Publication E

Yoan Miche, Patrick Bas, Amaury Lendasse, Christian Jutten, and Olli Simula. 2009. Reliable steganalysis using a minimum set of samples and features. EURASIP Journal on Information Security, volume 2009, article ID 901381, 13 pages.

Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2009, Article ID 901381, 13 pages
doi:10.1155/2009/901381

*Research Article*

# Reliable Steganalysis Using a Minimum Set of Samples and Features

**Yoan Miche,[1,2] Patrick Bas,[2] Amaury Lendasse,[1] Christian Jutten (EURASIP Member),[2] and Olli Simula[1]**

[1] *Laboratory of Information and Computer Science, Helsinki University of Technology, P.O. Box 5400, FI-02015 HUT, Finland*
[2] *GIPSA-Lab, 961 rue de la Houille Blanche, BP 46, F-38402 Grenoble Cedex, France*

Correspondence should be addressed to Yoan Miche, ymiche@cc.hut.fi

Received 1 August 2008; Revised 14 November 2008; Accepted 13 March 2009

Recommended by Miroslav Goljan

This paper proposes to determine a sufficient number of images for reliable classification and to use feature selection to select most relevant features for achieving reliable steganalysis. First dimensionality issues in the context of classification are outlined, and the impact of the different parameters of a steganalysis scheme (the number of samples, the number of features, the steganography method, and the embedding rate) is studied. On one hand, it is shown that, using Bootstrap simulations, the standard deviation of the classification results can be very important if too small training sets are used; moreover a minimum of 5000 images is needed in order to perform reliable steganalysis. On the other hand, we show how the feature selection process using the OP-ELM classifier enables both to reduce the dimensionality of the data and to highlight weaknesses and advantages of the six most popular steganographic algorithms.



## 1. Introduction

Steganography has been known and used for a very long time, as a way to exchange information in an unnoticeable manner between parties, by embedding it in another, apparently innocuous, document.

Nowadays steganographic techniques are mostly used on digital content. The online newspaper Wired News reported in one of its articles [1] on steganography that several steganographic contents have been found on web sites with very large image database such as eBay. Provos and Honeyman [2] have somewhat refuted these facts by analyzing and classifying two million images from eBay and one million from USENet network and not finding any steganographic content embedded in these images. This could be due to many reasons, such as very low payloads, making the steganographic images less detectable to steganalysis and hence more secure.

In practice the concept of security for steganography is difficult to define, but Cachin in [3] mentions a theoretic way to do so, based on the Kullback-Leibler divergence. A stego process is thus defined as $\epsilon$-secure if the Kullback-Leibler divergence $\delta$ between the probability density functions of the cover document $p_{\text{cover}}$ and those of this very same content embedding a message $p_{\text{stego}}$ (i.e., stego) is less than $\epsilon$:

$$\delta\left(p_{\text{cover}}, p_{\text{stego}}\right) \leq \epsilon. \tag{1}$$

The process is called *secure* if $\epsilon = 0$, and in this case the steganography is perfect, creating no statistical differences by the embedding of the message. Steganalysis would then be impossible.

Fortunately, such high performance for a steganographic algorithm is hardly achievable when the payload (the embedded information) is of nonnegligible size; also, several schemes have weaknesses.

One way of measuring the payload is the *embedding rate*, defined as follows.

Let $A$ be a steganographic algorithm, and let $C$ be a cover medium. $A$, by its design, claims that it can embed at most $T_{\text{Max}}$ information bits within $C$; $T_{\text{Max}}$ is called the *capacity* of the medium and highly depends on the steganographic (stego) algorithm as well as the cover medium itself. The

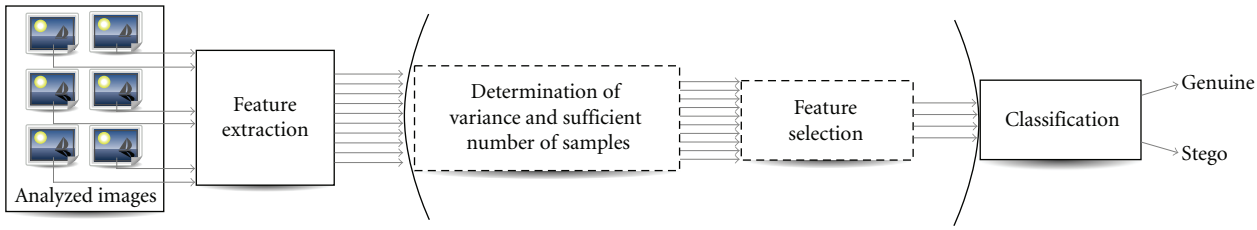


Figure 1: Overview of the typical global processing for an analyzed image: features are first extracted from the image and then processed through a classifier to decide whether the image is cover or stego. In the proposed processing is added an extra step aimed at reducing the features number and having an additional interpretability of the steganalysis results, by doing a feature selection.

embedding rate $T$ is then defined as the part of $T_{\text{Max}}$ used by the information to embed.

For $T_i$ bits to embed in the cover medium, the embedding rate is then $T = T_i/T_{\text{Max}}$, usually expressed as percentage. There are other ways to measure the payload and the relationship between the amount of information embedded and the cover medium, such as the number of *bits per nonzero coefficient*. Meanwhile, the embedding rate has the advantage of taking into account the stego algorithm properties and is not directly based on the cover medium properties—since it uses the stego algorithm estimation of the maximum capacity. Hence the embedding rate has been chosen for this analysis of stego schemes.

This paper is focused onto feature-based steganalysis. Such steganalysis typically uses a certain amount of images for training a classifier: features are extracted from the images and fed to a binary classifier (usually Support Vector Machines) for training. The output of this classifier is "stego" (modified using a steganographic algorithm) or "cover" (genuine). This process is illustrated on Figure 1 for the part without parenthesis.

The emphasis in this paper is more specifically on the issues related to the increasing number of features, which are linked to the universal steganalyzers. Indeed, the very first examples of LSB-based steganalysis made use of less than ten features, with an adapted and specific methodology for each stego algorithm. The idea of "universal steganalyzers" then became popular. In 1999, Westfeld proposes a $\chi^2$-based method, on the LSB of DCT coefficients [4]. Five years after, Fridrich in [5] uses a set of 23 features obtained by normalizations of a much larger set, whilst Lyu and Farid already proposed in 2002 a set of 72 features [6]. Some feature sets [7] also have variable size depending on the DCT block sizes. Since then, an increasing number of research works use supervised learning-based classifiers in very high-dimensional spaces. The recent work of Shi et al. [8] is an example of an efficient result by using 324 features based on JPEG blocks differences modeled by Markov processes.

These new feature sets usually do achieve better and better performance in terms of detection rate and enable to detect most stego algorithm for most embedding rates. Meanwhile, there are some side-effects to this growing number of features. It has been shown, for example, in [9] that the feature space dimensionality in which the considered classifier is trained can have a significant impact on its performances: a too small amount of images regarding dimensionality (the number of features) might lead to an improper training of the classifier and thus to results with a possibly high statistical variance.

In this paper is addressed the idea of a practical way of comparing steganalysis schemes in terms of performance reliability. Ker proposed [10] such comparison by focusing on the pdf of one output of the classifier. Here are studied multiple parameters that can influence this performance:

(1) the number of images used during the training of the classifier: how to determine a sufficient number of images for an efficient and reliable classification (meaning that final results have acceptable variance)?

(2) the number of features used: what are the sufficient and most relevant features for the actual classification problem?

(3) the steganographic method: is there an important influence of the stego algorithm on the general methodology?

(4) the embedding rate used: does the embedding rate used for the steganography modify the variance of the results and the retained best features (by feature selection)?

It can also be noted that images of higher sizes would lead to a smaller secure steganographic embedding rate (following a root-square law), but this phenomenon has already been studied by Filler et al. [11].

The next section details some of the problems related to the number of features used (dimensionality issues) and commonly encountered in steganalysis: (1) the empty space and the distance concentration phenomena, (2) the large variance of the results obtained by the classifier whenever the number of images used for training is not sufficient regarding the number of features, and finally, (3) the lack of interpretability of the results because of the high number of features. In order to address these issues, the methodology sketched on Figure 1 is used and more thoroughly detailed: a sufficient number of images regarding the number of features is first established so that the classifier's training is "reliable" in terms of variance of its results; then, using feature selection the interpretability of the results is improved.

The methodology is finally tested in Section 4 with six different stego algorithms, each using four different embedding rates. Results are finally interpreted thanks to the most relevant selected features for each stego algorithm.

A quantitative study of selected features combinations is then provided.

## 2. Dimensionality Issues and Methodology

The common term "curse of dimensionality" [12] refers to a wide range of problems related to a high number of features. Some of these dimensionality problems are considered in the following, in relation with the number of images and features.

### 2.1. Issues Related to the Number of Images

*2.1.1. The Need for Data Samples.* In order to illustrate this problem in a low-dimensional case, one can consider four samples in a two-dimensional space (corresponding to four images out of which two features have been extracted); the underlying structure leading to the distribution of these four samples seems impossible to infer, and so is the creation of a model for it. Any model claiming it can properly explain the distribution of these samples will behave erratically (because it will extrapolate) when a new sample is introduced. On the contrary, with hundreds to thousands of samples it becomes possible to see clusters and relationships between dimensions.

More generally, in order for any tool to be able to analyze and find a structure within the data, the number of needed samples is growing exponentially with the dimensionality. Indeed, consider a $d$-dimensional unit side hypercube; the number of samples needed to fill the Cartesian grid of step $\epsilon$ inside of it is growing as $O((1/\epsilon)^d)$. Thus using a common grid of step 1/10 in dimension 10, it requires $10^{10}$ samples to fill the grid.

Fortunately, for a model to be built over some high-dimensional data, that data does not have to fill the whole space in the sense of the Cartesian grid. The required space to fill highly depends on the density to be estimated.

In practice, most data sets in steganalysis use at least 10 to 20 dimensions, implying a "needed" number of samples impossible to achieve: storing and processing such number of images is currently impossible. As a consequence, the feature space is not filled with enough data samples to estimate the density with reliable accuracy, which can give wrong or high variance models while building classifiers, having to extrapolate for the missing samples: obtained results can have rather high confidence interval and hence be statistically irrelevant. A claim of performance improvement of 2% using a specific classifier/steganalyzer/steganographic scheme with a variance of 2% is rather meaningless.

*2.1.2. The Increasing Variance of the Results.* The construction of a proper and reliable model for steganalysis is also related to the variance of the results it obtains. Only experimental results are provided to support this claim: with a low number of images regarding the number of features (e.g., a few hundreds of images for 200 features), the variance of the classifier's results can be very important (i.e., the variance of the detection probability).

When the number of images increases, this variance decreases toward low enough values for feature-based steganalysis and performances comparisons. These claims are verified in the next section with the experiments.

*2.1.3. Proposed Solution to the Lack of Images.* Overall, these two problems lead to the same conclusion: the number of images has to be important regarding dimensionality. Theory states that this number is exponential with the number of features, which is impossible to reach for feature-based steganalysis. Hence, the first step of the proposed methodology is to find a "sufficient" number of images for the number of features used, according to a criterion on the variance of the results.

A Bootstrap [13] is proposed for that task: the number of images used for the training of the classifier is increased, and for each different number of images, the variance of the results of the classifier is assessed. Once the variance of the classifier is below a certain threshold, a sufficient number of images have been found (regarding the classifier and the feature set used).

### 2.2. Issues Related to the Number of Features

*2.2.1. The Empty Space Phenomenon.* This phenomenon that was first introduced by Scott and Thompson [14] can be explained with the following example: draw samples from a normal distribution (zero mean and unit variance) in dimension $d$, and consider the probability to have a sample at distance $r$ from the mean of the distribution (zero). It is given by the probability density function:

$$f(r,d) = \frac{r^{d-1}}{2^{d/2-1}} \cdot \frac{e^{-r^2/2}}{\Gamma(d/2)} \tag{2}$$

having its maximum at $r = \sqrt{d-1}$. Thus, when dimension increases, samples are getting farther from the mean of the distribution. A direct consequence of this is that, for the previously mentioned hypercube in dimension $d$, the "center" of it will tend to be empty, since samples are getting concentrated in the borders and corners of the cube.

Therefore, whatever model is used in such a feature space will be trained on scattered samples which are not filling the feature space at all. The model will then not be proper for any sample falling in an area of the space where the classifier had no information about during the training. It will have to extrapolate its behavior for these empty areas and will have unstable performances.

*2.2.2. Lack of Interpretability for Possible "Reverse Engineering".* The interpretability (and its applications) is an important motivation for feature selection and dimensionality reduction: high performances can indeed be reached using the whole 193 features set used in this paper for classification. Meanwhile, if we are looking for the weaknesses and reasons why these features react vividly to a specific algorithm, it seems rather impossible on this important set.

Reducing the required number of features to a small amount through feature selection enables to understand

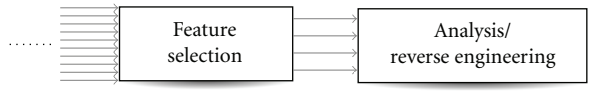


Figure 2: Scheme of the possible reverse engineering on an unknown stego algorithm, by using feature selection for identification of the specific weaknesses.

better why a steganographic model is weak on these particular details, highlighted by the selected features. Such analysis is performed in Section 4.3 for all six steganographic algorithms.

Through the analysis of these selected features, one can consider a "reverse engineering" of the stego algorithm as illustrated on Figure 2. By the identification of the most relevant features, the main characteristics of the embedding method can be inferred, and the steganographic algorithm can be identified if known, or simply understood.

*2.2.3. Proposed Solution to the High Number of Features.* These two issues motivate the feature selection process: if one can reduce the number of features (and hence the dimensionality), the empty space phenomena will have a reduced impact on the classifier used. Also, the set of features obtained by the feature selection process will give insights on the stego scheme and its possible weaknesses.

For this matter, a classical feature selection technique has been used as the second step of the proposed methodology.

The following methodology is different from the one presented previously in [15, 16]. Indeed, in this article, the goal is set toward statistically reliable results. Also, feature selection has the advantage of reducing the dimensionality of the data (the number of features), making the classifier's training much easier. The interpretation of the selected features is also an important advantage (compared to having only the classifier's performance) in that it gives insights on the weaknesses of the stego algorithm.

## 3. Methodology for Benchmarking of Steganographic Schemes

*Addressed Problems.* The number of data points to be used for building a model and classification is clearly an issue, and in the practical case, how many points are needed in order to obtain accurate results—meaning results with small standard deviation.

Reduction of complexity is another main addressed concern in this framework. Then for the selected number of points to be used for classification and also the initial dimensionality given by the features set, two main steps remain.

(i) Choosing the feature selection technique. Since analysis and computation can hardly be done on the whole set of features, the technique used to reduce the dimensionality has to be selected.

(ii) Building a classifier. This implies choosing it, selecting its parameters, training, and validating the chosen model.

The following paragraphs presents the solutions for these two major issues, leading to a methodology combining them, presented on Figure 3.

*3.1. Presentation of the Classifier Used: OP-ELM.* The Optimally-Pruned Extreme Learning Machine (OP-ELM [17, 18]) is a classifier based on the original Extreme Learning Machine (ELM) of Huang et al. [19] (available at: http://www.cis.hut.fi/projects/tsp/index.php?page=OPELM). This classifier makes use of single hidden layer feedforward neural networks (SLFNs) for which the weights and biases are randomly initialized. The goal of the ELM is to reduce the length of the learning process for the neural network, usually very long (e.g., if using classical back-propagation algorithms). The two main theorems on which ELM is based will not be discussed here but can be found in [19]. Figure 4 illustrates the typical structure of an SLFN (simplified to a few neurons in here).

Supposing the neural network is approximating the output $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ perfectly, we would have

$$\sum_{i=1}^{M} \boldsymbol{\beta}_i f\left(\mathbf{w}_i \mathbf{x}_j + b_i\right) = \mathbf{y}_j, \quad j \in [\![1, N]\!], \tag{3}$$

where $N$ is the number of inputs $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ (number of images in our case), and $M$ is the number of neurons in the hidden layer. In the case of steganalysis as performed in this article, $\mathbf{x}_i$ denotes the feature vector corresponding to image $i$, while $\mathbf{y}_i$ is the corresponding class of the image (i.e., stego or cover).

As said, the novelty introduced by the ELM is to initialize the weights $\mathbf{W}$ and biases $\mathbf{B}$ randomly. OP-ELM, in comparison to ELM, brings a greater robustness to data with possibly dependent/correlated features. Also, the use of other functions $f$ (activation functions of the neural network) makes it possible to use OP-ELM for the case where linear components have an important contribution in the classifier's model, for example.

The validation step of this classifier is performed using classical Leave-One-Out cross-validation, much more precise than a $k$-fold cross-validation and hence not requiring any test step [13]. It has been shown on many experiments [17, 18] that the OP-ELM classifier has results very close to the ones of a Support Vector Machine (SVM) while having computational times much smaller (usually from 10 to 100 times).

*3.2. Determination of a Sufficient Number of Images.* A proper number of images, regarding the number of features, has to be determined. Since theoretical values for that number are not reachable, a sufficient number regarding a low enough value of the variance of the results is taken instead (standard deviation will be used instead of variance, in the following).

The OP-ELM classifier is hence used along with a Bootstrap algorithm [13] over 100 repetitions; a subset of the

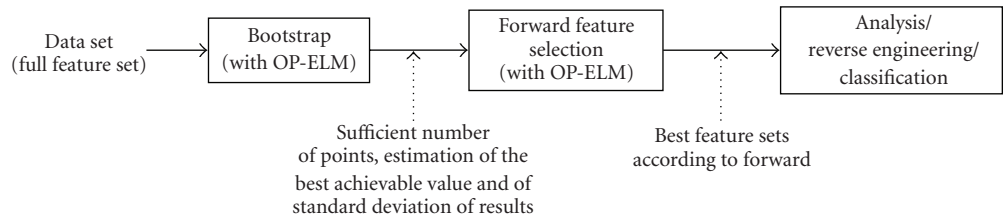


Figure 3: Schematic view of the proposed methodology. (1) An appropriate number of data samples to work with are determined using a Bootstrap method for statistical stability. (2) The Forward selection is performed using an OP-ELM classifier to find a good features set, from which follows a possible interpretation of the features or the typical classification for steganalysis.

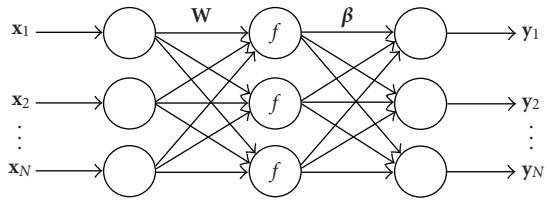


Figure 4: Structure of a classical Single Layer Feedforward Neural Network (SLFN). The input values (the data) $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are weighted by the $\mathbf{W}$ coefficients. A possible bias $\mathbf{B}$ (not on the figure) can be added to the weighted inputs $\mathbf{w}_i\mathbf{x}_i$. An activation function $f$ taking this weighted inputs (plus bias) as input is finally weighted by output coefficients $\boldsymbol{\beta}$ to obtain the output $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$.

**R** = $\{x^i, i \in [\![1, d]\!]\}$
**S** = $\emptyset$
**while R** $\neq \emptyset$ **do**
  **for** $x^j \in$ **R do**
    Evaluate performance with **S** $\cup\, x^j$
  **end for**
  Set **S** = **S** $\cup\, \{x^k\}$, **R** = **R** $- x^k$ with $x^k$ the dimension giving the best result in the loop
**end while**

Algorithm 1: Forward.

complete data set (10000 images, 193 features) is randomly drawn (with possible repetitions). The classifier is trained with that specific subset. This process is repeated 100 times (100 random drawings of the subset) to obtain a statistically reliable estimation of the standard deviation of the results. The size of the subset drawn from the complete data set is then increased, and the 100 iterations are repeated for this new subset size.

The criterion to stop this process is a threshold on the value of the standard deviation of the results. Once the standard deviation of the results gets lower than 1%, it is decided that the subset size $S$, is sufficient. $S$ is then used for the rest of the experiments as a sufficient number of images regarding the number of features in the feature set.

*3.3. Dimensionality Reduction: Feature Selection by Forward with OP-ELM.* Given the sufficient number of images for reliable training of the classifier, $S$, feature selection can be performed. The second step of the methodology, a Forward algorithm with OP-ELM (Figure 3), is used.

*3.3.1. The Forward Algorithm.* The forward selection algorithm is a greedy algorithm [20]; it selects one by one the dimensions, trying to find the one that combines best with the already selected ones. The algorithm is detailed in Algorithm 1 (with $x^i$ denoting the $i$th dimension of the data set).

Algorithm 1 requires $d(d-1)/2$ instances to terminate (to be compared to the $2^d - 1$ instances for an exhaustive search), which might reach the computational limits, depending on the number of dimensions and time to evaluate the performance with one set. With the OP-ELM as a classifier, computational times for the Forward selection are not much of an issue.

Even if its capacity to isolate efficient features is clear, the Forward technique has some drawbacks. First, if two features present good results when they are put together but poor results if only one of them is selected, Forward might not take these into account in the selection process.

Table 1: Performances for OP-ELM LOO for the best features set along with the size of the reduced feature set (number). Performances using the reduced set are within the 1% range of standard deviation of the best results. The size of the set has been determined to be the smallest possible one giving this performance.

| | 5% | Number | 10% | Number |
|---|---|---|---|---|
| F5 | 73.3 | 46 | 83.9 | 38 |
| JPHS | 90.7 | 41 | 92.1 | 21 |
| MBSteg | 63.3 | 57 | 70.9 | 93 |
| MM3 | 78.00 | 81 | 86.2 | 49 |
| OutGuess | 81.2 | 65 | 93.2 | 49 |
| Steghide | 82.3 | 149 | 91.2 | 89 |
| | 15% | Number | 20% | Number |
| F5 | 90.5 | 33 | 96.3 | 15 |
| JPHS | 93.7 | 41 | 97.3 | 25 |
| MBSteg | 83.5 | 73 | 88.5 | 69 |
| MM3 | 86.6 | 57 | 86.6 | 73 |
| OutGuess | 98.8 | 33 | 100.0 | 29 |
| Steghide | 96.4 | 73 | 99 | 73 |

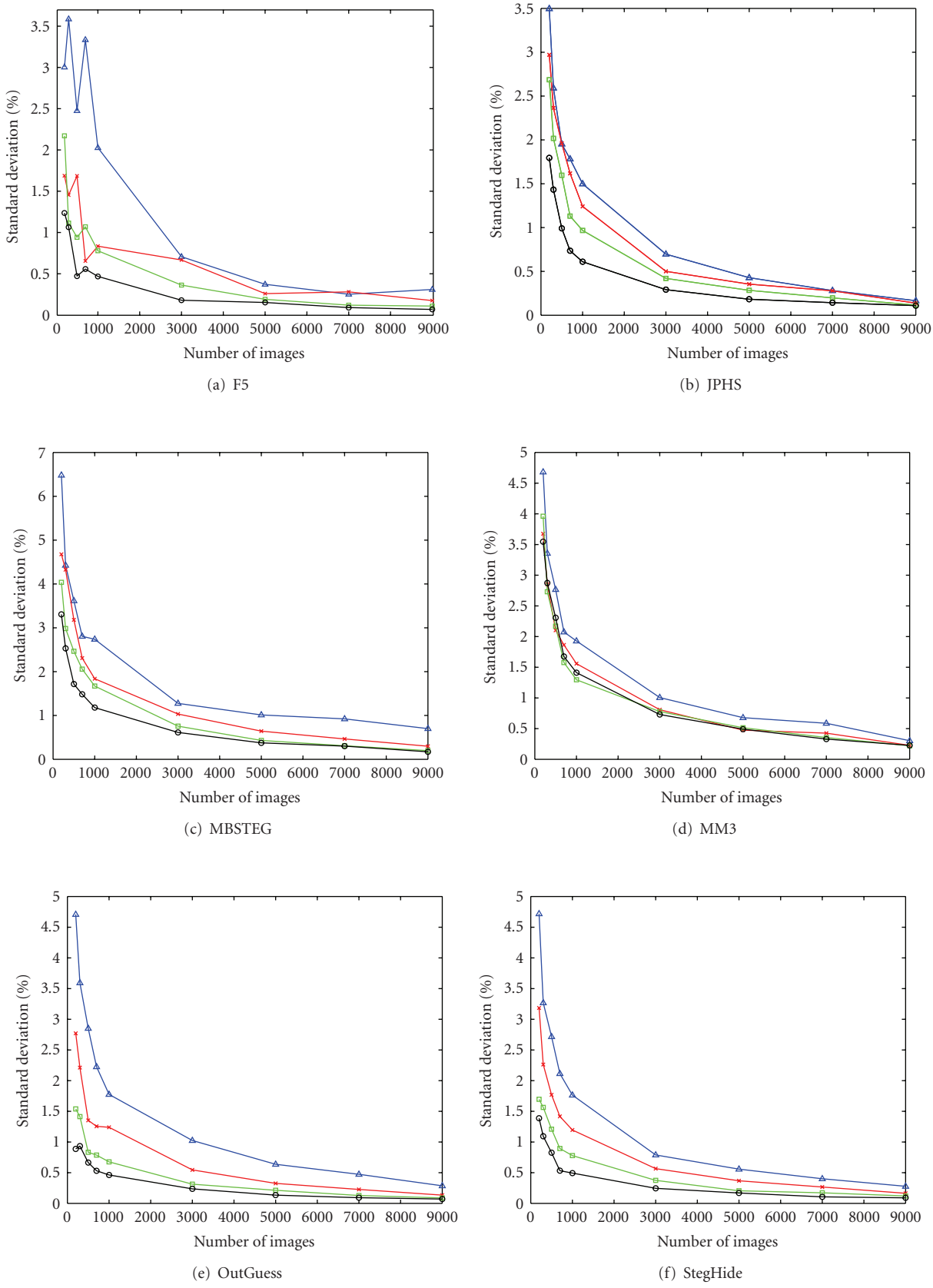


Figure 5: Standard deviation in percentage of the average classification result versus the number of images, for all six steganographic algorithms, for the four embedding rates: black circles (○) for 20%, green squares (□) for 15%, red crosses (×) for 10%, and blue triangles (△) for 5%. These estimations have been performed with the Bootstrap runs (100 iterations). Plots do not have the same scale, vertically.

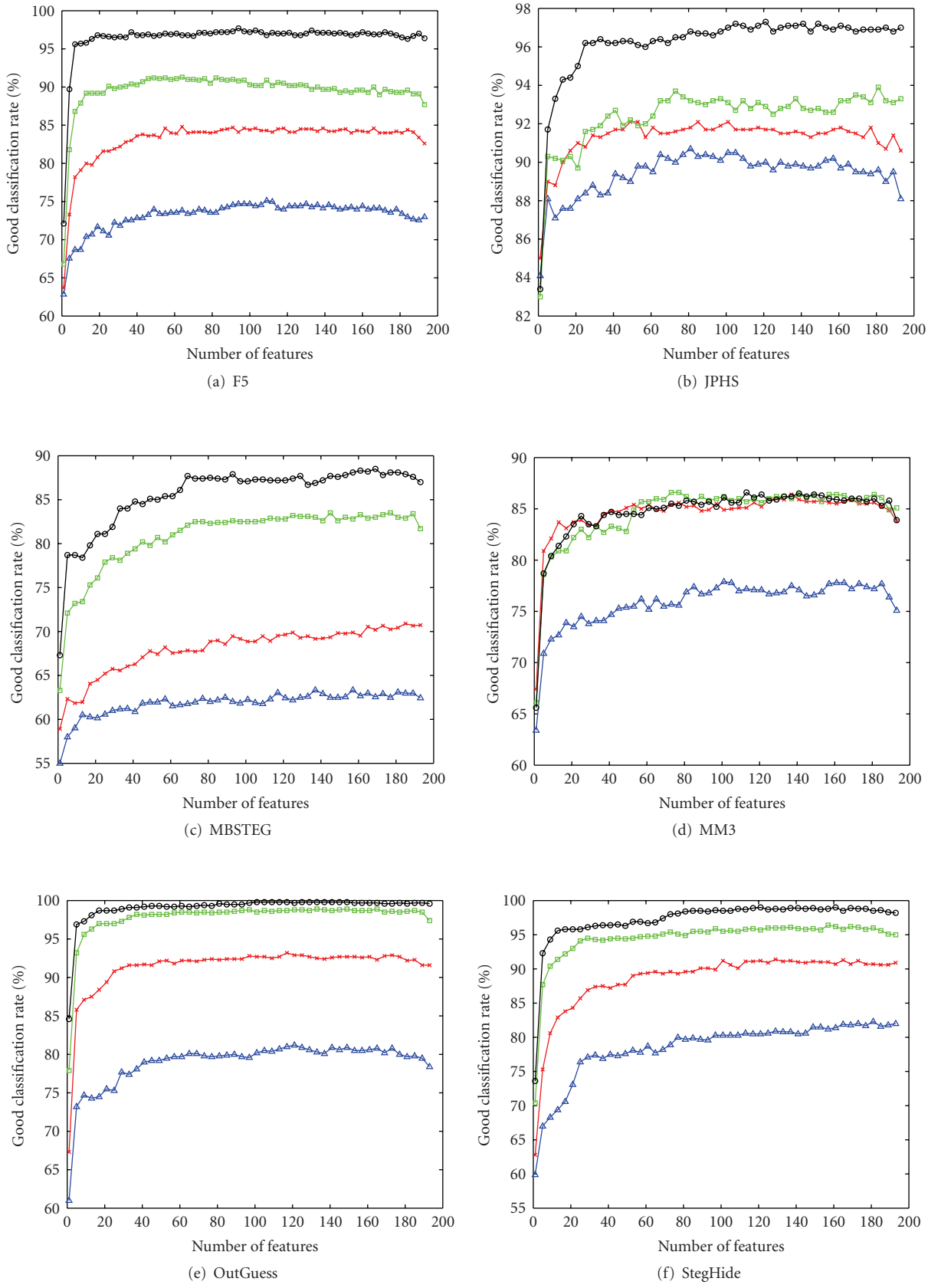


FIGURE 6: Performance in detection for all six stego algorithms versus the number of features, for the four embedding rates: black circles (○) for 20%, green squares (□) for 15%, red crosses (×) for 10%, and blue triangles (△) for 5%. Features are ranked using the Forward selection algorithm. These plots are the result of a single run of the Forward algorithm. Plots do not have the same scale, vertically.

Table 2: The 23 features previously detailed.

| Functional/Feature | Functional **F** |
|---|---|
| Global histogram | $\mathbf{H}/\|\mathbf{H}\|$ |
| Individual histogram for 5 DCT Modes | $\mathbf{h}^{21}/\|\mathbf{h}^{21}\|, \mathbf{h}^{12}/\|\mathbf{h}^{12}\|, \mathbf{h}^{13}/\|\mathbf{h}^{13}\|,$ $\mathbf{h}^{22}/\|\mathbf{h}^{22}\|, \mathbf{h}^{31}/\|\mathbf{h}^{31}\|$ |
| Dual histogram for 11 DCT values | $\mathbf{g}^{-5}/\|\mathbf{g}^{-5}\|,\ \mathbf{g}^{-4}/\|\mathbf{g}^{-4}\|, \ldots,$ $\mathbf{g}^{4}/\|\mathbf{g}^{4}\|, \mathbf{g}^{5}/\|\mathbf{g}^{5}\|$ |
| Variation | **V** |
| $L1$ and $L2$ blockiness | $\mathbf{B}_1, \mathbf{B}_2$ |
| Cooccurrence | $N_{00}, N_{01}, N_{11}$ |

Second, it does not allow to "go back" in the process, meaning that if performances are decreasing along the selection process, and that the addition of another feature makes performances increase again, combinations of previously selected features with this last one are not possible anymore.

The Forward selection is probably not the best possible feature selection technique, and recent contribution to these techniques such as Sequential Floating Forward Selection (SFFS) [21] and improvements of it [22] has shown that the number of computations required for feature selection can be reduced drastically. Nevertheless, the feature selection using Forward has been showing very good results and seems to perform well on the feature set used in this paper. It is not used here in the goal of obtaining the best possible combination of features but more to reduce the dimensionality and obtain some meaning out of the selected features. Improvements of this methodology could make use of such more efficient techniques of feature selection.

*3.4. General Methodology.* To summarize the general methodology on Figure 3 uses first a Bootstrap with 100 iterations on varying subsets sizes, to obtain a sufficient subset size and statistically reliable classifiers' results regarding the number of features used. With this number of images feature selection is performed using a Forward selection algorithm; this enables to highlight possible weaknesses of the stego algorithm.

This methodology has been applied to six popular stego algorithms for testing. Experiments and results as well as a discussion on the analysis of the selected features are given in the next section.

## 4. Experiments and Results

### *4.1. Experiments Setup*

*4.1.1. Steganographic Algorithms Used.* Six different steganographic algorithms have been used: F5 [23], Model-Based (MBSteg) [24], MMx [25] (in these experiments, MM3 has been used), JP Hide and Seek [26], OutGuess [27], and StegHide [28]; all of them with four different embedding rates: 5%, 10%, 15%, and 20%.

*4.1.2. Generation of Image Database.* The image base was constituted of 10 000 images from the BOWS2 Challenge [29] database (hosted by Westfeld [30]). These images are $512 \times 512$ PGM greyscale (also available in color).

The steganographic algorithms and the proposed methodology for dimensionality reduction and steganalysis are only performed on these $512 \times 512$ images. It can also be noted that depending on image complexity, as studied in [31], local discrepancies might be observed (a classically trained steganalyzer might have troubles for such images), but on a large enough base of images, this behavior will not be visible.

*4.1.3. Extraction of the Features.* In the end, the whole set of images is separated in two equal parts: one is kept as untouched cover while the other one is stego with the six steganographic algorithms at four different embedding rates: 5%, 10%, 15%, and 20%. Fridrich's 193 DCT features [32] have been used for the steganalysis.

*4.2. Results.* Results are presented following the methodology steps. A discussion over the selected features and the possible interpretability of it are developed afterward. In the following, the term "detection rate" stands for the performance of the classifier on a scale from 0% to 100% of classification rate. It is a measure of the performance instead of a measure of error.

*4.2.1. Determination of Sufficient Number of Samples.* Presented first is the result of the evaluation of a sufficient number of images, as explained in the previous methodology, in Figure 5. The Bootstrap (100 rounds) is used on randomly taken subsets of 200 up to 9000 images out of the whole 10 000 from the BOWS2 challenge.

It can be seen on Figure 5 that the standard deviation behaves as expected when increasing the number of images for the cases of JPHS, MBSteg, MMx, OutGuess, and StegHide: its value decreases and tends to be below 1% of the best performance when the number of images is 5000 (even if for MBSteg with embedding rate of 5% it is a bit above 1%). This sufficient number of samples is kept as the reference and sufficient number. Another important point is that with very low number of images (100 in these cases), the standard deviation is between 1% and about 6.5% of the average classifier's performance; meaning that results computed with small number of images have at most a $\pm 6.5\%$ confidence interval. While the plots decrease very quickly when increasing the number of images, values of the standard deviation remain very high until 2000 images; these results have to take into account the embedding rate, which tends to make the standard deviation higher as it decreases.

Indeed, while differences between 15% and 20% embedding rates are not very important on the four previously mentioned stego algorithms, there is a gap between the 5%–10% plots and the 20% ones. This is expected when looking at the performances of the steganalysis process: low embedding rates tend to be harder to detect, leading to a range of possible performances wider than with high embedding rates. Figure 6 illustrates this idea on all six cases (F5, JPHS, MMx, MBSteg, StegHide, and OutGuess).

The final "sufficient" number of samples retained for the second step of the methodology—the feature selection—is 5000, for two reasons: first, the computational times are acceptable for the following computations (feature selection step with training of classifier for each step); second, the standard deviation is small enough to consider that the final classification results are given with at most 1% of standard deviation (in the case of MBSteg at 5% of embedding rate).

*4.2.2. Forward Feature Selection.* Features have first been ranked, using the Forward feature selection algorithm, and detection rates are plotted with increasing number of features (using the ranking provided by the Forward selection) on Figure 6.

The six analyzed stego algorithms give rather different results.

(i) F5 reaches very quickly the maximum performance for all embedding rates: only few features contribute to the overall detection rate.

(ii) JPHS reaches a plateau in performance (within the standard deviation of 1%) for all embedding rates with 41 features and remains around that performance.

(iii) OutGuess has this same plateau at 25 features, and performances are not increasing anymore above that number of features (still within the standard deviation of the results).

(iv) StegHide can be considered to have reached the maximum result (within the standard deviation interval) at 60 features.

(v) In the MM3 case, performances for embedding rates 10%, 15%, and 20% are very similar as are selected features. Performances stable at 40 features. The difference for the 5% case is most likely due to matrix embedding which makes detection harder when the payload is small.

(vi) Performances for MBSteg are stable using 70 features for embedding rates 15% and 20%. Only 30 are enough for embedding rate 5%. The case of embedding rate 10% has the classifier's performances increasing with the addition of features.

Interestingly, the features retained by the Forward selection for each embedding rate differ slightly within one steganographic algorithm. Details about the features ranked as first by the Forward algorithm are discussed afterward.

*4.3. Discussion.* First, the global performances, when using the reduced and sufficient feature sets mentioned in the results section above, are assessed. Note that feature selection for performing reverse engineering of a steganographic algorithm is theoretically efficient only if the features are carrying different information (if two features represent the same information, the feature selection will select only one of them).

*4.3.1. Reduced Features Sets.* Based on the ranking of the features obtained by the Forward algorithm, it has been decided that once performances were within 1% of the best performance obtained (among all Forward tryouts for all different sets of features), the number of features obtained was retained as a "sufficient" feature set. Performances using reduced feature sets (proper to each algorithm and embedding rate) are first compared in Table 1. It can be seen that, globally, the required size of the set of features for remaining within 1% of the best performance decreases.

It should be noted that since the aim of the feature selection is to reduce as much as possible the feature set while keeping overall same performance, it is expected that within the standard deviation interval the performance with the lowest possible number of features is behind the "maximum" one.

It remains possible, for the studied algorithms, as Figure 6 shows, to find a higher number of features for which the performance is closer or equal to the maximum one—even though this is very disputable, considering the maximal 1% standard deviation interval when using 5000 images. But this is not the goal of the feature selection step of the methodology.

*4.4. Feature Sets Analysis for Reverse Engineering.* Common feature sets have been selected according to the following rule: take the first common ten features (in the order ranked by the Forward algorithm) to each feature set obtained for each embedding rate (within one algorithm). It is hoped that through this selection the obtained features will be generic regarding the embedding rate.

We recall first the meaning of the different features used in this steganalysis scheme. Notations for the feature set used [32] are given for the original 23 features set, in Table 2.

This set of 23 features is expanded up to a set of 193, by removing the $L_1$ norm used previously and keeping all the values of the matrices and vectors. This results in the following 193 features set.

(i) A global histogram of 11 dimensions $\mathbf{H}(i)$, $i = [\![-5, 5]\!]$.

(ii) 5 low frequency DCT histograms each of 11 dimensions $\mathbf{h}^{21}(i) \cdots \mathbf{h}^{31}(i)$, $i = [\![-5, 5]\!]$.

(iii) 11 dual histograms each of 9 dimensions $\mathbf{g}^{-5}(i) \cdots \mathbf{g}^{5}(i)$, $i = [\![1, 9]\!]$.

(iv) Variation between blocks, of dimension 1 $\mathbf{V}$.

(v) 2 blockinesses of dimension 1 $\mathbf{B}_1, \mathbf{B}_2$.

(vi) Cooccurrence matrix of dimensions $5 \times 5$ $\mathbf{C}_{i,j}$, $i = [\![-2, 2]\!]$, $j = [\![-2, 2]\!]$.

The following is a discussion on the selected features for each steganographic algorithm.

Tables of selected feature sets are presented in Tables 3–8, with an analysis for each algorithm. Fridrich's DCT features are not the only ones having a possible physical interpretation. They have been chosen here because it is believed that most of the features can be interpreted. The

Table 3: Common feature set for F5 with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{B}_1$ | $\mathbf{C}_{-1,-1}$ | $\mathbf{C}_{-2,0}$ | $\mathbf{H}(0)$ | $\mathbf{B}_2$ |
| (4) | (8) | (12) | (13) | (19) |
| $\mathbf{V}$ | $\mathbf{g}^0(1)$ | $\mathbf{h}^{22}(-3)$ | $\mathbf{h}^{12}(3)$ | $\mathbf{h}^{13}(-3)$ |
| (21) | (22) | (26) | (31) | (55) |

Table 4: Common feature set for MM3 with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{C}_{-1,1}$ | $\mathbf{C}_{-2,0}$ | $\mathbf{h}^{13}(-1)$ | $\mathbf{H}(-1)$ | $\mathbf{h}^{21}(-3)$ |
| (1) | (3) | (7) | (22) | (22) |
| $\mathbf{g}^{-5}(1)$ | $\mathbf{C}_{1,0}$ | $\mathbf{h}^{22}(-3)$ | $\mathbf{h}^{31}(-2)$ | $\mathbf{H}(-3)$ |
| (35) | (40) | (41) | (42) | (49) |

Table 5: Common feature set for JPHS with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{g}^0(4)$ | $\mathbf{B}_1$ | $\mathbf{h}^{21}(-3)$ | $\mathbf{h}^{21}(-1)$ | $\mathbf{H}(-5)$ |
| (1) | (25) | (26) | (30) | (30) |
| $\mathbf{h}^{13}(3)$ | $\mathbf{h}^{31}(3)$ | $\mathbf{g}^0(5)$ | $\mathbf{g}^3(7)$ | $\mathbf{g}^{-3}(1)$ |
| (34) | (52) | (61) | (61) | (65) |

Table 6: Common feature set for MBSteg with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{C}_{2,-2}$ | $\mathbf{C}_{2,2}$ | $\mathbf{C}_{-2,2}$ | $\mathbf{C}_{2,0}$ | $\mathbf{g}^{-2}(3)$ |
| (4) | (6) | (10) | (10) | (24) |
| $\mathbf{g}^0(4)$ | $\mathbf{H}(-2)$ | $\mathbf{C}_{-1,-2}$ | $\mathbf{H}(1)$ | $\mathbf{H}(2)$ |
| (27) | (31) | (32) | (36) | (50) |

Table 7: Common feature set for OutGuess with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{C}_{-2,0}$ | $\mathbf{H}(-2)$ | $\mathbf{C}_{-2,-2}$ | $\mathbf{C}_{0,-2}$ | $\mathbf{h}^{13}(0)$ |
| (3) | (3) | (7) | (8) | (12) |
| $\mathbf{H}(-3)$ | $\mathbf{C}_{-1,0}$ | $\mathbf{h}^{22}(1)$ | $\mathbf{H}(0)$ | $\mathbf{g}^{-4}(8)$ |
| (14) | (23) | (31) | (41) | (45) |

Table 8: Common feature set for StegHide with average rank for each feature.

| | | | | |
|---|---|---|---|---|
| $\mathbf{C}_{2,0}$ | $\mathbf{C}_{-2,2}$ | $\mathbf{C}_{-2,0}$ | $\mathbf{B}_1$ | $\mathbf{B}_2$ |
| (6) | (22) | (22) | (25) | (27) |
| $\mathbf{g}^1(1)$ | $\mathbf{h}^{13}(5)$ | $\mathbf{h}^{21}(-3)$ | $\mathbf{C}_{-2,-1}$ | $\mathbf{g}^{-1}(4)$ |
| (28) | (45) | (46) | (47) | (54) |

proposed short analysis of the weaknesses of stego algorithms is using this interpretation.

*4.4.1. F5I .* F5 (Table 3) is rather sensitive to both blockiness detections and, interestingly, is the only of the six tested algorithms to be sensitive to the variation $\mathbf{V}$. As for other algorithms, cooccurrence coefficients are triggered.

*4.4.2. MM3.* MM3 (Table 4) tends to be sensitive to global histogram features as well as DCT histograms, which are not preserved.

*4.4.3. JPHS.* JPHS (Table 5) seems not to preserve the DCT coefficients histograms. Also the dual histograms react vividly for center values and extremes ones ($-3$ and 3).

*4.4.4. MBSteg.* The features used (Table 6) include global histograms with values 1, $-2$, and 2, which happens only because of the calibration in the feature extraction process. MBSteg preserves the coefficients' histograms but does not take into account a possible calibration. Hence, the unpreserved histograms are due to the calibration process in the feature extraction. Information leaks through the calibration process. Also cooccurrence values are used, which is a sign that MBSteg does not preserve low and high frequencies.

*4.4.5. OutGuess.* Cooccurrence values are mostly used (values $-2$, $-1$) in the feature set for OutGuess (Table 7) and a clear weak point. The calibration process has also been of importance since the global histograms of extreme values $-3$ and $-2$ have been taken into account.

*4.4.6. StegHide.* For StegHide (Table 8), blockiness and cooccurrence values are mostly used, again for low and high frequencies.

From a general point of view, it can be seen that most of the analyzed algorithms are sensitive to statistics of lowpass-calibrated DCT coefficients, represented by features $\mathbf{h}^{13}$ and $\mathbf{h}^{21}$. This is not surprising since these coefficients contain a large part of the information of a natural image; their associated densities are likely to be modified by the embedding process.

## 5. Conclusions

This paper has presented a methodology for the estimation of a sufficient number of images for a specific feature set using the standard deviation of the detection rate obtained by the classifier as a criterion (a Bootstrap technique is used for that purpose); the general methodology presented can nonetheless be extended and applied to other feature sets. The second step of the methodology aims at reducing the dimensionality of the data set, by selecting the most relevant features, according to a Forward selection algorithm; along with the positive effects of a lower dimensionality, analysis of the selected features is possible and gives insights on the steganographic algorithm studied.

Three conclusions can be drawn from the methodology and experiments in this paper.

(i) Results on standard deviation for almost all studied steganographic algorithms have proved that the feature-based steganalysis is reliable and accurate only if a sufficient number of images is used for the actual training of the classifier used. Indeed, from most of the results obtained concerning standard deviation values (and therefore statistical stability of the results), it is rather irrelevant to possibly increase detection performance by 2% while working with a standard deviation for these same results of 2%.

Table 9: The 40 first features ranked by the Forward algorithm for the F5 algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $h^{13}(0)$ | $H(0)$ | $B_1$ | $V$ | $C_{0,0}$ | $g^0(2)$ | $h^{31}(-1)$ | $C_{2,1}$ | $g^0(7)$ | $C_{2,-1}$ |
| $g^{-2}(7)$ | $g^{-3}(4)$ | $B_2$ | $h^{12}(-5)$ | $g^{-1}(9)$ | $g^4(5)$ | $g^5(3)$ | $g^{-4}(5)$ | $g^{-4}(9)$ | $g^3(5)$ |
| $h^{31}(3)$ | $h^{13}(1)$ | $g^{-1}(6)$ | $g^{-2}(1)$ | $h^{13}(2)$ | $h^{12}(5)$ | $g^3(6)$ | $C_{1,-2}$ | $h^{13}(-5)$ | $h^{22}(5)$ |
| $g^{-4}(1)$ | $g^4(9)$ | $C_{2,-2}$ | $g^{-3}(6)$ | $g^{-5}(9)$ | $h^{12}(3)$ | $h^{31}(0)$ | $h^{21}(-4)$ | $g^2(9)$ | $g^0(9)$ |

Table 10: The 40 first features ranked by the Forward algorithm for the JPHS algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $g^0(4)$ | $h^{22}(0)$ | $C_{1,0}$ | $B_1$ | $H(1)$ | $h^{21}(0)$ | $g^1(4)$ | $g^0(8)$ | $g^{-2}(9)$ | $g^{-2}(5)$ |
| $g^4(5)$ | $g^0(5)$ | $g^1(9)$ | $g^{-1}(2)$ | $B_2$ | $g^2(8)$ | $C_{0,0}$ | $h^{31}(5)$ | $g^0(9)$ | $h^{22}(1)$ |
| $g^{-2}(2)$ | $g^{-1}(7)$ | $g^{-3}(8)$ | $g^0(1)$ | $h^{31}(-3)$ | $h^{21}(-1)$ | $h^{22}(-1)$ | $g^{-4}(6)$ | $C_{-1,-2}$ | $g^5(7)$ |
| $h^{12}(-5)$ | $g^{-5}(8)$ | $h^{21}(2)$ | $g^0(7)$ | $h^{12}(-2)$ | $h^{22}(-4)$ | $h^{31}(0)$ | $C_{0,2}$ | $H(2)$ | $g^5(5)$ |

Table 11: The 40 first features ranked by the Forward algorithm for the MBSteg algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $g^{-2}(1)$ | $H(2)$ | $g^{-4}(7)$ | $h^{13}(1)$ | $h^{22}(1)$ | $C_{2,-2}$ | $C_{-1,-1}$ | $h^{31}(1)$ | $g^4(7)$ | $g^{-2}(4)$ |
| $h^{21}(0)$ | $h^{31}(-4)$ | $h^{21}(-4)$ | $C_{0,2}$ | $C_{1,2}$ | $h^{31}(-1)$ | $H(0)$ | $h^{21}(3)$ | $g^{-5}(6)$ | $h^{22}(-3)$ |
| $h^{13}(-1)$ | $C_{2,0}$ | $C_{1,2}$ | $g^5(6)$ | $C_{-2,-1}$ | $g^{-3}(6)$ | $g^5(4)$ | $g^{-2}(7)$ | $g^{-1}(7)$ | $g^{-4}(8)$ |
| $h^{22}(-1)$ | $g^2(1)$ | $g^0(8)$ | $h^{22}(-5)$ | $H(-2)$ | $h^{12}(-4)$ | $g^5(5)$ | $h^{12}(-2)$ | $g^2(4)$ | $h^{21}(-3)$ |

Table 12: The 40 first features ranked by the Forward algorithm for the MM3 algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $C_{-1,-1}$ | $h^{13}(-1)$ | $C_{0,-2}$ | $C_{1,1}$ | $g^0(9)$ | $C_{2,0}$ | $h^{21}(-1)$ | $h^{13}(1)$ | $g^{-3}(2)$ | $C_{1,0}$ |
| $H(-2)$ | $g^4(4)$ | $g^2(2)$ | $C_{-2,0}$ | $C_{0,-1}$ | $C_{-1,-2}$ | $g^{-2}(3)$ | $h^{22}(-3)$ | $g^2(3)$ | $h^{13}(3)$ |
| $h^{31}(-1)$ | $g^{-1}(9)$ | $g^{-2}(8)$ | $g^0(7)$ | $h^{21}(-5)$ | $h^{21}(3)$ | $C_{-1,1}$ | $g^{-1}(3)$ | $g^5(3)$ | $h^{31}(1)$ |
| $g^0(3)$ | $B_1$ | $C_{-2,1}$ | $B_2$ | $g^{-4}(6)$ | $C_{0,2}$ | $H(-1)$ | $g^2(5)$ | $h^{13}(0)$ | $g^2(7)$ |

Table 13: The 40 first features ranked by the Forward algorithm for the Outguess algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $h^{13}(0)$ | $C_{0,-1}$ | $C_{-2,0}$ | $H(-2)$ | $B_1$ | $C_{0,-2}$ | $g^0(7)$ | $h^{31}(-3)$ | $C_{-2,-1}$ | $g^0(2)$ |
| $B_2$ | $H(-1)$ | $g^{-2}(2)$ | $h^{13}(-1)$ | $h^{22}(-1)$ | $h^{22}(0)$ | $h^{12}(-3)$ | $g^{-2}(5)$ | $g^1(8)$ | $h^{21}(-2)$ |
| $g^{-2}(9)$ | $g^1(1)$ | $H(5)$ | $H(4)$ | $g^2(1)$ | $g^0(1)$ | $g^{-3}(5)$ | $g^0(9)$ | $g^{-3}(8)$ | $g^{-3}(3)$ |
| $g^{-5}(4)$ | $g^{-5}(5)$ | $C_{-2,-2}$ | $g^{-1}(6)$ | $g^{-2}(6)$ | $g^4(3)$ | $C_{-1,-1}$ | $C_{-1,0}$ | $g^{-2}(7)$ | $C_{-1,1}$ |

Table 14: The 40 first features ranked by the Forward algorithm for the Steghide algorithm at 5% embedding rate.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $C_{0,-1}$ | $g^0(2)$ | $C_{0,2}$ | $C_{2,-2}$ | $B_1$ | $B_2$ | $C_{1,1}$ | $C_{0,-2}$ | $C_{-2,2}$ | $h^{13}(-1)$ |
| $g^{-5}(3)$ | $h^{21}(-3)$ | $C_{0,1}$ | $h^{13}(0)$ | $C_{1,-1}$ | $h^{31}(-1)$ | $g^{-3}(3)$ | $g^3(6)$ | $h^{31}(-2)$ | $g^1(3)$ |
| $h^{22}(1)$ | $C_{-2,-2}$ | $g^{-4}(4)$ | $h^{13}(1)$ | $C_{-2,0}$ | $g^1(4)$ | $C_{2,1}$ | $H(-1)$ | $C_{2,2}$ | $h^{22}(5)$ |
| $g^2(5)$ | $C_{-1,-1}$ | $g^1(9)$ | $C_{2,0}$ | $g^2(7)$ | $g^{-1}(1)$ | $h^{31}(5)$ | $H(-2)$ | $h^{21}(1)$ | $g^{-2}(9)$ |

(ii) Through the second step of the methodology, the required number of features for steganalysis can be decreased. This with three main advantages: (a) performances remain the same if the reduced feature set is properly constructed; (b) the selected features from the reduced set are relevant and meaningful (the selected set can possibly vary, according to the feature selection technique used) and make reverse-engineering possible; (c) the weaknesses of the stego algorithm also appear from the selection; this can lead, for example, to improvements of the stego algorithm.

(iii) The analysis on the reduced common feature sets obtained between embedding rates of the same stego algorithm shows that the algorithms are sensitive to roughly the same features, as a basis. Meanwhile, when embedding rates get as low as 5%, or for very efficient algorithms, some very specific features appear.

Hence, the first step of the methodology is a requirement for not only any new stego algorithm but also new feature sets/steganalyzers, willing to present its performances: a sufficient number of images for the stego algorithm and the

steganalyzer used to test it have to be assessed in order to have stable results (i.e., with a small enough standard deviation of its results to make the comparison with current state of the art techniques meaningful).

Also, from the second step of the methodology, the most relevant features can be obtained and make possible a further analysis of the stego algorithm considered, additionally to the detection rate obtained by the steganalyzer.

## Appendix

## Features Ranked by the Forward Algorithm

In appendix are given the first 40 features obtained by the Forward ranking for each stego algorithm with 5% embedding rate (Tables 9–14). Only one embedding rate result is given for space reasons. 5% embedding rate results have been chosen since they tend to be different (in terms of ranked features by the Forward algorithm) from the other embedding rates and also because 5% embedding rate is a difficult challenge in terms of steganalysis; these features are meaningful for this kind of difficult steganalysis with these six algorithms.

## Acknowledgments

The authors would like to thank Jan Kodovsky and Jessica Fridrich for their implementation of the DCT Feature Extraction software. Also many thanks to Tomás Pevný for his helpful comments and suggestions on this article. The work in this paper was supported (in part) by the French national funding under the project RIAM Estivale (ANR-05-RIAM-O1903), ANR projects Nebbiano (ANR-06-SETI-009), and TSAR French Project (ANR SSIA 2006-2008).

## References

[1] D. McCullagh, "Secret Messages Come in .Wavs. Wired News," February 2001, http://www.wired.com/news/politics/0,1283,41861,00.html.

[2] N. Provos and P. Honeyman, "Detecting steganographic content on the internet," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '02)*, San Diego, Calif, USA, February 2002.

[3] C. Cachin, "An information-theoretic model for steganography," in *Proceedings of the 2nd International Workshop on Information Hiding (IH '98)*, vol. 1525 of *Lecture Notes in Computer Science*, pp. 306–318, Portland, Ore, USA, April 1998.

[4] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proceedings of the 3rd International Workshop on Information Hiding (IH '99)*, vol. 1768 of *Lecture Notes in Computer Science*, pp. 61–76, Springer, Dresden, Germany, September-October 2000.

[5] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Proceedings of the 6th International Workshop on Information Hiding (IH '04)*, vol. 3200 of *Lecture Notes in Computer Science*, pp. 67–81, Toronto, Canada, May 2004.

[6] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," in *Proceedings of the 5th International Workshop on Information Hiding (IH '02)*, vol. 2578 of *Lecture Notes in Computer Science*, pp. 340–354, Noordwijkerhout, The Netherlands, October 2003.

[7] S. S. Agaian and H. Cai, "New multilevel dct, feature vectors, and universal blind steganalysis," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681 of *Proceedings of SPIE*, pp. 653–663, San Jose, Calif, USA, January 2005.

[8] Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Proceedings of the 8th International Workshop on Information Hiding (IH '06)*, vol. 4437 of *Lecture Notes in Computer Science*, pp. 249–264, Alexandria, Va, USA, July 2007.

[9] D. François, *High-dimensional data analysis: optimal metrics and feature selection*, Ph.D. thesis, Université Catholique de Louvain, Louvain, Belgium, September 2006.

[10] A. D. Ker, "The ultimate steganalysis benchmark?" in *Proceedings of the 9th Multimedia and Security Workshop (MM/Sec '07)*, pp. 141–148, Dallas, Tex, USA, September 2007.

[11] T. Filler, A. D. Ker, and J. Fridrich, "The square root law of steganographic capacity for Markov covers," in *Media Forensics and Security*, E. J. Delp III, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., vol. 7254 of *Proceedings of SPIE*, pp. 1–11, San Jose, Calif, USA, January 2009.

[12] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, USA, 1961.

[13] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall/CRC, Londres, Argentina, 1994.

[14] D. W. Scott and J. R. Thompson, "Probability density estimation in higher dimensions," in *Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface*, S. R. Douglas, Ed., pp. 173–179, North-Holland, Houston, Tex, USA, March 1983.

[15] Y. Miche, P. Bas, A. Lendasse, C. Jutten, and O. Simula, "Extracting relevant features of steganographic schemes by feature selection techniques," in *Proceedings of the 3rd Wavilla Challenge (Wacha '07)*, pp. 1–15, St. Malo, France, June 2007.

[16] Y. Miche, B. Roue, A. Lendasse, and P. Bas, "A feature selection methodology for steganalysis," in *Proceedings of the International Workshop on Multimedia Content Representation, Classification and Security (MRCS '06)*, vol. 4105 of *Lecture Notes in Computer Science*, pp. 49–56, Istanbul, Turkey, September 2006.

[17] Y. Miche, P. Bas, C. Jutten, O. Simula, and A. Lendasse, "A methodology for building regression models using extreme learning machine: OP-ELM," in *Proceedings of the 16th European Symposium on Artificial Neural Networks (ESANN '08)*, pp. 1–6, Bruges, Belgium, April 2008.

[18] A. Sorjamaa, Y. Miche, R. Weiss, and A. Lendasse, "Long-term prediction of time series using NNE-based projection and OP-ELM," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 2674–2680, Hong Kong, June 2008.

[19] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[20] F. Rossi, A. Lendasse, D. François, V. Wertz, and M. Verleysen, "Mutual information for the selection of relevant variables in spectrometric nonlinear modelling," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, no. 2, pp. 215–226, 2006.

[21] D. Ververidis and C. Kotropoulos, "Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition," *Signal Processing*, vol. 88, no. 12, pp. 2956–2970, 2008.

[22] D. Ververidis and C. Kotropoulos, "Fast sequential floating forward selection applied to emotional speech features estimated on des and susas data collections," in *Proceeding of the 14th European Signal Processing Conference (EUSIPCO '06)*, EURASIP, Ed., pp. 1–5, Florence, Italy, September 2006.

[23] A. Westfeld, "F5—a steganographic algorithm," in *Proceedings of the 4th International Workshop on Information Hiding (IH '01)*, vol. 2137 of *Lecture Notes in Computer Science*, pp. 289–302, Pittsburgh, Pa, USA, April 2001.

[24] P. Sallee, "Model-based steganography," in *Proceedings of the 2nd International Workshop Digital Watermarking (IWDW '03)*, vol. 2939 of *Lecture Notes in Computer Science*, pp. 254–260, Seoul, Korea, October 2004.

[25] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proceedings of the 8th International Workshop on Information Hiding (IH '06)*, vol. 4437 of *Lecture Notes in Computer Science*, pp. 314–327, Alexandria, Va, USA, July 2007.

[26] A. Latham, "Jphide & seek," August 1999, http://linux01.gwdg.de/~alatham/stego.html.

[27] N. Provos, "Defending against statistical steganalysis," in *Proceedings of the 10th USENIX Security Symposium*, p. 24, Washington, DC, USA, August 2001.

[28] S. Hetzl and P. Mutzel, "A graph-theoretic approach to steganography," in *Proceedings of the 9th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security (CMS '05)*, vol. 3677 of *Lecture Notes in Computer Science*, pp. 119–128, Springer, Salzburg, Austria, September 2005.

[29] "Watermarking Virtual Laboratory (Wavila) of the European Network of Excellence ECRYPT," The 2nd bows contest (break our watermarking system), 2007.

[30] A. Westfeld, "Reproducible signal processing (bows2 challenge image database, public)".

[31] Q. Liu, A. H. Sung, B. Ribeiro, M. Wei, Z. Chen, and J. Xu, "Image complexity and feature mining for steganalysis of least significant bit matching steganography," *Information Sciences*, vol. 178, no. 1, pp. 21–36, 2008.

[32] T. Pevny and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505 of *Proceedings of SPIE*, pp. 1–13, San Jose, Calif, USA, January 2007.