STEGANALYSIS WITH COVER-SOURCE MISMATCH AND A SMALL LEARNING DATABASE

Jérôme PASQUET[2,3], *Sandra BRINGAY*[2,3,4], *Marc CHAUMONT*[1,2,3]

[1] UNIVERSITE DE NIMES, F-30021 Nîmes Cedex 1, France

[2] UNIVERSITE MONTPELLIER 2, UMR5506-LIRMM, F-34095 Montpellier Cedex 5, France

[3] CNRS, UMR5506-LIRMM, F-34392 Montpellier Cedex 5, France

[4] AMIS, UNIVERSITE MONTPELLIER 3, Route de Mende 34199 Montpellier Cedex 5, France

{jerome.pasquet, sandra.bringay, marc.chaumont}@lirmm.fr

ABSTRACT

Many different hypotheses may be chosen for modeling a steganography/steganalysis problem. In this paper, we look closer into the case in which Eve, the steganalyst, has partial or erroneous knowledge of the cover distribution. More precisely we suppose that Eve knows the algorithms and the payload size that has been used by Alice, the steganographer, but she ignores the images distribution. In this source-cover mismatch scenario, we demonstrate that an Ensemble Classifier with Features Selection (EC-FS) allows the steganalyst to obtain the best state-of-the-art performances, while requiring 100 times smaller training database compared to the previous state-of-the art approach. Moreover, we propose the *islet approach* in order to increase the classification performances.

Index Terms— Steganalysis, Cover-Source Mismatch, Ensemble Classifiers with Post-Selection of Features, Ensemble Average Perceptron, Clustering.

1. INTRODUCTION

During the BOSS[1] competition [2], the effects of *cover-source mismatch* were clearly observed.

A set of cover and stego image couples (18 000 images) were given to the steganalysts. The images were uncompressed 512×512 grey-level images from 7 different cameras. All the steganalysts were then given a test set of 1000 images used by the organizers to evaluate the detection capability of each competing steganalyst. Some of the images of the test set were from a camera that was not used in the learning set. Thus, the steganalysts encountered inconsistency between the image model, learned during the learning step, and the image model of the test set. This inconsistency is called the *cover-source mismatch* [3].

[1]BOSS (Break Our Steganography System) was the first steganalysis challenge. The challenge started on the 9th of September 2010 and ended on the 10th of January 2011. The goal of the player was to figure out which images contained a hidden message and which images did not. http://www.agents.cz/boss/BOSSFinal/. The steganographic algorithm was HUGO [1].

The *cover-source mismatch* phenomenon was initially reported in [4], but the only solution to manage image diversity was proposed in 2012 by Lubenko and Ker [5, 6]. Their hypothesis is that in order to have a sufficiently descriptive model of the image, one should work on very a huge variety of images. Andrew Ker says "Google knows the model since it owns all the images". Their approach was then to use millions of images during the learning step. Evidently that kind of approach may be very time consuming. Thus, they decided to choose a steganalyzer named Ensemble Average Perceptron (EAP) [5] that has linear complexity. They downloaded millions of JPEG images, with a quality factor of 85, from a social network, then they embedded a message with nsF5 at 0.05 bits per non-zero DCT.

In scenarios with the largest diversity (i.e. with many different image sources), they showed that the EAP [5] gave the best testing accuracy (85.1%). The Ensemble Classifier (EC) [7] testing accuracy was 83.6%, and the Support Vector Machine (SVM) [8] testing accuracy was 80.9%. For complexity reasons, the size of the training set was not equal for each classfier: 6 000 images for SVM, 20 000 images for EC, and 1 000 000 images for EAP. In those conditions, the experimental comparison seemed to show that the EAP was better than the EC when there was *cover-source mismatch* [5].

In this paper, we refute the hypothesis that millions of images are necessary, and we show experimentally that EC with post-features selection (EC-FS) [9] allows us to obtain better results with 100 times smaller training database. Moreover, we also introduce an additional pre-processing that really overcomes the problem of *cover-source mismatch*. This pre-processing consists in organizing images in clusters, and associating a steganalyzer with each cluster, which thus reduces the diversity inside each cluster. We call this proposition the *islet approach*.

In section 2, we recall the principle of the EC-FS [9] and EAP [5] classifiers, that were used to carry out the experiments when there is a huge variety of images. The experimental results validate our intuition and show that the selection of

features is a good tool in order to deal with *cover-source mismatch* phenomenon. We consider this as a contribution for the domain since it gives clues to manage the *cover-source mismatch* phenomenon.

In section 3, we present the islet concept. This is our second contribution, and this confirms something that has already been observed : the increase of the similarity between the learning set and the testing set implies an increase of the steganalysis performances [5].

In section 4, we give the experimental results and analyze them.

2. EC-FS AND EAP CLASSIFIERS

The classifier learning phase is performed on a database of size N cover and stego images. This database is represented by a set of couples (feature vector, class number). We note the set $\mathcal{B} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{i=N}$, with $\mathbf{x}_i \in \mathbb{R}^d$ being a vector of dimension d characterizing the i^{th} image, and $y_i \in \{0, 1\}$ the associated class number (-1 for a cover image and $+1$ for a stego image).

In the next two subsections, we present the Ensemble Classifier with Post-Selection of Features [9], denoted EC-FS, and the Ensemble Average Perceptron [5], denoted EAP.

2.1. Ensemble Classifier with Post Selection of Features (EC-FS)

The Ensemble Classifier with Post-Selection of Features (EC-FS) was presented at IEEE ICIP'2012 [9]. It is an extension of the EC [7].

The EC [7] is made of a set of L weak classifiers. During the learning step, each weak classifier learns separately on the same image database. A weak classifier, denoted h_l with $l \in \{1, .., L\}$, takes the same $\mathbf{x} \in \mathbb{R}^d$ feature vector as input and returns a class number (-1 for a cover image, and $+1$ for a stego image):

$$\begin{array}{rcl} h_l : \mathbb{R}^d & \rightarrow & \{-1, +1\} \\ \mathbf{x} & \rightarrow & h_l(\mathbf{x}) \end{array} \quad (1)$$

Each weak classifier performs its learning in a space of d_{red} dimension, with $d_{red} \ll d$. In practice, each weak classifier pseudo-randomly selects features from the feature vector of dimension d. The merging of the votes of the weak classifiers is then obtained by a majority vote, such that, for a $\mathbf{x} \in \mathbb{R}^d$ feature vector, we have:

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} h_l(\mathbf{x}) \leq 0, \\ 1 \text{ otherwise.} \end{cases} \quad (2)$$

The EC with Post Selection of Features (EC-FS) compared to EC, reduces d_{red} dimensions and also removes features that could disrupt classification process.

The idea of EC-FS is to improve the performance of each weak classifier through the selection of features. In addition, the selection of features adds an additional variability to the EC algorithm because each weak classifier selects a different number of features (and not always d_{red} features). This additional variability enhances the classification model and leads to improved performance. To keep the complexity equivalent to the EC's one, we apply a selection process **after** the learning step and we do not re-run any learning step. Thus, once a weak classifier learned, it will seek to take away some features in order to reduce its probability of error.

Each weak classifier performs its learning in a space of d_{red} dimension, with $d_{red} \ll d$. After the learning phase of a weak classifier, only a subset of the features set is kept. Five low complexity metrics evaluating the importance of a feature have been proposed in [9], and this leads to an order for the selection of features leading to the smallest probability of error.

In the article [9], we report an average gain on the recall of 1.7% in a clairvoyant scenario [10] (cover distribution, stego ditribution, and the payload size of the message are known) without *cover-source mismatch*, using BossBase v1.00 (http://www.agents.cz/boss/BOSSFinal/), and the HUGO algorithm [1] at 0.4 bpp for the embedding. Because of the difficulty of scrounging percentages during the BOSS competition [3], this gain is significant. We thus decided to test how the post-selection of features may help in the case of a high diversity of images, and thus some source-cover mismatch phenomena.

2.2. Ensemble Average Perceptron (EAP)

The Ensemble Average Perceptron (EAP) is a classifier built with a set of L weak classifiers [5]. The EAP is constructed exactly like the Ensemble Classifier [7] explained in section 2.1 (see Equ. 1 and Equ. 2).

Each weak classifier is an average perceptron [5], h_l with $l \in \{1, .., L\}$, defined such as:

$$\begin{array}{rcl} h_l : \mathbb{R}^d & \rightarrow & \{-1, +1\} \\ \mathbf{x} & \rightarrow & h_l(\mathbf{x}) = sign(\mathbf{w}^{avg}.\mathbf{x}) \end{array} \quad (3)$$

with $\mathbf{x} \in \mathbb{R}^d$ being a feature vector, $sign$ the function returning -1 or $+1$ depending on the sign of the input value, and $\mathbf{w}^{avg}$ a vector defining the separating plane of the two classes (-1 for cover, and $+1$ for stego):

$$\mathbf{w}^{avg} = \frac{\mathbf{w}^{sum}}{N} \quad (4)$$

with $N \in \mathbb{N}$ being the number of images used for the learning step, and $\mathbf{w}^{sum}$ the sum of the successive weight vectors $\mathbf{w}^{(i)}$:

$$\mathbf{w}^{sum} = \sum_{i=1}^{N} \mathbf{w}^{(i)} \quad (5)$$

During the learning phase, for an incoming feature vector $\mathbf{x}_i$ with a class number $y_i \in \{-1, +1\}$, the weight vector $\mathbf{w}^{(i)}$

is updated such that:

$$\mathbf{w}^{(i)} = \begin{cases} \mathbf{w}^{(i-1)} & if\ y_i = sign(\mathbf{w}^{avg}.\mathbf{x}_i) \\ \mathbf{w}^{(i-1)} + y_i.\mathbf{x}_i & if\ y_i \neq sign(\mathbf{w}^{avg}.\mathbf{x}_i) \end{cases} \quad (6)$$

3. ISLET PARTITIONING APPROACH

In a real world scenario, the image cover model is not known by the steganalyst. The diversity of the images is due to the lossy or lossless formats, the compression rates, the way images are generated (synthetics, scan, digital, numerical photos), the type of scene, luminosity, focus, etc. A classifier will manage this diversity more easily, and thus the *cover-source mismatch*, if we restrict its learning and classification to a set of "homogeneous" images. By "homogeneous" we mean images that have close feature vectors.

Our proposition is thus to apply a pre-processing to the image database in order to partition it into a few clusters. Then, we associate a classifier, i.e. EC-FS or EAP, to each cluster, which will learn and classify only vectors that belong to the cluster. We named this technique the *islet* approach.

The *learning step* consists of two stages. The first stage consists in running a k-means algorithm on a subset of the entire training database. The k-means algorithm is achieved on feature vectors representing the images. We are obtaining a set of K *means vectors* noted $\{\mu_k\}_{k=1}^{k=K}$.

The second step consists in creating K classifiers EC-FS or EAP. Conceptually, a classifier is associated to a *mean vector*. There is thus one classifier per cluster. In this second step, we rescan the entire training database, and for each feature vector $\mathbf{x}_i$, we select the closest cluster, i.e. we select the cluster numbered k, owning the smallest L2 distance between $\mathbf{x}_i$ and $\mu_{k,\ with\ k\in\{1,\ldots,K\}}$, and then we pass this $\mathbf{x}_i$ feature vector to the k^{th} classifier so that it can learned.

The *classification step* involves one stage. Given a feature vector $\mathbf{x}_i$ to be classified, we first select the closest cluster, i.e. we select the cluster numbered k, owning the smallest L2 distance between $\mathbf{x}_i$ and $\mu_{k,\ with\ k\in\{1,\ldots,K\}}$. Second, we pass this $\mathbf{x}_i$ feature vector to the k^{th} classifier, for its classification into -1 for cover, and $+1$ for stego.

Given an input image, the classifier of which learning has been achieved on images with "similar" feature values will be less sensitive to the *cover-source mismatch* problem.

4. RESULTS

The database is obtained by downloading 1 million color images from the TwitPic website[2] mostly in jpeg format but also in uncompressed format. Then, images are decompressed, transformed in grey-levels images, cropped to 450×450, and a spatial embedding with the HUGO [1] algorithm at 0.35 bits per pixel is achieved. This leads to a database of 2 million images. Various payload are embedded leading to a database of 3.8 million pairs of images.

For the EAP learning, once the learning is completed, we re-run the learning step, which "virtually" leads to a learning on 7 million pairs. For each experiment, three simulations are conducted, and the database images are considered in a different order. The probability of error, $P_E = \frac{P_{FA}+P_{MD}}{2}$, is obtained on a database of 40 000 images that do not belong to the learning database[3]. Let us note that the test database has been downloaded from TwitPic at a different date of the training database. This implies that the training set is made of images of which sources are different from those of the learning data-base. The *cover-source mismatch* phenomenon is thus present in our experiment. Finally, the average probability of error, $\overline{P_E}$, is computed by averaging the probability of error of the three simulations. In our experiment we report the prediction rate = $1 - \overline{P_E}$.

On this large database, we evaluated the EC [7], EC-FS [9], and EAP [5]. We used personal C++ implementations, and we set parameters L and d_{red} at the same values as [5,6], i.e. $L = 100$ and $d_{red} = 2000$.

The feature vectors are rich models vectors for spatial images, of dimension 34 671, and described in [11]. Those vectors were extracted with a personal C++ implementation. All those vectors were normalized using the maximum norm which showed an increase in the classification performance of more than 8% compared to the variance norm.

All the experiments were carried out at the Center of High Performance Computing HPC@LR[4] which provided access to parallel programming, and 24 GB of RAM per node. The HPC@LR requires C++ implementations.

4.1. Comparison between EC, EC-FS and EAP

The EC and EC-FS computational complexity is $O(d_{red}^2.N.L)$ and it is higher than the EAP which is $O(d_{red}.N.L)$. The square factor in the complexity of EC and EC-FS leads to such a huge computational time that it is impossible to train and learn on the entire database (d_{red} has been fixed to 2000). We then tested EC and EC-FS with 20 000, 30 000, 50 000, 100 000 and 150 000 learning images (see Table 1). Moreover, the learning step for EAP was achieved on the entire database ($2 \times 3.8 \approx 7$ millions of images).

Table 1 and Figure 1 reveal that the EAP converge around 93%. This value exceeded the maximum value of 83% of the

[2]http://twitpic.com

[3]There is no any automatic adjustment for EC and EAP. This adjustment is useless in the *cover-source mismatch* scenario for EC [5], and not adapted to the online behavior of EAP [5]. Nevertheless, in EC-FS [9], the features are selected in order to minimize the probability.

[4]HPC@LR (High Performance Computing at Languedoc Roussillon - France) gives access to a 15-Teraflop hybrid hardware platforms consisting of 84 nodes of Intel Hexacore Dual-processors, 2 SPM servers with 80 cores, CPU/GPU Servers Nodes, 4 Cell Dual-processor Nodes, 1 Power 7 Node, Infiniband QDR Network, and Disk Storage.

training size	20 000	30 000	50 000	100 000	150 000
PR* EC	81%	83%	84%	82%	**50%**
PR* EC-FS	92%	94%	95%	95%	95%

*PR = prediction rate

Table 1. Comparison between EC and EC-FS.

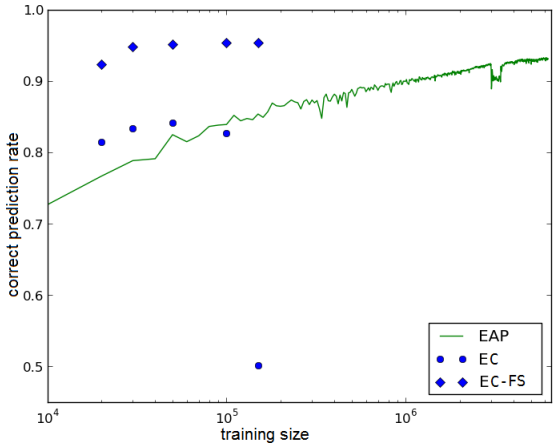

Fig. 1. Results of EC, EC-FS, and EAP, with a logarithmic scale on abscissa.

EC. Moreover, the EC performance fell when the number of learning samples exceeded 50 000. This counter-performance may be explained by the too strong heterogeneity and thus a strong *cover-source mismatch* phenomenon. A high value for d_{red} implies a counter-performance. Indeed, a linear classification is no more efficient when $d_{red} = 2000$ and when the features are extremely diverse and probably noisy. Our intuition is that lots of features should not be grouped together, or should be selected for producing a *weak classifier* insensitive to the *cover-source mismatch*. This also explains why the EC-FS approach is more efficient.

The behavior of the EC-FS was completely different. We did not observe any performance collapse. Moreover, the EC-FS was more efficient than EC and EAP. Indeed, EC-FS obtained a performance of 95%, which was 2.3% higher than EAP (93%). Moreover, EC-FS converged using only 50 000 to 100 000 learning images. EC-FS required 100 times fewer images (50 000 images) than EAP (5 000 000 images) to give better results.

This result is very promising and encourages future works on feature selection, feature reduction, feature combination, etc, in the clairvoyant scenario or in a *cover-source mismatch* scenario. Our intuition is that the features selection acts as a denoising of the features space, or said differently the features selection creates an invariant space which is invariant to the cover-source variations, but sensitive to a message embedding. In that sense, the EC-FS approach shares some similarity to what has been proposed in [12], with a totally different technique.

In addition, experiments revealed that a correct cover model may be extracted without a million images. The approach is thus a more practical solution than those of Lubenko and Ker that require more than a million images [5, 6].

In this section, we compare two very efficient classifiers for the *cover-source mismatch* scenario. EC-FS gave better results than EAP with 100 times fewer images. In the next section, we evaluate islet partitioning to increase the classification performances and overcome the source-cover mismatch phenomenon.

4.2. Islets experiments

4.2.1. Islet parameters

In the *islet approach*, we have to decide on the number K of islet (K vectors $\{\mu_k\}_{k=1}^{k=K}$, and K classifiers). Increasing the number of clusters is very beneficial to overcome the cover source mismatch problem. Indeed, the higher the islet K number, the closer the image to be classified will be from a center μ_k, and the more the associated classifier will be adapted to this image since, during its learning, this classifier would have learned with "similar" images. However, increasing K will necessitate a larger database since, on average, the number of learning images in each islet is N/K with N being the number images in the entire database and K the number of clusters. In order to converge, the cluster may require a sufficient number of images (50 000 to 150 000 for EC-FS, and 10^6 to 10.10^6 for EAP). This multiplicative factor (K) on the database size necessitates using specific architectures for experiments, such as the High Performance Computing Center of the Languedoc Roussillon.

4.2.2. Islets with EC-FS

In this section, we combine islets with EC-FS. The purpose is to obtain better results for the prediction rate than the 95% obtained at the convergence with around 50 000 images. Let us note that, practically, EC-FS is a good candidate for the *islet approach* since it converges rapidly. In the experiments, a k-mean was achieved on 80 000 images.

K islets	Training size per islet	Prediction rate
1	150 000	95.39%
2	75 000	95.81% (+0.41%)
3	50 000	95.83% (+0.43%)
4	37 500	95.82% (+0.43%)
5	30 000	95.88% (+0.49%)
6	25 000	**96.06%** (**+0.67%**)
7	21 428	95.72% (+0.33%)

Table 2. Results of islets with EC-FS.

Table 2 gives the prediction rate as a function of the number K of islets for 150 000 learning images. The performance

has maximum for 6 islets with a prediction of 96%. When the number of islets is low, the number of images per islet is high, but this is useless because the EC-FS converges at around 50 000 images. Conversely, when the number of islets is too high, there are not enough training images, the classifiers have not yet converged, and thus the performance drops.

Partitioning of the learning database via the islets creates areas which are more homogeneous. On these areas, the EC-FS converge quickly and when the tested image is almost similar to the learning images, the performance of EC-FS becomes better.

With those preliminary results, we show that the gain obtained by islets is 0.67% which is not negligible and very promising. Indeed, islets further increase the performance of EC-FS.

In conclusion, EC-FS gives better results than EAP (gain of 2.3% in the prediction rate), requires 100 times fewer images and when it is combined with the *islet approach*, it increases the gain by about 0.7% in order to obtain a final prediction rate of 96%.

5. CONCLUSION

In this article, we first show that EC-FS is a very efficient tool for managing very heterogeneous data. It has good performance to limit the *cover-source mismatch* problem and requires a learning set 100 times smaller than EAP. Indeed, with 150 000 images, the method is 2.3% more efficient than EAP, which requieres more than one million images. Secondly, by combining EC-FS with the *islet approach*, we obtain a total gain of 3% compared to EAP approach. The selection of features and the partitioning are thus very promising techniques to overcome the problem of heterogeneous data, also known as the *cover-source mismatch* problem.

REFERENCES

[1] T. Pevný, T. Filler, and P. Bas, "Using High-Dimensional Image Models to Perform Highly Undetectable Steganography," in *Information Hiding, 12th International Conference, IH'2010*, Calgary, Alberta, Canada, June 2010, vol. 6387 of *Lecture Notes in Computer Science*, pp. 161–177, Springer.

[2] P. Bas, T. Filler, and T. Pevný, "'Break Our Steganographic System': The Ins and Outs of Organizing BOSS," in *Information Hiding, 13th International Conference, IH'2011*, Prague, Czech Republic, May 2011, vol. 6958 of *Lecture Notes in Computer Science*, pp. 59–70, Springer.

[3] J. Fridrich, J Kodovský, V. Holub, and M. Goljan, "Breaking HUGO - The Process Discovery," in *Information Hiding, 13th International Conference, IH'2011*, Prague, Czech Republic, May 2011, vol. 6958 of *Lecture Notes in Computer Science*, pp. 85–101, Springer.

[4] G. Cancelli, G. J. Doërr, M. Barni, and I. J. Cox, "A comparative study of +/-1 steganalyzers," in *Workshop Multimedia Signal Processing, MMSP'2008*, 2008, pp. 791–796.

[5] I. Lubenko and A. D. Ker, "Steganalysis with mismatched covers: do simple classifiers help?," in *Multimedia and Security Workshop, MM&Sec'2008, Proceedings of the 14th ACM multimedia*, Coventry, UK, Sept. 2012, MM&Sec'2012, pp. 11–18.

[6] I. Lubenko and A. D. Ker, "Going from small to large data in steganalysis," in *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 22th Annual Symposium on Electronic Imaging, SPIE'2012*, San Francisco, California, USA, Feb. 2012, vol. 8303.

[7] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[8] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[9] M. Chaumont and S. Kouider, "Steganalysis by ensemble classifiers with boosting by regression, and post-selection of features," in *IEEE International Conference on Image Processing, ICIP'2012*, Lake Buena Vista (suburb of Orlando), Florida, USA, Sept. 2012, pp. 1133–1136.

[10] T. Pevný, "Detecting messages of unknown length," in *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, San Francisco, California, USA, Feb. 2011, vol. 7880.

[11] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 3, pp. 868–882, 2012.

[12] T. Pevný and A. D. Ker, "The challenges of rich features in universal steganalysis," in *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 23th Annual Symposium on Electronic Imaging, SPIE'2013*, San Francisco, California, USA, Feb. 2013, vol. 8665.