DRAFT PAPER WITH **LARGE FONTS**
ACCEPTED TO IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING 2012 (ICIP'2012).
SOME MINOR ERRORS HAVE BEEN CORRECTED COMPARED TO THE IEEE PUBLICATION.

STEGANALYSIS BY ENSEMBLE CLASSIFIERS WITH BOOSTING BY REGRESSION, AND POST-SELECTION OF FEATURES

Marc CHAUMONT[1,2,3] *and Sarra KOUIDER*[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
email: marc.chaumont@lirmm.fr

ABSTRACT

In this paper we extend the state-of-the-art steganalysis tool developed by Kodovský and Fridrich: the *Kodovský's ensemble classifiers*. We propose to boost the weak classifiers composing the Kodovský classifier. For this, we minimize the probability of error thanks to a regression approach of low complexity. We also propose a post-selection of features, achieved *after* the learning step of all the weak classifiers. For each weak classifier, we identify a subset of features reducing the probability of error. Both proposals are of negligeable complexity compared to the complexity of the Kodovský classifier. Moreover, these two proposals significantly increase the performance of classification.

Index Terms— Steganlaysis, Ensemble classifiers, Boosting, Features selection.

1. INTRODUCTION

During the BOSS[1] competition [2], two approaches stood out and have finished first and second respectively [3, 4].

The first approach [3] is based on the definition of a feature vector of very large dimension ($d = 33963$) and the use of a classification algorithm [5] of low complexity which is scalable vis-a-vis d dimension and the size of the training set.

The second approach [4] involves two steps. In the first step, a set of features are selected experimentally by assessing their discriminant capacity. In this step, authors use a set of linear classifiers of low complexity based on a learning by linear regression. In the second step, the most representative features are used for learning thanks to a set of weighted SVMs. Note that Gul and Kurugollu also deals with two interesting concepts: automatic selection of features, and *"training on a contaminated database"* [6] thanks to a DCT filtering of the test database.

In this paper, we pursue the study about the *Kodovský's ensemble classifiers* [5], in the scenario of *clairvoyant steganalysis* [7] (cover distribution, stego ditribution, and the payload of the message are known) without cover-source mismatch[2]. We propose to weight the votes of each weak classifier from the approach Kodovský [5]. For this, we determine those weights by minimizing the probability of misclassification. Additionally, we also propose to select features after learning.

In the section 2, we recall the important concepts of the Kodovský classifier. In section 3, we introduce the boosting by regression, and the selection of feature after learning. We give experimental results in section 4, and we conclude in section 5.

2. THE KODOVSKÝ'S ENSEMBLE CLASSIFIERS

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

The classification with *Kodovský's ensemble classifiers* is to learn separately each weak classifiers. These weak classifiers, denoted h_l with $l \in \{1,..,L\}$, take the same $\mathbf{x} \in \mathbb{R}^d$ features vector as input and returns a class number:

$$\begin{array}{rcl} h_l : \mathbb{R}^d & \rightarrow & \{0,1\} \\ \mathbf{x} & \rightarrow & h_l(\mathbf{x}) \end{array} \quad (1)$$

Each weak classifier performs its learning on a space of d_{red} dimension, with $d_{red} \ll d$. In practice, each weak clas-

This investigation was supported by the "Ministère de l'Enseignement Supérieur et de la Recherche Scientifique de la République Algérienne Démocratique et Populaire".

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

[2] This scenario is essentially that of BOSS competition.

sifier pseudo-randomly selects the features from the features vector of dimension d.

The weak classifiers are based on a classification into two classes by a Fisher Linear Discriminant (FLD) approach. In practice, this amounts to calculating the covariance matrices of $d_{red} \times d_{red}$ size, for each weak classifier, using the entire training database. From this matrix, the vector defining the separating plane of the two classes is easily deduced.

The merging of all the votes of the weak classifiers is then obtained by a majority vote such that for a $\mathbf{x} \in \mathbb{R}^d$ features vector, we have:

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

According to Kodovský [8], the classification by *ensemble classifiers* is an approach by "random forest with the FLD as a base learner instead of a random decision tree as in [9]. The randomization is in the feature subspace generation".

The approach Kodovský [5] provides performances equivalent to that of a Support Vector Machine (SVM) [10] for steganalysis of large databases with large feature vectors. Additionally, the approach Kodovský has the following properties: it is scalable vis-a-vis d dimension, it is of low computational complexity, it is of low memory complexity, and it is easily parallelizable.

3. BOOSTING BY REGRESSION, AND POST-SELECTION OF FEATURES

3.1. Boosting by regression

In the approach Kodovský, each weak classifier vote with equal prominence in the final decision; See Equation 2 and its majority vote. However, some weak classifiers are less efficient than others. It is therefore possible to use a weighted voting such that the result of classification is:

$$C(\mathbf{x}) = \begin{cases} 0 \text{ if } \sum_{l=1}^{l=L} \alpha_l h_l(\mathbf{x}) \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\ 1 \text{ otherwise,} \end{cases} \quad (3)$$

with $\alpha_l \in \mathbb{R}$ the weight associated to the h_l classifier as defined in Equation 2, and $\mathbf{x} \in \mathbb{R}^d$ a features vector.

The approach by weighted voting is a concept well known in machine learning and a way to determine the weights $\alpha_l, l \in \{1, ..., L\}$ is to use boosting approaches (eg AdaBoost [11]). The learning of the weights is done in multiple iterations by focusing on the samples (also called examples) of the base that are misclassified. For this, we assign a weight to each learning sample. At each iteration, either we resample the training set according to the sample-weights distribution, either we directly use the sample-weights in the learning process.

The integration of weights in the learning process can not be applied to weak FLD classifiers and the database resampling is computationally complex. Moreover, the boosting approaches insist on some difficult samples and therefore force some weak classifiers to dwell on these difficult samples. It is not necessarily good for the approach Kodovský because some weak classifiers have very poor performances, and therefore those weak classifiers will not improve the final result.

An approach avoiding to dwell on the samples, and also less complex is to express the problem of calculating the weights $\alpha_l, l \in \{1, ..., L\}$, as an optimization problem. We call this approach the boosting by regression. The problem is expressed by the minimization of the P_E error probability:

$$P_E = \frac{1}{N} \sum_{i=1}^{i=N} \left(f\left(\sum_{l=1}^{l=L} \alpha_l h_l(\mathbf{x}_i) \right) - y_i \right)^2, \quad (4)$$

with f a thresholding function defined by:

$$f : \mathbb{R} \rightarrow \{0, 1\} \quad (5)$$

$$x \rightarrow f(x) = \begin{cases} 0 \text{ if } x \leq \frac{\sum_{l=1}^{l=L} \alpha_l}{2}, \\ 1 \text{ otherwise.} \end{cases} \quad (6)$$

We thus looking for:

$$\{\alpha_l\} = \underset{\{\alpha_l\}}{\arg\min}\, P_E. \quad (7)$$

We are seeking a differentiable linear model in order to solve the problem with a low computational complexity. The function f (Heavyside type) must be replaced by a differentiable function. If one uses the functions $Arctan$ or $tanh$, this leads to a system of nonlinear equations whose resolution is of high computational complexity. P_E error probability (Equation 4) can also be approximated by taking $f(x) = x$ (identity function). In this case, the cancellation of the derivatives leads to the linear system:

$$\forall t \in \{1, ...L\}, \sum_{l=1}^{l=L} \alpha_l \sum_{n=1}^{n=N} h_t(\mathbf{x}_n) h_l(\mathbf{x}_n) = \sum_{n=1}^{n=N} h_t(\mathbf{x}_n) y_n \quad (8)$$

ie, $A.X = B$, with $X \in \mathbb{R}^{L \times 1}$ the vector of weights, $A \in \mathbb{R}^{L \times L}$ the symmetric matrix:

$$A_{i,j} = \sum_{n=1}^{n=N} h_i(\mathbf{x}_n) h_j(\mathbf{x}_n), \quad (9)$$

and $B \in \mathbb{R}^{L \times 1}$ the vector:

$$B_i = \sum_{n=1}^{n=N} h_i(\mathbf{x}_n) y_n. \quad (10)$$

The system is then solved thanks to a library of linear algebra.

The computational complexity comes from the filling of the A matrix and is $O(L^2 \times N)$. In approach Kodovský, the complexity of learning for a weak classifier is $O(d_{red}^2 \times N)$. In our experiments $L \ll d_{red}$. Thus, by adding the boosting by regression, the overall order of complexity remains unchanged.

3.2. Post-selection of features

A classical approach in machine learning is to select the features that are best able to classify as cover or stego. The aim is both to reduce the d dimension, but also to remove the features that can disrupt the process of classification.

In the article of Gul and Kurugollu [4], a selection of features is done by assigning to each feature a measure of correlation with the payload. Their measure is expensive in computation time. It must at first, from the same cover database, generate multiple bases of learning (each database has a different embedding payload). In a second step, it must measure the covariance between each feature and the payload. The measure, associated with each feature, defines an order of removal of components of the features vectors. Then, it must remove the features in that order and keep the set of features that give the lowest P_E probability of error. This approach has two drawbacks: it requires knowing the steganography algorithm (since it needs generate cover images at different payloads), and is also very complex in computational cost.

For the *Kodovský's ensemble classifiers*, the reduction of d dimension may seem of little importance since the classifier is scalable vis-a-vis d. Indeed, we do not need to reduce the dimension in order to obtain a learning in reasonable time[3]. For cons, the performance of each weak classifier can be improved through the selection of features. In addition, the selection of features adds an additional variability to the Kodovský algorithm because each weak classifier selects different number of features (and not always d_{red} features). This additional variability can enhance the classification model and lead to improving performances.

To keep complexity low, we do not wish to use the same principle as Gul and Kurugollu (generate multiple databases and re-run the learning). We will apply a selection process **after** the learning and we will not re-run a learning step. Thus, once a weak classifier learned, we will seek to take away some features so reduce the probability of error of the weak classifier.

Each weak classifier performs its learning on a space of d_{red} dimension, with $d_{red} \ll d$. After the learning phase of a weak classifier, we want to keep only a subset of features (the subset cardinality is lower or equal to d_{red}). There are $2^{d_{red}} - 1$ different subsets different from the empty set. It is not reasonable to test each of those subsets in order to keep the one that yields the lowest probability of error. However, in the same spirit than Gul and Kurugollu, we can define metrics evaluating the importance of a feature, and so, define an order of selection of features. With simple metrics, it is possible, with a low computational complexity, to compute the metrics, to compute an order of selection of features, and to calculate the evolution of the probability of error during the selection of features.

For each l classifier, we chose the five following metrics $c_1^{(l)}$,... , $c_5^{(l)}$, $\forall j \in \{1, .., d_{red}\}$:

$$c_1^{(l)}[j] = \frac{|\mu_1[j] - \mu_0[j]|}{\sqrt{\sigma_1^2[j] + \sigma_0^2[j]}},$$

with $\mu_0[j]$ (resp. $\mu_1[j]$) the average of the class 0 (resp. class 1) for the j^{th} feature, and $\sigma_0^2[j]$ (resp. $\sigma_1^2[j]$) the variance of the class 0 (resp. 1) for the j^{th} feature;

$$c_2^{(l)}[j] = \sum_{i=1}^{i=N} count(\mathbf{x}_i^{(l)}[j], \mathbf{w}^{(l)}[j], y_i),$$

with:

$$count(x, w, y) = \begin{cases} 1 \text{ if } [(x.w > 0 \text{ and } y = 1) \\ \quad\quad \text{or } (x.w < 0 \text{ and } y = 0)], \\ 0 \text{ otherwise}, \end{cases}$$

with $\mathbf{x}_i^{(l)} \in \mathbb{R}^{d_{red}}$, the features vector used by classifier l, and obtained by picking features from $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{w}^{(l)} \in \mathbb{R}^{d_{red}}$ the vector orthogonal to the plane separating the two classes, and $y_i \in \{0, 1\}$ the class number;

$$\begin{aligned} c_3^{(l)}[j] &= \sum_{i=1}^{i=N} \frac{count(\mathbf{x}_i^{(l)}[j], \mathbf{w}^{(l)}[j], y_i)}{\sum_{k=1}^{k=d_{red}} count(\mathbf{x}_i^{(l)}[k], \mathbf{w}^{(l)}[k], y_i)}, \\ c_4^{(l)}[j] &= corr(\mathbf{x}^{(l)}[j], y) \\ &= \frac{\sum_{i=1}^{i=N} \left(\mathbf{x}_i^{(l)}[j] - \overline{\mathbf{x}^{(l)}[j]}\right)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{i=N} \left(\mathbf{x}_i^{(l)}[j] - \overline{\mathbf{x}^{(l)}[j]}\right)^2} \sqrt{\sum_{i=1}^{i=N} (y_i - \overline{y})^2}}, \\ c_5^{(l)}[j] &= corr(\mathbf{x}^{(l)}[j].\mathbf{w}^{(l)}[j], y). \end{aligned}$$

Note that metrics c_1 and c_4 could be computed before the learning step. Each criterion $c_1, ..., c_5$, can be calculated in a single reading of the training database, and the computational complexity is $O(d_{red} \times N)$ for each weak classifier. Once the measures are calculated, we define five orders of scan on features. It costs only $O(d_{red} log(d_{red}))$. Finally, a second reading of the database is used to calculate the P_E probability of error associated to the $5 \times (d_{red} - 1)$ subsets of features. The complexity is $O(d_{red} \times N)$. Then we keep the subset of features that gives the lowest probability of error. The total order of complexity of the process of post-selection is thus $O(d_{red} \times N)$ and thus the overall order of complexity of the all classification system remains unchanged.

[3] At their experimentation for the BOSS competition [2], in late 2010, Gul and Kurugollu get on their PC, a running time of more than half a day for learning and classification by SVM, for a database of $N = 8074$ images, and $d = 1237$ features [4]. In our implementation, the learning and the classification with *Kodovský's ensemble classifiers* take less than an hour of execution for $N = 10000$ images, $d = 5330$, $L = 31$, $d_{red} = 350$, on a PC with a processor Intel(R) Core(TM) 2 Duo 8600 at 2.4 GHz with 4 GB memory. Moreover we obtain an execution time of less than 8 minutes by parallelizing the code (with library OpenMP; http://openmp.org/wp/) and using an architecture made of 8 processors *Quad-Core AMD Opteron(tm) Processor 8384*, at 2.69 GHz; Less than 30 cores are used.

4. RESULTS

Our experiments were conducted on the database BossBase v1.00 (http://www.agents.cz/boss/BOSSFinal/). This training database is made of 10000 512×512 greyscale cover images in the pgm format, and the same 10000 images embedding a message at 0.4 bpp with HUGO algorithm [1] with default parameters. We then separated the database into two parts: a training database consisting of $N = 10000$ images (5000 covers and the associated 5000 stegos) and a test database with the 10000 other images. We do not use, for the tests, the BossRank database because the problem of cover-source mismatch is not the subject of our paper.

On each image we compute $d = 5330$ features (the set of features comes from the 1458 dimensional MINMAX, [6] page 7, vector with $T = 4$, and the 3872 dimensional SUM3 vector [6] page 11) from HOLMES features [3]. We have chosen L such that $L \geq 2d/d_{red}$ so that there is a fairly good coverage of the feature vector from the *Kodovský's ensemble classifiers*. In addition, d_{red} must be large enough to cover enough features ($d/50 \leq d_{red}$) but must not be too large: 1- for there to be an enough number of observations of each features in average (for an SVM, we often choose $d < N/10$; for ensemble classifier d_{red} should probably be below $N/10$), and 2- for there have not too many contradictory features used by a weak classifier ($d_{red} \leq d/10$). By taking $L = 31$, and d_{red} in the interval $[200, 500]$, we always have $L \geq 2d/d_{red}$, and also, as shown in Figure 1, there is an local extremum at ($d_{red} = 350$, recall = $1 - P_E = 1 - 0.2336 = 76.64$ %). We thus set $L = 31$ and $d_{red} = 350$.

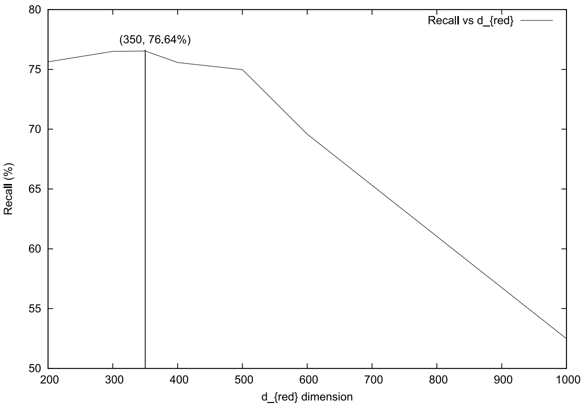

Fig. 1. Recall as a function of d_{red} with $L = 31$ weak classifiers.

We achieved five experiments for each setting and we report the average recall. For each experiment we assign differently half of the 10000 couples of cover/stego images in the training base and the rest in the test base. Furthermore, each time a different seed is used for the *Kodovský's ensemble classifiers*.

For $L = 31$ and $d_{red} = 350$ the average recall = 75.54 %. With automatic adjustment of the decision threshold (the threshold allows to position the separating plane for the two classes), for each weak classifier, by minimizing the probability of error (the algorithm is inspired by the computation of the ROC curve given on page 200 of book [12]) we get a small improvement with an average recall = 75.84 %. By only using our post-selection of features (section 3.2), we get an average recall of 76.92 %. The post-selection of features makes it possible to gain 1.4 % on the average recall. Note that, over five experiments, there is an average of 22 suppressions per weak classifier. If one performs the post-selection of features followed by an automatic adjustment of thresholds, the average recall = 76.96 %. The automatic adjustment of the thresholds is thus not necessary since it does not increase significantly the average recall and requires re-scan the entire training database. If we perform only the boosting by regression we get an average recall of 77.07 %. The boosting by regression allows an interesting gain of 1.5 %. If we apply a post-selection of features followed by a boosting by regression, the average recall increases to 77.22%. For $L = 31$ and $d_{red} = 350$, the boosting by regression and the post-selection are two approaches that increase the performance of *Kodovský's ensemble classifiers* more than 1.4 % and when assembled further increase the performance since for the five experiments the recalls are all greater than 77 %. These experiments are summarized in Figure 2, where we give the average recall under different settings.

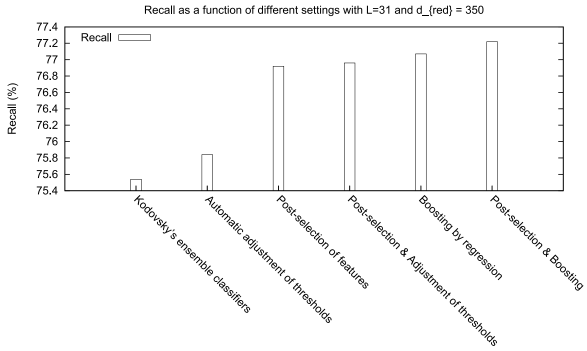

Fig. 2. Average recall for different settings. $L = 31$, $d_{red} = 350$.

We also decided to increase the number of weak classifiers in order to obtain probabilities of errors even lower. We set $L = 71$ weak classifiers, and without using the boosting or the post-selection the average recall = 76.04 %. The performance of the ensemble classifiers is slightly better than that with the first 31 weak classifiers alone (average recall was equal to 75.54 %). If we apply the post-selection of features alone[4], the average recall goes to 77.36 % and if we apply the boosting by regression alone, the average recall increases to 77.67 %. By using together the post-selection and

[4] With $L = 71$ and $d_{red} = 350$, over five experiments, there is an average of 22 suppressions per weak classifier.

the boosting, we obtain an average recall of 77.73 % which increases the average recall of *Kodovskýs ensemble classifiers* of 1.7 %. These experiments are summarized in Figure 3, where we give the average recall under different settings.

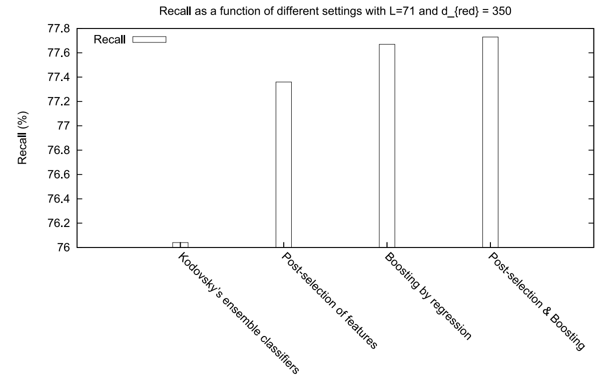

Fig. 3. Average recall for different settings. $L = 71$, $d_{red} = 350$.

In view of the difficulty to scrounge percentages during the competition BOSS [6], an average gain of 1.7 % presents a significant gain. Moreover, the proposed approach does not change the overall order of complexity.

5. CONCLUSION

In this paper, we presented a technique of boosting by regression, and a technique of post-selection, that significantly increase the efficiency of the *Kodovský's ensemble classifiers*. On training databases of 10 000 images, whose 5 000 contain a hidden message via HUGO algorithm [1], we have increased the average recall of 1.7%. This increase is significant in view of the results of the competition BOSS [6].

In addition, we believe that the *Kodovský's ensemble classifiers* can be further improved while maintaining its low complexity. It is possible to integrate other classifiers. The weak classifiers could possibly vote in ways not binary. This classification tool can also be extended. We can transform it into a multi-class version, or integrate a module treating the cover-source mismatch problem.

6. 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, “Steganalysis of Content-Adaptive Steganography in Spatial Domain,” in *Information Hiding, 13th International Conference, IH'2011*, Prague, Czech Republic, May 2011, vol. 6958 of *Lecture Notes in Computer Science*, pp. 102–117, Springer.

[4] G. Gul and F. Kurugollu, “A New Methodology in Steganalysis: Breaking Highly Undetectable Steganograpy (HUGO),” in *Information Hiding, 13th International Conference, IH'2011*, Prague, Czech Republic, May 2011, vol. 6958 of *Lecture Notes in Computer Science*, pp. 71–84, Springer.

[5] J. Kodovský and J. Fridrich, “Steganalysis in high dimensions: fusing classifiers built on random subspaces,” 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.

[6] 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.

[7] 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.

[8] J. Kodovský, “Ensemble classification in steganalysis Cross-validation and AdaBoost,” Tech. Rep., Digital Data Embedding Laboratory (DDE), Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY 13902-6000, Aug. 2011.

[9] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[10] 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.

[11] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, Aug. 1997.

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