, +1), \ldots, (\mathbf{x}_{\tilde{l}}, +1), (\mathbf{x}_{\tilde{l}+1}, -1), \ldots, (\mathbf{x}_l, -1)\},$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_{\tilde{l}} \sim p_c$ and $\mathbf{x}_{\tilde{l}+1}, \ldots, \mathbf{x}_l \sim \mu$. The decision function $f(\mathbf{x})$ can be learned by any method for binary classification. The authors showed that if the probability measure defined by p_c is absolutely continuous with respect to the probability measure defined by μ and if the decision function $f(\mathbf{x})$ is implemented by Support Vector Machines, this approach guarantees nearly optimal finite sample performance. In [19], DLD-SVMs were compared to other approaches and were reported to perform very well.

The problem with DLD-SVMs is that under no prior information about μ, the uniform distribution μ does not scale well with the dimensionality of the feature space d. The scalability issue can be illustrated by the following simple example. Let μ be uniform, the dimension of feature space be $d = 300$, and the number of training examples be 2×100000 (a very optimistic scenario). With this setting, we have relatively $\log_{300}(10000) \approx 2.01$ examples drawn according to μ per each dimension, which is clearly not enough to learn $f(\mathbf{x})$ with reasonable precision. We need examples from μ to "surround" examples from p_c.

The only way to remedy this curse of dimensionality is to localize the region of possible novelties. In steganography, we can do so by training a cover vs. all-stego classifier on examples of cover and stego images embedded by some "known" algorithms (algorithms detected by the multi-class detector from section 2). Similar to the multi-class detector, we hope that if the training set contains stego images from a large number of sufficiently diverse steganographic algorithms, the detector should be able to detect new algorithms. Unfortunately, as shown in Section 4 stego algorithms with previously unseen embedding mechanisms may be misclassified. On the other hand, this approach offers a very good detection accuracy on "known" algorithms, which is important if the steganography detector is used as a pre-classifier for a multi-class detector assigning images to known steganographic algorithms.

4. EXPERIMENTAL COMPARISON

In this section, we present experimental comparison of the novelty detection methods described in Section 3. The training and testing conditions were similar to the conditions under which the multi-class detector from Section 2 was prepared. The database of 6004 source raw images was divided into two disjoint sets. The first set with 3500 images was used only for training and the second set with 2504 images was used solely for testing. The steganographic techniques involved in our experiments were divided into known algorithms: F5, JP Hide&Seek, MBS1, MBS2, Steghide, and OutGuess, and unknown algorithms –F5, Jsteg, MM2, MM3, and F5 without shrinkage (nsF5) [6]. While the known algorithms could be possibly used during training of the classifiers, the unknown algorithms had to be kept unknown because they are needed to estimate the ability of the detectors to detect novel algorithms. All experiments were performed exclusively on single-compressed JPEG images with quality factor 75. All classifiers used the Merged feature set [11]. The parameters of methods were set either according to heuristics (OC-SVM, OC-NM) or by grid-search (DLD-SVM) to obtain 1% false positive rate.

[5]Private discussion with Bernard Schölkopf.

For a OC-SVM, we followed the heuristics described in Section (3.1) and set the width of the Gaussian kernel $\gamma = 0.181526$ according to the "median" rule, and $\nu = 0.01$, which is the desired false positive rate. The training data were scaled so that all features were in the range $[-1, +1]$ (the scaling parameters were derived from cover images only).

An important design decision in OC-NMs is the choice of the sparsity measure $M(\mathbf{x}, S_l)$. The original paper [8] describes several different sparsity measures. We tried all of them but only report the results for the measure (5), because it gave us the best performance. We used this measure with the following values of the parameter $h = \{0.01, 0.02, 0.05, 0.08, 0.1\}$ based on the recommendations in the paper. The detection accuracy varied very little with h. The results presented in this paper were obtained for $h = 0.01$.

The data pre-processing in DLD-SVM with μ being uniform probability distribution is not so straightforward, because the data from μ has to be artificially generated. We did so in the following manner. We first derived the scaling parameters on 3400 examples of cover images to bring all features to the range $[-1, +1]$. Then, we generated 15000 artificial samples from the underlying pdf μ according to uniform distribution $\mu = U([-1, +1]^d)$. Because the resulting training set with 18400 examples was imbalanced (there are more examples from one of the classes), we used weighted Support Vector Machines (2C-SVM) with Gaussian kernel. The hyper-parameters (C^+, C^-, γ) were selected from a multiplicative grid, where we estimated the accuracy on the cover and μ (stego) classes by means of a 5-fold cross-validation. We have to point out that all triplets (C^+, C^-, γ) evaluated during the grid-search had the false negative rate (class μ detected as cover) always equal to 0. This shows that we did not provide enough samples from μ. Even though it is easy to generate more samples from μ, the problem becomes quickly computationally intractable, since the complexity of training a SVM is approximately $l^{2.3}$, where l is number of training examples. Nevertheless, for the sake of completeness we did include results of this approach under the label "DLD-SVM$_{uni}$".

In order to localize the novelties in the input space, a binary SVM with Gaussian kernel was trained on 3400 examples of cover images and 3400 examples of images embedded by "known" algorithms with message lengths 100%, 50%, and 25% of their capacity (the only exception were images from MBS2 that were embedded with 30% of capacity of MBS1). As in the case of the DLD-SVM$_{uni}$, the hyper-parameters C and γ were determined by a grid-search combined with 5-fold cross-validation. This approach is a practical embodiment of a cover vs. all-stego binary classifier.

The accuracy of detectors was estimated on JPEG images created from 2504 raw images not used during training. We embedded messages with length 100%,75%, 50%, 25%, 20%, 10%, and 5% bits per non zero AC coefficient (bpac) by –F5, Jsteg and nsF5, and messages with length 0.66, 0.42, and 0.26 bpac by MM2 and MM3 (the message lengths for MMx correspond to the maximal messages for Hamming codes (1,3,2), (1,7,3), and (1,15,4)).

4.1 Accuracy on stego images

The DLD-SVM$_{loc}$ detector performed best on all known algorithms and all unknown algorithms with the exception of Jsteg, where it grossly failed. This is rather surprising considering the fact that the multi-class detector and DLD-SVM$_{loc}$ were constructed under similar conditions and Jsteg was easily detectable by the multi-class detector (Section 2). Apparently, DLD-SVM$_{loc}$ suffers from the same drawback as the multi-class detector. It may fail to detect stego algorithms with a completely different embedding mechanism. In contrast, all true novelty detection methods (OC-SVM, OC-NM, and DLD-SVM$_{uni}$) detected Jsteg even at low embedding rates.

In order to compare OC-SVM, OC-NM, and DLD-SVM$_{uni}$ more fairly, we shifted the threshold of OC-SVM so that the false positive rate of OC-SVM and OC-NM on testing images was the same. The performance of this shifted OC-SVM (labeled in Table 6 "OC-SVM$_{shift}$") is better than the performance of OC-NM, especially on "known" algorithms (Table 5).

Table 6 shows that the –F5 algorithm, which was not detected by the multi-class detector in Section 2, is now reliably detected by all classifiers except DLD-SVM$_{uni}$, which detected it only when the images were embedded with 75% or larger payload.

As expected, the DLD-SVM$_{uni}$ method performed the worst because it suffers from curse of dimensionality. It can only detect poor algorithms, such as –F5 or Jsteg.

The results also reveal differences between novelty and binary detectors. The DLD-SVM$_{loc}$ (and all binary classifiers in general) identify the boundary between cover and stego images only in those parts of the feature space that are occupied by the features from the known stego methods. In those regions of the feature space where the examples from the stego class are absent, the decision boundary can be too far from the cover class making the classifier vulnerable to catastrophic failures to detect stego images falling into this region. By contrast, novelty detectors try to find the decision boundary in all parts of the feature space, which makes them suitable for universal steganalysis.

The comparison of different universal steganography detectors shows that the choice has to be made with respect to the intended application. If the detector is going to be used as a pre-classifier module in a multi-class detector, the DLD-SVM$_{loc}$ (cover vs. all-stego) is a good choice because of its superior performance on "known" algorithms in comparison to other approaches (see Table 5). For a universal blind detector trained only on cover images, the OC-SVM offers slightly better performance than OC-NM, though the tricky setting of hyper-parameters makes them difficult to implement in practice.

4.2 Steganalysis of processed images

It has been recognized by the research community that the source of covers has a major influence on steganalysis in the spatial domain. Steganalysis of JPEG covers is generally expected to be less sensitive to the cover source due to the quantization performed during JPEG compression. The one-class novelty detectors described in the previous section, however, may be more sensitive to the cover source because they are only trained on covers. In this section, we test the performance of the blind steganalyzers on images that underwent various processing to see if covers processed by common image processing operations are likely to be mistaken for stego images by the universal blind steganalyzers.

To this end, the testing database of 2504 images was processed using the following operations: blurring with Gaus-

Target	OC-SVM	OC-SVM$_{shift}$	OC-NM	DLD-SVM$_{uni}$	DLD-SVM$_{loc}$
F5 100%	100.00%	99.60%	98.96%	1.92%	99.96%
F5 50%	78.11%	29.09%	20.10%	0.12%	99.60%
F5 25%	13.06%	2.64%	2.40%	0.12%	90.73%
JP Hide&Seek 100%	100%	99.68%	99.52%	0.52%	99.84%
JP Hide&Seek 50%	85.18%	54.19%	41.73%	0.48%	98.28%
JP Hide&Seek 25%	40.13	21.60%	19.04%	0.44%	73.52%
MBS1 100%	100.00%	100.00%	99.92%	0.20%	99.96%
MBS1 50%	97.88%	53.36%	29.50%	0.16%	99.80%
MBS1 30%	35.12%	7.03%	4.27%	0.12%	98.88%
MBS1 25%	21.33%	3.59%	2.56%	0.12%	96.81%
MBS1 15%	9.95%	1.84%	1.76%	0.12%	71.19%
MBS2 30%	93.49%	55.95%	32.47%	0.12%	99.12%
MBS2 15%	33.63%	5.51%	2.88%	0.12%	77.92%
OutGuess 100%	100.00%	100.00%	100.00%	1.72%	99.96%
OutGuess 50%	99.80%	80.07%	57.51%	0.20%	99.96%
OutGuess 25%	41.25%	8.15%	5.19%	0.12%	98.12%
Steghide 100%	100.00%	99.96%	99.44%	0.20%	99.96%
Steghide 50%	85.90%	27.76%	16.61%	0.16%	99.84%
Steghide 25%	18.61%	4.19%	2.84%	0.20%	96.37%

Table 5: Detection accuracy of universal blind steganalyzers on known stego algorithms. Note that the DLD-SVM$_{loc}$ was the only detector that needed examples of stego images for its training. The detector OC-SVM$_{shift}$ is an OC-SVM classifier with the threshold shifted to match the false positive rate of the OC-NM classifier.

Target	OC-SVM	OC-SVM$_{shift}$	OC-NM	DLD-SVM$_{uni}$	DLD-SVM$_{loc}$
–F5 100%	100%	100%	100.00%	98.88%	99.08%
–F5 75%	100%	100%	100.00%	89.50%	99.44%
–F5 50%	100%	100%	100.00%	6.75%	99.60%
–F5 25%	100%	95.64%	93.93%	0.12%	98.48%
–F5 20%	99.6%	66.57%	55.87%	0.16%	96.09%
–F5 10%	17.73%	3.7%	3.27%	0.16%	33.11%
–F5 5%	6.70%	1.55%	1.48%	0.12%	3.55%
nsF5 100%	100%	100%	99.96%	16.41%	99.96%
nsF5 75%	100%	99.96%	99.92%	3.04%	99.96%
nsF5 50%	98.76%	74.56%	80.91%	0.20%	99.72%
nsF5 25%	11.50%	2.87%	3.19%	0.12%	88.86%
nsF5 20%	9.78%	2.07%	2.24%	0.12%	72.12%
nsF5 10%	5.99%	1.47%	1.44%	0.12%	6.11%
nsF5 5%	5.47%	1.31%	1.40%	0.12%	1.72%
MM2-(1,3,2)	100%	100%	100.00%	18.37%	99.64%
MM2-(1,7,3)	100%	100%	99.92%	0.12%	99.20%
MM2-(1,15,4)	62.61%	20.24%	17.69%	0.12%	53.67%
MM3-(1,3,2)	100%	100%	100.00%	18.29%	99.72%
MM3-(1,7,3)	100%	100%	99.92%	0.12%	99.32%
MM3-(1,15,4)	51.71%	17.17%	15.14%	0.12%	58.51%
Jsteg 100%	100%	100%	100%	98.24%	42.41%
Jsteg 75%	100%	100%	100%	87.85%	42.33%
Jsteg 50%	100%	100%	100%	66.85%	42.37%
Jsteg 40%	100%	100%	100%	60.54%	42.29%
Jsteg 25%	100%	99.84%	99.45%	56.94%	42.05%
Jsteg 20%	99.88%	99.12%	98.09%	56.78%	42.09%
Jsteg 10%	96.13%	83.11%	65.36%	56.62%	32.98%
Jsteg 5%	78.87%	63.53%	40.27%	56.62%	5.99%
Cover	94.76%	98.64%	98.64%	99.88%	98.96%

Table 6: Detection accuracy of universal blind steganalyzers on four unknown stego algorithms. The detector OC-SVM$_{shift}$ is an OC-SVM classifier with the threshold shifted to match the false positive rate of the OC-NM classifier.

Target	OC-SVM	OC-SVM$_{shift}$	OC-NM	DLD-SVM$_{uni}$	DLD-SVM$_{loc}$
Blurring $\sigma = 0.4$	94.33%	98.48%	98.92%	99.88%	98.84%
Blurring $\sigma = 0.8$	93.97%	98.32%	98.76%	99.80%	98.84%
Blurring $\sigma = 1.2$	91.85%	98.00%	98.72%	99.76%	98.76%
Blurring $\sigma = 1.6$	88.06%	97.68%	98.52%	99.80%	98.44%
Blurring $\sigma = 2.0$	79.03%	96.92%	98.08%	99.80%	98.04%
Color quantization	93.13%	98.72%	99.00%	99.88%	97.60%
Despeckling	93.05%	98.08%	98.68%	99.76%	98.64%
Gamma corr. $\gamma = 0.7$	95.21%	98.52%	98.88%	99.84%	98.64%
Normalization	94.97%	99.04%	99.44%	100.00%	99.60%
Sharpened $\sigma = 0.4$	94.81%	98.72%	98.52%	99.88%	98.84%
No processing	94.76%	98.64%	98.64%	99.88%	98.96%

Table 7: Percentage of processed covers detected correctly as covers.

sian kernel with kernel width $\sigma \in \{0.4, 0.8, 1.2, 1.6, 2.0\}$, sharpening with $\sigma \in \{0.4, 0.8, 1.2, 1.6, 2.0\}$, despeckling, color quantization to 256 colors, histogram normalization in all three color channels, and gamma correction with $\gamma \in \{0.7, 0.9, 1.1, 1.3\}$. All operations were carried out in Image Magick's Convert routine. To avoid producing double compressed JPEGs, we always processed the raw, never compressed image and then saved it as 75% quality JPEG.

Table 7 shows the percentage of correctly classified processed covers by all five tested steganalyzers. Blurring with Gaussian kernel with $\sigma = 1.6$ and $\sigma = 2.0$ increased the false positive rate the most, especially for OC-SVM. The other processing did not have a significant influence on the detection accuracy.

Because the Merged features are computed directly from quantized DCT coefficients, they are very sensitive to repetitive JPEG compression. A double-compressed cover image will exhibit artifacts due to double quantization of DCT coefficients, which is likely to be misinterpreted by the one-class detectors as an anomalous image. Table 8 confirms this educated guess. It shows the percentage of correctly classified covers that were double JPEG compressed with the primary quality factor PQF $\in \{65, 70, 80, 85, 90\}$ and secondary quality factor 75. The negative influence of double compression on steganalysis that uses features computed from DCT coefficients is well-known. This problem can be resolved by estimating the primary quantization matrix [12] and training appropriate detectors for double-compressed JPEG images [10].

Overall, we can say that OC-NM is more robust to processing than OC-SVM$_{shift}$ (see Tables 7 and 8). On the other hand, as reported in Section 4, OC-SVM$_{shift}$ better detects stego content than OC-NM. This indicates that the decision boundary of OC-SVM$_{shift}$ surrounds cover images more tightly.

Binary classifiers (DLD-SVM$_{uni}$ and DLD-SVM$_{loc}$) are less likely to misclassify processed images (especially double-compressed images) than novelty detectors.

5. CONCLUSIONS

A steganalyzer trained to detect variety of steganographic algorithms does not necessarily have to be a good universal steganography detector because it can fail to recognize images produced by steganographic methods with a completely novel embedding mechanism as stego. This applies to both multi-class detectors and binary cover-against-all-stego detectors.

The task of recognizing novelty in machine learning is also known as anomaly detection. We adapted several existing approaches to anomaly detection for steganalysis and compared their performance. The methods differed in their machine learning techniques as well as in utilizing side information in the form of examples of "known" steganographic algorithms. Among the techniques that do not utilize any information about stego images, the one-class SVM trained only on examples of cover images had the best overall performance and was less prone to failures to detect an unknown stego method. The detection accuracy of one-class detectors on known stego algorithms is understandably somewhat worse than detection accuracy of binary cover-against-all-stego detectors trained on such stego images.

For applications where reliable universal blind detector is required, such as for automatic traffic monitoring, targeted steganalyzers or multi-class detectors should be supplemented with a reliable one-class detector.

6. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grant number FA8750-04-1-0112 and by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFRL, AFOSR, or the U.S. Government.

Tomáš Pevný is partially supported by the National French projects Nebbiano ANR-06-SETIN-009, ANR-RIAM Estivale, and ANR-ARA TSAR.

7. REFERENCES

[1] C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–318, Portland, OR, April 14–17, 1998. Springer-Verlag, New York.

[2] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of

PQF	OC-SVM	OC-SVM$_{shift}$	OC-NM	DLD-SVM$_{uni}$	DLD-SVM$_{loc}$
65	0.00%	0.00%	0.00%	4.67%	0.04%
70	0.00%	0.08%	0.28%	95.61%	0.12%
80	0.00%	0.00%	0.32%	99.32%	99.84%
85	0.00%	0.00%	0.08%	92.93%	99.84%
90	6.67%	27.76%	38.54%	99.96%	32.15%

Table 8: Percentage of correctly classified covers that were double-compressed using primary quality factor PQF and secondary quality factor 75.

steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[3] S. Hetzl and P. Mutzel. A graph–theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, Salzburg, Austria, September 19–21, 2005.

[4] M. Kharrazi, H. T. Sencar, and N. D. Memon. Benchmarking steganographic and steganalytic techniques. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 252–263, San Jose, CA, January 16–20, 2005.

[5] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[6] J. Kodovský, J. Fridrich, and T. Pevný. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[7] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.

[8] A. Munoz and J. M. Moguerza. Estimation of high-density regions using one-class neighbor machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):476–480, 2006.

[9] T. Pevný and J. Fridrich. Towards multi-class blind steganalyzer for JPEG images. In M. Barni, I. J. Cox, T. Kalker, and H. J. Kim, editors, *International Workshop on Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, Siena, Italy, September 15–17, 2005. Springer-Verlag, Berlin.

[10] T. Pevný and J. Fridrich. Multiclass blind steganalysis for JPEG images. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 257–269, San Jose, CA, January 16–19, 2006.

[11] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29 – February 1, 2007.

[12] T. Pevný and J. Fridrich. Estimation of primary quantization matrix for steganalysis of double-compressed JPEG images. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 11 1–11 13, San Jose, CA, January 27–31, 2008.

[13] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Proceedings of the ACM Symposium on Applied Computing, August 13–17, 2001.

[14] P. Sallee. Model-based steganography. In T. Kalker, I. J. Cox, and Y. Man Ro, editors, *Digital Watermarking, 2nd International Workshop*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167, Seoul, Korea, October 20–22, 2003. Springer-Verlag, New York.

[15] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), 2001.

[16] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, 2001.

[17] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of *Lecture Notes in Computer Science*, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[18] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of *Lecture Notes in Computer Science*, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

[19] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine*

Learning Research, 6:211–232, 2005. Los Alamos National Laboratory Technical Report LA-UR-04-4716.

[20] I. Steinwart, D. Hush, and C. Scovel. Density level detection is classification. *Neural Information Processing Systems*, 17:1337–1344, 2005. Los Alamos National Laboratory Technical Report LA-UR-04-3768.

[21] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.