Detecting messages of unknown length

Tomáš Pevný

Department of Cybernetics at Czech Technical University in Prague, Czech Republic

ABSTRACT

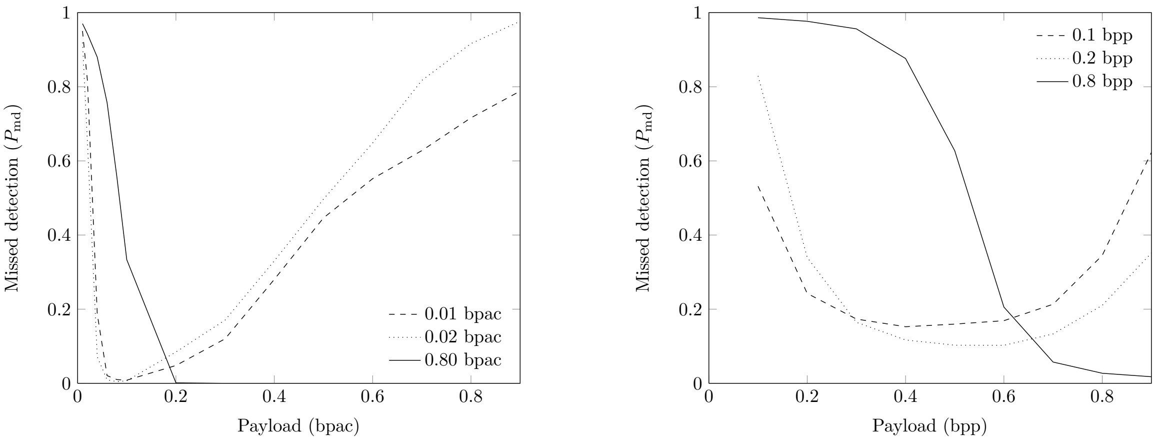
This work focuses on the problem of developing a blind steganalyzer (a steganalyzer relying on machine learning algorithm and steganalytic features) for detecting stego images with different payload. This problem is highly relevant for practical forensic analysis, since in practice, the knowledge about the steganographic channel is very limited, and the length of hidden message is generally unknown. This paper demonstrates that the discrepancy between payload in training and testing / application images can significantly decrease the accuracy of the steganalysis. Two fundamentally different approaches to mitigate this problem are then proposed. The first solution relies on quantitative steganalyzer. The second solution transforms one-sided hypothesis test (unknown message length) to simple hypothesis test by assuming a probability distribution on length of messages, which can be efficiently solved by many machine-learning tools, e.g. by Support Vector Machines. The experimental section of the paper (a) compares both solutions on steganalysis of F5 algorithm with shrinkage removed by wet paper codes for JPEG images and LSB matching for raw (uncompressed) images, (b) investigates the effect of the assumed distribution of the message length on the accuracy of the steganalyzer, and (c) shows how the accuracy of steganalysis depends on Eve's knowledge about details of steganographic channel.

1. INTRODUCTION

The usual scenario considered in steganography assumes that the eavesdropper, Eve, knows all details about the steganographic channel used by Alice and Bob except the stego key (the Kerckhoffs' principle). Particularly, Eve knows the steganographic algorithm, and probability distributions of cover images, hidden messages, and stego-keys. In this scenario, important for the evaluation of the security of steganographic algorithms, the security of the channel entirely relies on the stego-key. From the Eve's point of view, the assumptions about her knowledge are not realistic, since her knowledge is usually limited. In some cases, she might know the steganographic algorithm, but she rarely knows the length of hidden message (or the probability distribution of messages).

The accuracy of Eve's analysis obviously depends on her knowledge about the steganographic channel. This is especially true for the steganalyzers relying on a combination of machine learning tools and steganalytic feature (blind / feature-based steganalysis), where the large part of the knowledge is learned from the examples of cover and stego objects. This inference from the training set in fact tunes the steganalyzer to this particular setting. While the tuning can be in some cases advantageous, for example when Eve knows the source of cover images and other side information, generally, it is the opposite because Eve aims to have steganalyzer for as diverse set of stego objects as possible. As will be shown, she has to be aware of this problem of over-fitting her steganalyzer to particular conditions.

Feature-based steganalyzers, which nowadays present a state of the art in steganalysis, are very sensitive to discrepancies between development (training) and application (testing) conditions. Despite the seriousness of this problem, a little attention has been paid to it. To the best of our knowledge, the only works describing this phenomenon are[2] and.[10] Both works practically demonstrated that steganalyzers trained on different type of images than evaluated exhibit decreased performance. The first one also showed the same phenomenon for discrepancy in message length. To illustrate the problem, graphs on Figure 1 show the probability of missed detection of steganalyzers trained on stego images with a fixed payload. It is clearly seen that steganalyzers trained on low payloads fail to detect stego images with large payloads and vice versa. This behavior was already reported in,[2] but the decrease of the accuracy in the cited work was not as substantial as observed here.



(a) nsF5 (b) LSB matching

Figure 1: Probability of missed detection (1− probability of false negative) of steganalyzers of nsF5 (left) and LSB matching (right) algorithms trained on images with fixed payloads 0.01, 0.02, and 0.8bpac. / 0.1,0.2, and 0.8bpp respectively. All steganalyzers have fixed false positive rate at 0.01 on the testing set. The missed detection rates were estimated on images from the testing set. Graph of nsF5 shows the average payload, since in reality, we fixed the payload with respect to square root of non-zero AC coefficients.

In this work, we focus on the effect of mismatch between payload of images used for training and evaluation of feature-based steganalyzers. We believe that this problem is highly relevant for practical forensics analysis, when investigators are in possession of computer with a steganographic program. They want to detect the stego objects but they do not know the message length. The detectors not targeted to specific payload are also interesting from the point of view of Square Root Law[4,7] stating that for imperfect steganalytic schemes, their capacity grows with square root of number of usable (changeable) elements. This means that in practice it might be difficult to target the detector to certain relative message length, and detectors developed under the assumption of unknown message length might exhibit better accuracy than detectors targeted to fixed payload.

We present two very different approaches towards this problem. First solution consists from training a Support Vector Machine classifier on a set of cover and stego images with mixed payload. The second approach is based on thresholding output of quantitative steganalyzers implemented by Support Vector Regression.[13] Both solutions are compared to the "optimal" case when Eve knows the length of the hidden message, and to the case when Eve does not know anything about the steganographic channel except the samples from the probability distribution of cover images (i.e. universal steganalysis[12]).

This paper is organized as follows. Section 2 describes three common scenarios from Eve's point of view and defines hypothesis testing problems she needs to solve. Section 3 presents proposed solutions to the problem of detecting unknown payload. These solutions are experimentally compared in Section 4 on steganalysis of F5 with shrinkage removed by wet paper codes and of LSB matching. The same section also investigates, how the accuracy of Eve's classification depends on her knowledge about the channel. The paper is concluded in Section 5.

2. THEORETICAL POINT OF VIEW

This section presents three most important scenarios Eve might face, depending on her knowledge about the steganographic channel (Figure 2). For each of them, a detector according to detection theory is proposed.

The outline of the steganographic channel is depicted on Figure 2. The steganographic algorithm (consisting from embedding function $S_{\mathrm{E}} : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \mapsto \mathcal{C}$ and extracting function $S_{\mathrm{X}} : \mathcal{C} \times \mathcal{K} \mapsto \mathcal{M}$) together with probability distributions on cover images P_{C}, messages P_{M}, and keys P_{K} implicitly defines the probability distribution on stego objects P_{S}. In theory, if Eve knows exactly the probability distribution of cover objects

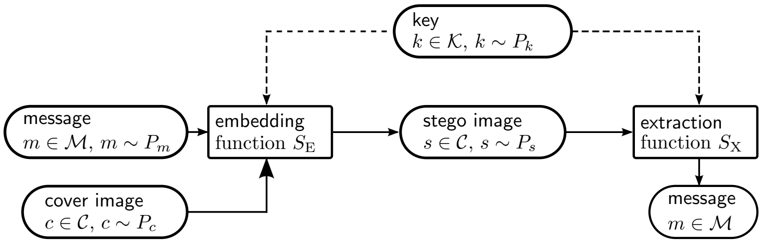

key
$k \in \mathcal{K}$, $k \sim P_k$
message
$m \in \mathcal{M}$, $m \sim P_m$
embedding
function S_{E}
stego image
$s \in \mathcal{C}$, $s \sim P_s$
extraction
function S_{X}
cover image
$c \in \mathcal{C}$, $c \sim P_c$
message
$m \in \mathcal{M}$

Figure 2: Steganographic channel

P_{C} and stego objects P_{S}, then she can create the perfect detector. But, what does Eve knows in practice? To which extent is she able to estimate the probability distribution of stego objects P_{S}?

Except special cases, the probability distribution of cover objects P_{C} is generally unknown. This problem is usually resolved by (a) projecting space of all cover objects $\mathcal{C}$ to low-dimensional feature space $\mathbb{R}^d$ by means of steganalytic feature set $f : \mathcal{C} \mapsto \mathbb{R}^d$, and by (b) estimating probability distribution $f(P_{\mathrm{C}})$ in the low-dimensional feature space from a finite set of samples. This turns out that if Eve knows at least the type of cover objects (digital photographs, scans, fixed size images, etc), she has at least a partial knowledge of P_{C} (the problem of cover mismatch is not solved here). Consequently in this paper it is assumed that Eve has some knowledge of P_{C} obtained through sampling, which albeit not being perfect allows her to create a detector. The same can be assumed about the distribution on steganographic keys P_{K}[*]. What remains unknown is the probability distribution of messages P_{M}, and may-be steganographic algorithm $(S_{\mathrm{E}}, S_{\mathrm{X}})$.

In the first scenario, Eve knows all details about the channel, namely probability distribution of stego-messages P_{M} and the steganographic algorithm. Consequently, she has the same knowledge about P_{S} as about P_{C}. In this case the detection problem can be then written as a simple hypothesis test

$$\begin{aligned} H_0 &: \mathbf{x} \sim P_{\mathrm{C}} \\ H_1 &: \mathbf{x} \sim P_{\mathrm{S}}. \end{aligned} \tag{1}$$

As was discussed above, the (1) case is of a limited use for practice. In this paper we call the detector for this scenario *clairvoyant detector.*

In the second scenario, Eve knows the embedding algorithms, but she does not know the distribution of stego messages P_{M}. For most practical settings, it can be safely assumed the hidden message to be encrypted and / or compressed, which makes the relative entropy of hidden message equal to one. Under this assumption, the distribution of stego messages P_{M} is parametrized only by the relative length of embedded message α, and Eve can get knowledge about the P_{S} parametrized by α.[†] Eve's optimal strategy is to perform a one-side hypothesis test

$$\begin{aligned} H_0 &: \alpha = 0 \\ H_1 &: \alpha > 0. \end{aligned} \tag{2}$$

Because for any reasonable steganographic algorithm should hold that $\lim_{\alpha \to 0} P_{\mathrm{S}}(\alpha) = P_{\mathrm{C}}$, to avoid excessively high false positive rate, Eve might want to set up a threshold, α_0, from which the images will be deemed as stego ones. The test would then look like

$$\begin{aligned} H_0 &: \alpha = 0 \\ H_1 &: \alpha > \alpha_0. \end{aligned} \tag{3}$$

[*] In practice the probability P_{K} will not be uniform, because humans tend to choose passwords that are easy to remember. It is more realistic to assume that the P_{K} is related to some dictionary.

[†] Due to the square root law,[4,7] the relative length of embedded message should be defined with respect to square root of number of usable elements in the cover object.

In this paper, we called detectors built for this scenario *targeted.*

The third scenario captures the case, when Eve does not know the embedding algorithms nor the message length. The detector aims to solve a composite hypothesis problem

$$\begin{aligned} H_0 &: \mathbf{x} \sim P_{\mathrm{C}} \\ H_1 &: \mathbf{x} \nsim P_{\mathrm{C}}. \end{aligned} \tag{4}$$

This case has been already treated in the prior art[12] and detectors for this problem are called *universal*, because they detect any steganographic algorithm (providing that the features are sensitive to the embedding operation and the message is long enough).

3. PRACTICAL SOLUTIONS

The previous section identified three most important situations Eve can face depending on the degree of her knowledge about the steganographic channel. This section proposes several practical approaches to deal with. It is assumed that Eve posses a set of cover images similar to the ones used by Alice and Bob and a set of steganographic features sensitive to the steganographic scheme.

The clairvoyant detectors solving the simple hypothesis test (1) have been already investigated. The usual approach, and the approach adopted here as well, is to create set of stego-images with a given payload and use machine learning algorithms (e.g. Support Vector Machines, Fisher Linear Discriminant, etc.) to create a detector targeted to given payload and steganographic algorithm. Because of the square root law, the payload should be defined with respect to square root of the number of usable elements (more in the experimental Section 4).

The targeted detector for one sided hypothesis test (2) is the main interest of this paper. Two, fundamentally different solutions are presented and compared.

1. Eve uses quantitative steganalyzer estimating relative payload α. Eve deems all images with estimated payload larger than threshold α_0 as stego images. The quantitative steganalyzer can be either heuristic,[6] or feature based.[13] To determine the threshold α_0, Eve uses the Neyman-Pearson criteria maximizing detection accuracy (probability of detecting stego images as stego images) under the constraint the false positive rate (cover images detected as stego images) being below given bound. This approach with bound on the false positive rate equal to 1% has been used in all experiments presented in Section 4.

2. Eve decides to convert the problem to simple hypothesis test by assuming a probability distribution on the payload (lengths of messages). Coupling this assumptions with the assumption that messages are encrypted / compressed before the embedding (see Section 2) gives her the needed knowledge of the probability distribution of messages P_{M}, and consequently P_{S} as well. Now, she can convert the one-sided hypothesis (2) test to the simple hypothesis test (1), which can be again solved by any machine learning algorithm. The remaining question is, which probability distribution on payload Eve should use. In the experimental section, we show that in order to be safe, Eve should use the uniform distribution.

The universal detectors for the third scenario are not the concern of this paper. The experiments presented in experimental section uses the approach based on one-class Support Vector Machines described in.[12]

4. EXPERIMENTS

This section presents comparison of detectors introduced in Section 3 on the steganalysis of F5 with shrinkage removed by wet paper codes and on the steganalysis of LSB matching. Because the development conditions and evaluation methodology is the same for both algorithms, the results are presented side by side, as the conclusions are very similar. The section starts with description of image databases used for experiments in

JPEG and spatial domain. Then, it proceeds to description of individual detectors (clairvoyant, targeted, and universal detectors). The section is closed by comparing accuracy of all three detectors.

Unless otherwise said, the thresholds of all classifiers were set such that the probability of false alarms on the *testing set* was 1%. Although this is not correct from the application point of view, because the setting of the threshold is a part of the classifier design for which the images from testing set should never be used, here it serves the purpose. It allows us to compare all approaches at the exactly same level of false positive rate.

4.1 Image databases

4.1.1 JPEG domain

Experiments in JPEG domain were performed on a database of approximately 9200 single-compressed JPEG images with quality factor 80. Images were acquired by 23 different digital cameras and had different sizes. Prior all manipulations, images were divided into training and testing set of equal size. By using F5 with shrinkage removed by wet paper codes and matrix embedding turned off (nsF5)[‡][9] (the original F5 algorithm was published in[14]), 15 sets of stego images were created: 14 sets with average payloads $\mathcal{L}_{\text{nsF5}}$ in bits per non-zero AC coefficient (bpac),

$$\mathcal{L}_{\text{nsF5}} = \{0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, \ldots, 0.8, 0.9\},$$

and one set with a mixed payload distributed uniformly in the range $[0, 0.90]$bpac. Because images in the database had different sizes and different numbers of AC coefficients, the relative message length was fixed with respect to square root of non-zero AC coefficients. The payloads $\mathcal{L}_{\text{nsF5}}$ shows the average payload in bits per non-zero AC coefficients in a given set. To mitigate the effects of the square root law, the actual length of message inserted into i-th image was $p\sqrt{\bar{n}n_i}$, where $p \in \mathcal{L}_{\text{nsF5}}$ is the payload, n_i is the number of non-zero AC coefficients in the i-the image, and $\bar{n}$ is the average of the same quantity over whole database of 9200 images. For the used image database, the $\sqrt{\bar{n}} = 677$.

It should be point out that nsF5 algorithm frequently fails to insert messages of average length of 0.7 bpac and higher. Due to this problem, experiments on high payload were evaluated on smaller number of images.

All steganalyzers for JPEG domain employed PF274 feature set,[11] which despite not being the state of the art offers a very good performance in detecting the nsF5 algorithm.[8]

4.1.2 Spatial domain

Experiments in spatial domain were performed on 10700 greyscale images of fixed size 512×512 from BOWS2 database.[1] Since all images in the set had the same size, the square root law did not applied here. By using LSB matching without matrix embedding (which means that the embedding efficiency was 2 bits per change), 9 sets of stego images with payloads

$$\mathcal{L}_{\text{LSB}} = \{0.10, 0.15, \ldots, 0.90\}$$

bits per pixel (bpp) were created.

All steganalyzers for LSB matching employed 2^{nd} order SPAM features with $T = 3$.[10]

4.2 Clairvoyant detector

For the scenario assuming Eve knowing the algorithm and the hidden message length (the optimal case for Eve), a separate detector for each payload out of $\mathcal{L}_{\text{nsF5}}/\mathcal{L}_{\text{pmk}}$ has been created. All 14 respectively 9 steganalyzers were implemented by soft-margin SVMs with Gaussian kernel. Their hyper-parameters were set as described in Appendix A and detection thresholds adjusted such that the false positive rate on testing images is 0.01 (see above). All detectors in both domains were trained on 2300 samples of cover images and 2300 samples of stego images.

[‡] Because the matrix embedding is turned off, the efficiency of the algorithm is 2 bits per change.

Although clairvoyant detectors are of limited practical use, they are valuable for benchmarking purposes. They enable to estimate the loss of accuracy of Eve's detection due to her absent knowledge about the details of the steganographic channel.

4.3 Targeted detector

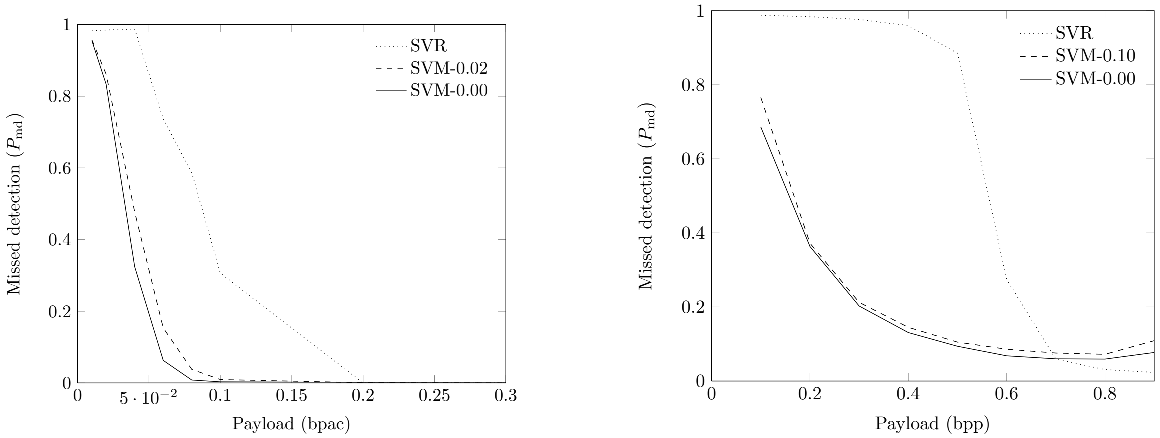
Section 3 presented two possible approaches to design a targeted detector: to create quantitative steganalyzer and threshold its output, or to convert the problem to classification. Because steganalyzers trained to detect low embedding rates do not generalize well on images with high payloads (see Figure 1), they are not considered as a solution.

The quantitative steganalyzers for nsF5 and LSB matching were implemented by Support Vector Regression (SVR) with Gaussian kernel and ϵ-insensitive loss (the methodology described in[13] has been followed). This configuration has three hyper-parameters, which were determined in similar manner as hyper-parameters of SVM (see Appendix B for details). Quantitative steganalyzers were trained on 4600 images with mixed payload in the range $[0, 0.90]$ for nsF5 and $[0, 1]$ for LSB matching algorithm. After the quantitative steganalyzers were trained, the thresholds on their output (estimated message length), from which images are deemed as stego were set such that the false positive rate on cover images from testing set was 0.01. In experiments presented here, the threshold was 0.078 bpac for steganalyzer of nsF5 and 0.46bpp for the steganalyzer of LSB matching. Notice the sharp contrast of the accuracy between estimators of payload of LSB matching (threshold was 0.46bpp) and nsF5 (threshold was 0.078bpac). This is due to higher sensitivity of PF274 features to embedding changes caused by nsF5 than of 2nd order SPAM features to changes caused by LSB matching.

In order to convert the one sided hypothesis test 2 to simple hypothesis test, two uniform distributions of payload for each case were used and compared. Steganalyzers of nsF5 were trained on uniform distributions $[0.02, 0.90]$ (denoted as SVM-0.02) and $(0.00, 0.90]$ (denoted as SVM-0.00), and steganalyzers of LSB matching were trained on uniform distributions $[0.1, 1]$ (denoted as SVM-0.10) and $(0, 1]$ (denoted as SVM-0.00). The rationale behind the first setting is that for practical purposes, it is impossible to detect stego images containing very small messages. By making a small margin between samples from both classes, the machine learning algorithm will not be confused by images with a very small message almost indistinguishable from cover images. The rationale behind the latter is that even the stego images with a fairly small message length might be useful to better determine the decision boundary between cover and stego images. As in previous cases, detection threshold of both classifiers were adjusted such that the false positive rate on testing set of both classifiers was 0.01.

Figure 3 shows the probability of missed detection (ratio of incorrectly classified stego images over all stego images) of proposed solutions empirically estimated from images in the testing set. The results reveal that the approach based on the quantitative steganalysis is inferior to both solutions based on the classification, where steganalyzers trained on samples with all message length dominates. The inferior performance of quantitative steganalyzer is because it solves more difficult problem (estimation) than it is actually needed (classification). Thus, the quantitative steganalyzer, despite its appealing features, is not the right solution here.

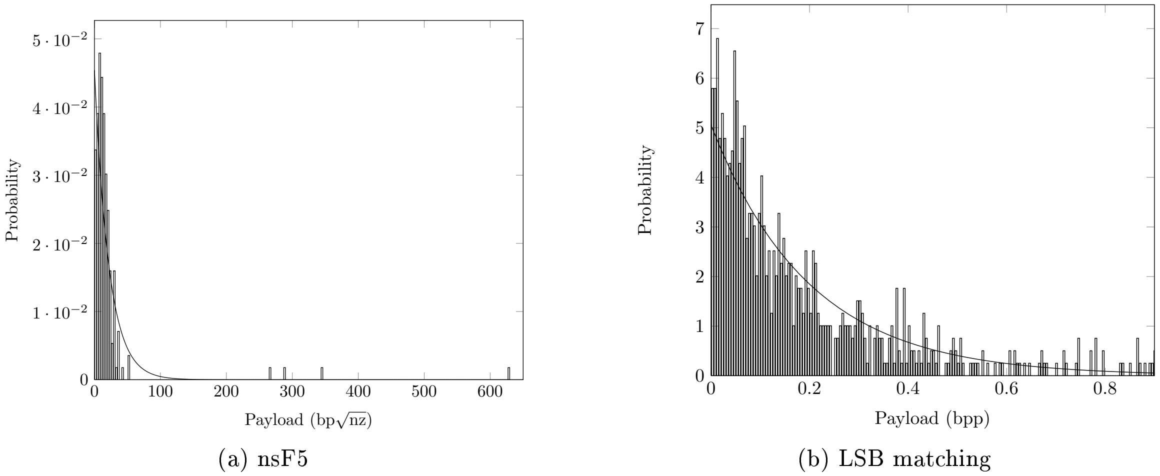
The distribution of payload in samples corresponding to support vectors in the steganalyzer trained on uniform distribution in the range $(0.00, 0.90]$ (nsF5) and $[0, 1]$(LSB matching) can reveal important information, which payloads in images from the training set are used. The support vectors are samples from the training set, which defines the decision boundary of the Support Vector Machine classifier. Consequently, if these samples are removed from training set, the decision boundary (and the classifier) changes. Contrary, if samples not corresponding to support vectors are removed from the training set, the decision boundary does not change and the solution remains the same. Thus, the distribution of payload among support vectors can show, how to better form a training set to improve the classification accuracy. Figure 4 shows this distribution of payload of support vectors in the steganalyzer of nsF5 (Figure 4 (a)), LSB matching (Figure 4(b)) trained on samples with uniform distribution in the range $(0, 0.9]$, $(0, 1]$ respectively. As expected, most support vectors correspond to samples with a very small payload, but there are also few samples with large one. Despite small in number, these samples are important, since they improve the accuracy of the classifier



(a) nsF5

(b) LSB matching

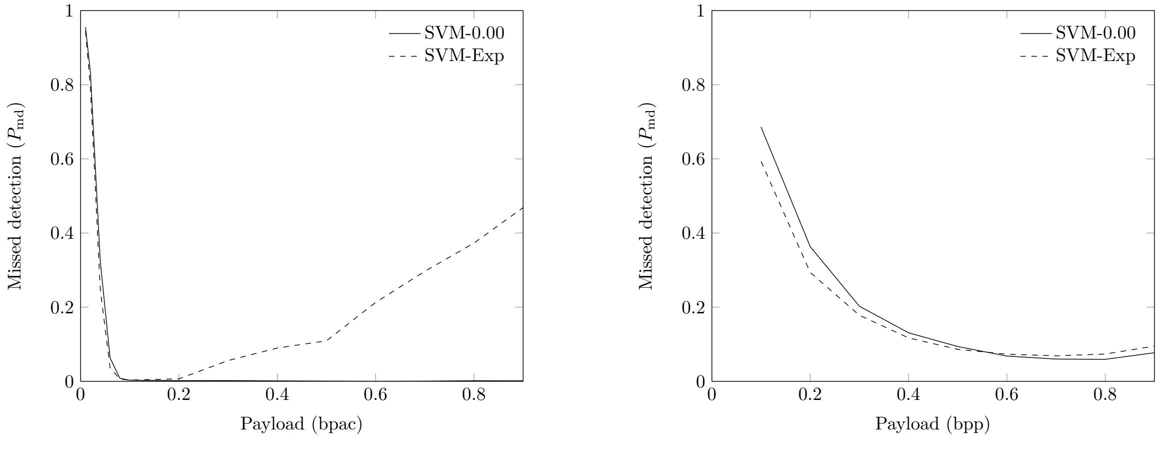
Figure 3: Probability of missed detection (1− probability of false negative) of the detectors of nsF5 (left) and LSB matching (right) algorithms developed to detect stego images with any payload. SVR corresponds to the solution based on Support Vector Regression, and SVM-0.00/SVM-0.02/SVM-0.1 corresponds to solution based on training SVM on stego images with uniformly distributed random payload in the range $(0.00, 0.9]/[0.02, 0.9]/[0.1, 0.9]$ respectively. The range of x-axis of the graph showing probability of missed detection of the steganalyzer of nsF5 is $[0, 0.4]$bpac, because on the higher payloads, all solutions provide near perfect detection and the differences are negligible.



(a) nsF5

(b) LSB matching

Figure 4: Left figure shows distribution of payload with respect to square root of non-zero AC coefficients in stego-samples corresponding to support vectors in SVM for nsF5 algorithms trained on stego-samples with uniform distribution of payload in the range $(0.00, 0.9]$. The right figure shows the same for the steganalyzer of the LSB matching trained on stego-samples with uniform distribution of payload in the range $(0, 1]$ with the caveat that the payload is measured in bits per pixel.



(a) nsF5

(b) LSB matching

Figure 5: Comparison of detection accuracy of classifiers trained on set with exponentially distributed payload in stego samples (SVM-Exp) with classifiers trained on a set with uniformly distributed payload in the range $(0, 0.9]/(0, 1]$respectively (SVM-0.00).

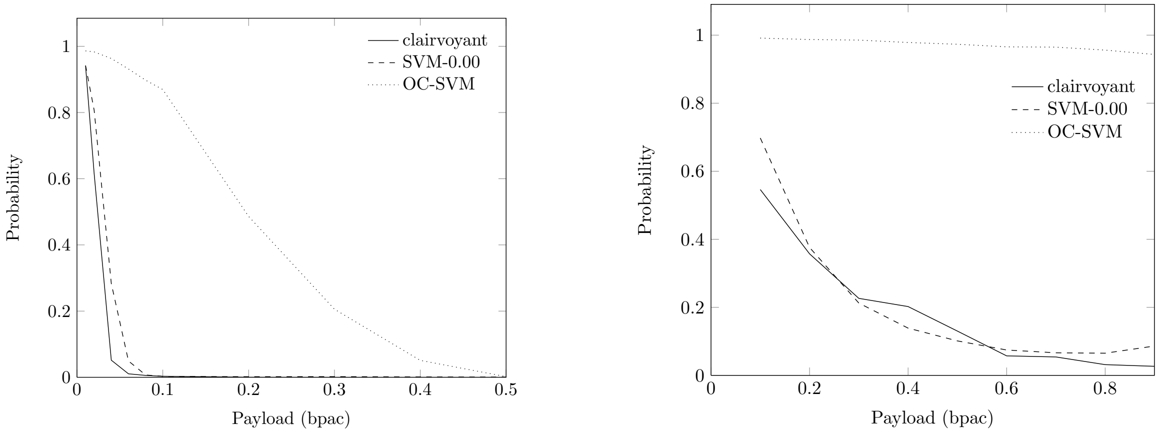
on the large payloads, where the steganalyzers trained on samples with small payload behaved poorly (see Figure 1). The exponential distribution fits the data with parameters $\mu = 0.19$ and $\lambda = 4.57 \cdot 10^{-2}$ for the steganalyzer of nsF5 and $\mu = 7.17 \cdot 10^{-4}$ and $\lambda = 5.03$ for the steganalyzer of LSB matching.

In order to observe the effect, how does the accuracy of the steganalyzer change when the classifier is trained on a set, where the payload in stego-samples follows estimated exponential distributions, SVM classifiers (same methodology as above) on training sets with this property have been trained. The comparison of the detection accuracy of these classifiers (denoted as SVM-Exp) with the best solution founded so far (SVM-0.00) is shown on Figure 5. It is rather surprising that the performance of classifiers trained on training set with stego-images with exponential distribution of payload is inferior to solutions trained on stego-images with uniform distribution of the paylaod. Although the SVM-Exp steganalyzers have slightly better detection accuracy on images with low payload, the detection accuracy on stego-images with high payload is significantly worse. For example the detection accuracy of the steganalyzer of nsF5 algorithm starts to decrease, as the payload of stego-images exceeds 0.2bpac. After examining carefully the training set of this steganalyzer, it have been found that samples of stego-images with high payload are almost missing, thus the worst accuracy is not surprise. Taking into consideration that complexity of modern implementations of SVM depends on number of support vectors and not on the total number of training samples, we believe that it is better to increase number of stego samples and use uniform distribution of the payload then search for optimal disribution of payload in training stego-images.

4.4 Effects of Eve knowledge

In order to observe effects of Eve's knowledge on the accuracy of her detection, universal steganalyzers by means of one-class SVMs were created. To keep the total number of samples used for training of all classifiers the same, the universal steganalyzers were trained on 4600 samples of cover images. The hyper-parameters were set as described in Appendix C. Because the used implementation from libsvm[3] does not allow to precisely control the false positive rate of the universal steganalyzer, the false positive rates of clairvoyant and targeted steganalyzers (SVM-0.00) were adjusted, such that all three steganalyzers had the same false positive rate. This means that in this subsection, steganalyzers of nsF5/LSB matching have false positive rate on the testing set 1.24% (steganalyzers of nsF5) and 0.86% (steganalyzers of LSB matching).

Figure 6 shows the probability of missed detection of clairvoyant, targeted (SVM-0.00), and universal detector. As expected, the clairvoyant detector is the best, but the accuracy of targeted detector is fairly close. The highest drop in accuracy is at the payload 0.04bpac (nsF5) / 0.1bpp (LSB matching), where the probability of missed detection increased from 5% to 28% (nsF5) / from 54% to 70% (LSB matching).



(a) nsF5 (b) LSB matching

Figure 6: Probability of missed detection of the clairvoyant, targeted and universal steganalyzer of nsF5 (left) and LSB matching (right) algorithms. The false positive rate of nsF5 / LSB matching steganalyzers was fixed to 1.24%, / 0.86% respectively.

Naturally, the accuracy of universal steganalyzer is the worst by a huge margin. This is not a surprise, since as explained in,[12] the universal steganalysis is generally a very difficult task.

It is interesting to compare all three scenarios from the point of view of Alice and Bob. Let's assume that Alice and Bob plan to use nsF5 algorithms with JPEG images as carriers. They would like the probability of missed detection of Eve's detector to be higher than 80% at a false alarm rate 1.24%. If Eve knows all the details about the steganographic channel, they can communicate at rate up to $9.478\mathrm{bp}\sqrt{\mathrm{ac}}$ (0.014bpac). If Eve does not know the length of hidden message, the capacity of the channel increases only by 43% to $13.54\mathrm{bp}\sqrt{\mathrm{ac}}$ (0.020bpac). Finally, if Eve does not know anything about the channel, the capacity increases by 685% to $74.47\mathrm{bp}\sqrt{\mathrm{ac}}$ (0.11 bpac). The problem is that *Alice and Bob does not know, what Eve knows*

5. CONCLUSION

This main focus of this work was exploring the possibilities to build a steganalyzer based on a combination of machine learning algorithm and steganalytic features capable of accurate detection of stego images with different payload. To motivate the problem, it has been demonstrated that the discrepancy between payload in training and application (evaluation) stego images can significantly decrease the accuracy of the steganalysis.

To resolve this problem, two fundamentally different approaches were proposed. The first approach is based on quantitative steganalyzer, the second is based on transforming one-sided hypothesis test to simple hypothesis test, which is solved by Support Vector Machines. The transformation of hypothesis tests is done by assuming uniform distribution of payload in stego-images.

In experimental section, both approaches were compared on steganalysis of F5 algorithm with shrinkage removed by wet paper codes for JPEG images and LSB matching for raw (uncompressed) images. The experiments showed that the solution based a transformation to simple hypothesis test solved by SVM to be superior.

The experimental section also demonstrates, how the accuracy of steganalysis depends on Eve's knowledge about details of steganographic channel. According to the results, the absence of the knowledge about payload in images can be overcome by the proper creation of the steganalyzer, but the absent knowledge of used steganographic algorithm forces Eve to use of universal steganalyzer, which has serious impact on accuracy of her steganalysis. The contrast between accuracy of targeted and universal steganalysis should motivate the further research in the area of universal steganalysis.

Acknowledgements

This material is based upon work supported by the ITC-A of the US Army under Contract No. W911NF-10-1-0070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ITC-A of the US Army. Effort is also sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-10-1-3016. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon. The work of Tomáš Pevný is also supported by Czech Ministry of Education grants 6840770038 and ME10051: AMVIS-AnomalyNET.

REFERENCES

[1] P. Bas and T. Furon. BOWS–2. http://bows2.gipsa-lab.inpg.fr, July 2007.

[2] G. Cancelli, G. Doërr, I. Cox, and M. Barni. A comparative study of ±1 steganalyzers. In *Proceedings IEEE, International Workshop on Multimedia Signal Processing*, pages 791–794, Queensland, Australia, October 2008.

[3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] T. Filler, A. D. Ker, and J. Fridrich. The Square Root Law of steganographic capacity for Markov covers. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 08 1–08 11, San Jose, CA, January 18–21, 2009.

[5] C. Hsu, C. Chang, and C. Lin. *A Practical Guide to ± Support Vector Classification.* Department of Computer Science and Information Engineering, National Taiwan University, Taiwan.

[6] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. 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 5 1–5 17, San Jose, CA, January 27–31, 2008.

[7] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[8] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, Princeton, NJ, September 7–8, 2009.

[9] 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.

[10] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by substractive pixel adjacency matrix. *Transactions on Information Forensics and Security*, 5:215–224, 2010.

[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. Novelty detection in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 167–176, Oxford, UK, September 22–23, 2008.

[13] T. Pevný, J. Fridrich, and A. D. Ker. From blind to quantitative steganalysis. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 0C 1–0C 14, San Jose, CA, January 18–21, 2009.

[14] 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.

APPENDIX A. SETTING OF HYPER-PARAMETERS OF SVM

Before training the SVM, the value of the penalization parameter C and the kernel parameters (in our case γ) need to be set. The values should be chosen such that the classifier trained with these values will have a good generalization. The standard approach is to estimate the error on unknown samples by cross-validation on the training set on a fixed grid of values, and then select the value corresponding to the lowest error (see[5] for details). The approach utilized in this paper was a slight modification. It first examined the multiplicative grid:

$$\begin{aligned} C &\in \{0.001, 0.01, \ldots, 10000\}. \\ \gamma &\in \{2^i | i \in \{-d-3, \ldots, -d+3\}, \end{aligned}$$

where d is number of features in the subset. Then, it checked, if the (C', γ') point corresponding to least estimated error is on the boundary of already explored set. If so, points in neighborhood $(C, \gamma) \in \{(k \cdot C', 2^i \cdot \gamma' | k \in \{0.1, 1, 10\}, i \in \{-1, 0, 1\}\}$ were explored and added to the set of already explored set. The algorithm kept doing so, until the point with least estimated error was inside explored set. The hyper-parameters corresponding to the point with least estimated error were then used to train the SVM.

The idea behind the two-phase search is to ensure that the point with the least estimated generalization error is not the boundary point of the explored set. Under the assumption that the generalization error surface is convex, which generally holds for vast majority of practical problems, this algorithm keeps the number of explored points relatively low, while returning a suitable set of hyper-parameters.

APPENDIX B. SETTING OF HYPER-PARAMETERS OF SVR

There are three hyper-parameters that need to be set prior to training of Support Vector Regression: the penalization parameter C, the width of the Gaussian kernel γ, and the insensitivity of the loss function ϵ. The choice of the hyper-parameters has a significant influence on the ability of the estimator to generalize (to accurately estimate the change rate on samples not in the training set). Since there is no optimal method to set them, an error by means of five-fold cross-validation have been estimated on predefined set of triplets (C, γ, ϵ). To decrease the computational complexity, the search consisted from two phases.

In the first phase, the parameters were estimated by five-fold cross-validation on the following grid

$$\begin{aligned} (C, \gamma, \epsilon) \quad \in \mathcal{S}_1 = \quad & \big\{ \big(10^i, 2^j, 0.005 \cdot k\big) \mid \\ & i \in \{-3, \ldots 4\}, \\ & j \in \{-11, \ldots, -5\}, \\ & k \in \{1, 2, 3, 4\} \big\} . \end{aligned}$$

The triplet $(C_1, \gamma_1, \epsilon_1)$ with the least estimated generalization error on $\mathcal{S}_1$ was used to seed the search in the second phase. The search in the second phase was performed on the grid

$$\mathcal{S}_2 = \big\{ \big(10^i, 2^j, 0.005 \cdot k\big) \mid i, j \in \mathbb{Z}, k \in \mathbb{N} \big\} .$$

In each iteration, the point with the least generalization error (again estimated by five-fold cross-validation) was checked whether it lay on the grid boundary. If so, the error was estimated on the neighboring points from the set $\mathcal{S}_2$ and the check was repeated. If not, the search was stopped and the triplet (C, γ, ϵ) with the least estimated generalization error was used for training.

APPENDIX C. SETTING OF HYPER-PARAMETERS OF ONE-CLASS SVM

The setting of a hyper-parameters, the penalization parameter ν and the kernel parameters (in our case γ), of One-Class SVM (OC-SVM) is difficult, because only samples from one class (in our case cover samples) are available. This paper used following heuristic. First, the false positive rate of the classifiers with hyper-parameters from a grid $(\nu, \gamma) \in \{(0.005 \cdot k, 2^i) | k \in \{1, 2, \ldots, 40\}, i \in \{-19, -18, \ldots, 3\}\}$ have been estimated (by using 5-fold cross-validation). The γ and ν used for training correspond to the point with smallest γ and ν, such that the estimated false positive rate was below 1%. The rationale behind this was to use the simplest solution (small γ and ν) with desired properties, since simple solutions generalize better than the complex ones.