Simple Algorithmic Modifications for Improving Blind Steganalysis Performance

Valentin Schwamberger
Max Planck Institute for Biological Cybernetics
Spemannstr. 38
72076 Tübingen, Germany
vschwamb@gmail.com

Matthias O. Franz
Institute for Optical Systems, HTWG Konstanz
Brauneggerstr. 55
78462 Konstanz, Germany
mfranz@htwg-konstanz.de

ABSTRACT

Most current algorithms for blind steganalysis of images are based on a two-stages approach: First, features are extracted in order to reduce dimensionality and to highlight potential manipulations; second, a classifier trained on pairs of clean and stego images finds a decision rule for these features to detect stego images. Thereby, vector components might vary significantly in their values, hence normalization of the feature vectors is crucial. Furthermore, most classifiers contain free parameters, and an automatic model selection step has to be carried out for adapting these parameters. However, the commonly used cross-validation destroys some information needed by the classifier because of the arbitrary splitting of image pairs (stego and clean version) in the training set. In this paper, we propose simple modifications of normalization and for standard cross-validation. In our experiments, we show that these methods lead to a significant improvement of the standard blind steganalyzer of Lyu and Farid.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Algorithms, Experimentation, Performance, Security

Keywords

Steganalysis, model selection, cross-validation, SVM

1. INTRODUCTION

Most steganalytic methods are not capable of detecting general steganographic manipulations in images (blind or universal steganalysis), since they are tuned to specific steganographic algorithms. The few currently available universal steganalytic algorithms [10, 6, 8, 1] are typically based on a two-stage approach: First, features are extracted from the tested images. Thereby, the dimensionality of the image representation and the proportion of the carrier signal is reduced such that steganographic manipulations are easier to detect. In the second stage, a previously trained classifier uses the prepared features as inputs. For instance, Support Vector Machines (SVMs) are powerful general purpose classifiers which allow for excellent results at reasonable computational costs. They are used in a number of algorithms for steganalysis [10, 6, 8, 1]. Nonlinear soft-margin SVMs (C- or ν-SVMs using a kernel) improve steganalysis performance significantly [10] at the expense of requiring careful normalization and model selection. Normalization is crucial, because the components of the feature vectors might vary significantly in their values. These SVMs contain a trade-off parameter (C or ν) for controlling the noise influence during training and typically also a kernel parameter such as the width of a Gaussian kernel or the degree of a polynomial kernel. Both parameters have to be optimized in a model selection procedure.

Cross-validation is the most common method for automatic model selection which is very successful in many applications, but standard versions thereof [4] do not consider the special properties of steganalysis training: In a training set for steganalysis, there exist pairs of images that are very similar, namely a clean image and its associated stego image which was created by embedding a message in the clean image. When a random partition of the training set is created during standard cross-validation, typically some of these image pairs are separated. Unfortunately, this makes it almost impossible for the classifier to find a suitable decision boundary between clean and stego images as we will demonstrate below. We therefore propose a simple modification of standard cross-validation, termed *paired cross-validation* which partitions always in such a way that pairs of clean and stego images are never separated.

Our experiments are based on a modified version of the well-known blind steganalyzer of Lyu and Farid [10] which we describe in the next section. In Sect. 3, we present our proposals for normalization and automatic model selection for an improved steganalysis performance. Experimental results for some important image classes and embedding types are shown in Sect. 4. Finally, we conclude with a brief summary in Sect. 5.

2. LYU & FARID'S STEGANALYZER AND MODIFICATIONS

The input of Lyu and Farid's algorithm is an image in its pixel representation. A wavelet pyramid [14, 15] is built for

MM&Sec'10, September 9–10, 2010, Roma, Italy.
Copyright 2010 ACM 978-1-4503-0286-9/10/09 ...$10.00.

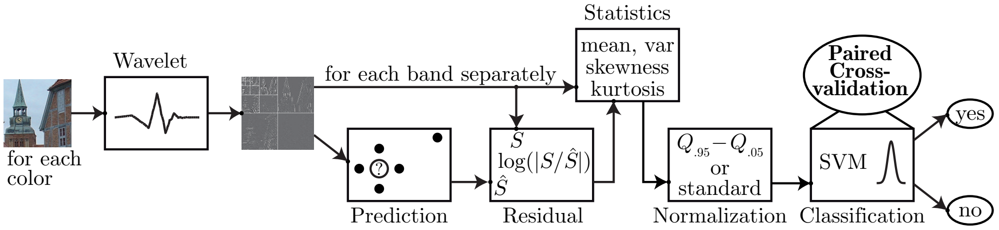

Figure 1: The algorithm—schematics

each color channel (as shown in the second block in Fig. 1) with a quadrature mirror filter of width 9 [14, 15]. We get $3(3s+1)$ subbands for an RGB image and a pyramid with s scales and three orientation subbands, i. e. diagonal, vertical, and horizontal orientation.

For predicting coefficients from their neighbors (fourth block in Fig. 1), we need to specify a neighborhood structure. The neighborhood structure is the same as in [10], i. e., the four neighboring coefficients of the same subband, parent coefficient from the larger scale and the corresponding cousins from the subbands of the same scale. In addition, it contains the corresponding central coefficient from the other color channels in their neighborhood. Due to only including the neighboring coefficients from closest orientations on the same scale (hence including horizontal and vertical coefficients for predicting the diagonal subband, but only diagonal coefficients for both the horizontal and vertical subbands), and correspondingly only one (diagonal) or two neighbors (horizontal and vertical) from the coarser scales, neighborhoods in the wavelet representation contain 9 coefficients.

The predictions are computed with linear regression applied to each subband separately, i. e., the magnitude of the central coefficient is obtained as a weighted sum of the magnitudes of its neighboring coefficients greater than a given threshold. It has been shown empirically that only the magnitudes of coefficients are correlated, and the correlation decreases for smaller magnitudes [2]. The weight sets over all subbands thus constitute the image model. In their original approach, Lyu and Farid used standard least-squares regression for this purpose. In our implementation, we use Gaussian process (GP) regression [11, 12] instead after normalizing all subband coefficients to the interval $[0, 1]$. This approach leads to slightly more robust, but essentially comparable results for the purpose of the experiments shown in this paper. GP regression needs an additional model selection step for estimating the noise content in the image. For that, we use Geisser's surrogate predictive probability [5]. It is computed on a subset of the coefficients: The finest scales are subsampled by a factor of 5 and the coarser by a factor of 3, each in both directions. Details on this regression technique can be found in [12].

Each estimator is trained and used for prediction on the same subband. Thus, training and test set coincide for this application. From the predicted coefficients $\widehat{S}$, small coefficients with amplitude below a threshold of $t = 1/255$ are set to zero. For reconstructing complete images, the algebraic signs are transferred from the original to the predicted subband coefficients. The residual r is computed by taking the logarithm of the coefficients of the input image transform S and the predicted coefficients $\widehat{S}$ and subtracting them subsequently, hence $r = \log S - \log \widehat{S}$ (fifth block).

Next, the four lowest statistical moments—i. e. mean, standard deviation, skewness, and kurtosis—of the subband coefficients (called *marginal statistics* in [10]) and of the subband residuals (called *error statistics*) are computed, again for each color and subband separately (sixth block). Finally, all these independently normalized statistics serve as feature inputs for a support vector machine [13]. In this study, we use $s = 3$ pyramid levels which results in a 120-dimensional feature vector. For our results, we normalized the resulting feature vectors with two different normalization methods described in the next section (seventh block). In the original work of Farid and Lyu [10], for the nonlinear SVM, they optimized all parameters of the SVM using a grid search (eighth block).

In our approach, the final classification was done with a 1-norm soft margin non-linear C-SVM using a Gaussian kernel. The dual optimization problem to be solved for training [13] consists of finding an optimal weight set α_i that maximizes a loss function $W(\boldsymbol{\alpha})$ under the so called box constraint

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} W(\boldsymbol{\alpha}) = \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \ , \qquad (1)$$

subject to $\quad 0 \leq \alpha_i \leq \frac{C}{m} \quad \forall i = 1, \ldots, m \ , \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \ ,$

where y_i is a label corresponding to the feature vector $\boldsymbol{x}_i$, $k(\cdot, \cdot)$ is a kernel, C a parameter that controls a trade-off between accuracy on the training set and smoothness of the solution, and m is the number of training examples $(\boldsymbol{x}_i, y_i)$. The resulting decision function of the SVM is then

$$f_{\boldsymbol{\alpha},b}(\boldsymbol{x}) = \operatorname{sgn}\left(\sum\nolimits_{i=1}^{m} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \right) \qquad (2)$$

where the offset b of the separating hyperplane can be set such that the SVM has a desired false alarm rate on the training set. This is a requirement for steganalysis, since the penalty for classifying a negative (clean) example as a false positive is generally assumed to be higher than the other way round: the penalty for classifying a manipulated image as a clean one. In other words, we want to make sure that the classifier gives alarm only if being really sure about it. Therefore the false positive rate is adapted to be below some given level: In the literature, a common rate is 1 %, or a true negative rate of 99 %, equivalently. Hence, an asymmetrical classifier with an integrated tuning of the offset b is needed. Adapting b in (2) means shifting the separating hyperplane to the class of clean images where margin errors are more and false positives are less tolerable. In doing so, the classifier can be readily set to every position on a receiver operating characteristic (ROC) curve. This asymmetry of the classifier does not only improve performance for the training of the final classifier, but for the adapted model selection stage as well. In each model selection step, the model was optimized with respect to the desired false positive rate, not only with respect to the best symmetrical accuracy. In our experiments, this affected the performance slightly for small embedding rates, and thus all results shown in this paper are created using this approach.

The choices of the soft margin parameter C and the width γ of a Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\big(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2\big) \qquad (3)$$

were based on the new paired cross-validation procedure described in the next section.

3. IMPROVEMENTS OF LYU & FARID'S STEGANALYZER

3.1 Normalization of Feature Vectors

In our experiments, we examined two different normalization methods in order to reduce the influence of large components and outliers. Both may lead to lower classification accuracies. For instance, the distance between two feature vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ in feature space which is measured by the SVM kernel can be easily dominated by a single component of high magnitude which can easily happen when higher-order statistics such as kurtosis are used. Hence, other components only have small influence.

The first approach is *standard normalization*, i. e. each component of the feature vectors $\boldsymbol{x}_i$—where i runs over the images—is normalized independently on $[0, 1]$. Hence the entries of the matrix $\mathbf{X} = (x_{ij})$ which contains the feature vectors $\boldsymbol{x}_i$ as its rows are

$$x'_{ij} = \frac{x_{ij} - \min_i(x_{ij})}{\max_i(x_{ij}) - \min_i(x_{ij})} \ . \qquad (4)$$

The vectors $\boldsymbol{x}'_i$ are the normalized vectors which serve as input to the classifier.

The second variant which we call *interquantile normalization* considers interquantile ranges, in our case values (feature vector components) in $Q_{.95} - Q_{.05}$. These interquantile ranges of each component are standardized to $[0, 1]$, exactly as in (4), while clipping values above and below to 0 or 1, respectively.

With normalization, features of high magnitude exert a less dominant influence on the results. Additionally, the detrimental effects of outliers can be reduced when using interquantile normalization.

3.2 Paired Cross-Validation

Let $Z = \{X, y\}$ be the data used for model selection, where X is the set of feature vectors and y the set of the corresponding labels. For standard k-fold cross-validation, Z is then split into k approximately equal-sized, pairwise disjoint and randomly chosen subsets $\{X_i, y_i\} = Z_i$, where $X_i \subset X$ and $y_i \subset y$. Hence $Z = \bigcup_{i=1}^{k} Z_i$.

The standard k-fold cross-validation accuracy can then be computed as the average of k accuracies. For each of the k runs, a different set Z_i is removed from Z and is used for testing instead:

$$\mathrm{CV}_k(Z) = \frac{1}{k} \sum_{i=1}^{k} \sum_{\boldsymbol{x}_i \in Z_i} \frac{y_i - f_{\alpha_i, b_i, \gamma, C}(\boldsymbol{x}_i)}{2|Z_i|} \ , \qquad (5)$$

where $f_{i,\gamma,C}$ is the decision function (see (2)) for given parameters C and γ, and the i indicates that $\boldsymbol{w}_i$ and b_i have been determined by training on the union of $k-1$ subsets of Z, namely on $Z \setminus Z_i$. In a grid search and possibly with some refinements around the maximum cross-validation accuracy, C and γ can be selected appropriately to the data.

However, when trying to automatically adapt the parameters C and γ of the SVM as described above, a special problem arises from the application to steganalysis: Low cross-validation accuracies are found, for small data sets potentially even below 50 %. For a two-class problem, the latter implies that the classification is not better than a random guess. It turns out that the effect is due to ignoring of natural pairs among the feature vectors. Natural pairs are composed of one feature vector for the clean, and another for the corresponding stego image. In feature space, they are located close to each other, since—although the carrier was mostly removed in the feature extraction stage of the algorithm—the manipulative embedding caused only a tiny difference between both feature vectors. For separating these very close data points, the decision boundary is extremely restricted and the model parameters can be optimally adapted. For standard cross-validation, the entire data set used for model selection is repeatedly divided randomly into test and training subsets without regarding natural pairs. This leads to unfavorable decision boundaries. From an image belonging to a divided natural pair, not much information is added to the classifier. For instance, consider the case where one image of such a natural pair is in the training set, and another is in the test set for a cross-validation run. Then, while facing just one image of the pair, the decision boundary in the region close to the image can be chosen rather arbitrarily because it is not very constrained. Thus, the test with the corresponding missing image of the pair is likely to fail, meaning it is likely to be treated as it would belong to the opposite class due to its opposite neighbor. Consequently, the cross-validation accuracy decreases. These unpaired examples can be regarded as counter examples for cross-validation, in the way that they give a false hint on the associated test example.

A paired form of cross-validation, which makes sure not to separate natural pairs, can suppress this influence. However, standard tools for cross-validation like the one delivered with libsvm [3] choose training and test points at random. For (5), Z_i is built without preserving natural pairs. Contrarily, pairwise means that all sets Z_i, $i = 1, \ldots, k$, contain only complete natural pairs, and correspondingly does $Z \setminus Z_i$. This application-specific adapted version of cross-validation was used in the experiments we show in the next section.

4. EXPERIMENTAL RESULTS

We tested for different compression schemes, embedding procedures, and embedding rates on an image database containing more than 1600 never compressed RGB images provided by the German Federal Office for Information Security. The Bayesian image model described in Sect. 2 was employed for determining both marginal and residual statistics for an image database containing more than 1600 natural images. They were taken by a digital camera and were never lossy compressed. This is known to be the most difficult setting for steganalysis, as the entropies of the images remain high. For instance, JPEG artifacts contained in the images from previous compression simplify steganalysis [8].

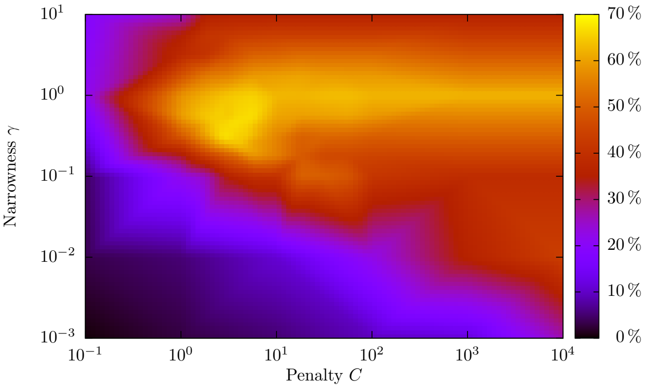

Figure 2: Paired cross-validation accuracies for different values of C and γ using a logarithmic grid for the modified steganalyzer described in Sect. 3. 50 % of the LSBs were replaced (random-spreaded).

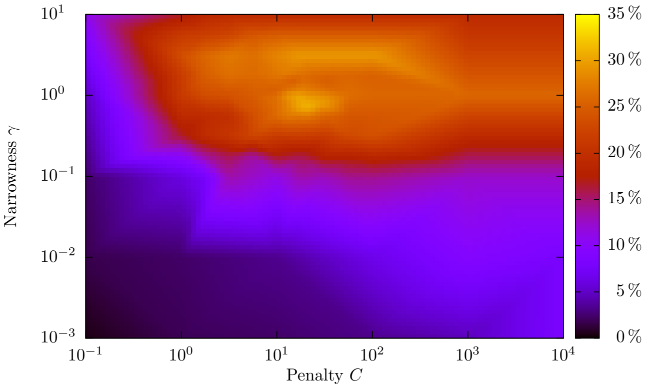

Figure 3: Paired cross-validation accuracies as in Fig. 2, but for 25 % replacement of the LSBs. One can see that the region of highest accuracies is smaller and the gradients are higher than for an embedding with a rate of 50 % of total cover capacity.

From the resulting statistical distributions, we computed the statistical moments for every color image that serve as associated feature vectors. The dimensionality of these vectors was $3 \cdot 40 = 120$. After normalizing the components of these vectors independently, we carefully selected the parameters of the support vector machine and its Gaussian kernel [3] C and γ. This was done by employing the paired 5-fold cross-validation scheme described in Sect. 3 on 1000 clean and 1000 stego images. More specifically, the model parameters γ and C are optimized for best cross-validation accuracy using a grid search with a two-step grid refinement around the maximum. The refinements led to a slight improvement of the results. By repeating the cross-validation $n_1 = 4$ times, we could average over the determined values of C and γ, which leads to a more reliable result. Furthermore, we could compute the mean CV and standard deviation $\overline{\sigma}_{\mathrm{CV}}$ for the determined cross-validation accuracies. They are stated in Tab. 1 and Tab. 2. Plots of this approach are shown in Fig. 2 and Fig. 3. We would like to point out that the region of highest cross-validation is smaller for the lower embedding rate, hence it is more difficult to adapt C and γ in this case. Subsequently, we trained the SVM on 2000 images (the training data, 1000 clean and 1000 stego images) using the values of C and γ determined in the last step, and then tested with another set of 1200 examples (600 clean and 600 stego images). For both model selection and test, we adapted the bias in order to achieve a true negative rate of at least 99 %. We randomly divided the entire set into training and test sets and averaged over $n_2 = 100$ splittings, which enabled us to estimate the standard deviation of the detection rate (true positive, "True+") on the test set. Due to adapting the SVM to a fixed false positive rate of 1 %, the results showed a considerable variability. The variances drop for lower embeddings rates, but the (relative) variation coefficients $\overline{\sigma}_{\mathrm{CV}}/\mathrm{CV}$ and $\overline{\sigma}_{+}/\mathrm{True+}$ are similar in size over all embedding rates in our experiments.

If adapting the wavelet-based classifier such that the false positive rate falls below 1 %, then 11.8 %, 12.6 %, 30.8 % and 69.5 % of the stego images can be detected for standard normalization, paired cross-validation and embedding rates of 5 % (ternary [7, 9], scanner-adaptive embedding, which aims to exploit artifacts introduced by flatbed scanners), 10 %, 25 % and 50 %, respectively. These embedding rates are the fraction of the size of the current payload and the total cover capacity, which is the size of the largest payload that can be hidden by the corresponding embedding algorithm in an image. For interquantile normalization, the corresponding detection rates are 8.6 %, 8.5 %, 14.4 %, and 65.7 %. The results, for both paired and standard cross-validation, and as well for LSB replacements and ternary embeddings are shown graphically in Fig. 4. The error bars show the associated standard deviation of the mean $\overline{\sigma}_n$. The performance of a classifier using a pair of (C, γ) determined with standard cross-validation is lower for all image classes, especially for low embedding rates where it is more difficult to grasp appropriate model parameters as described above. For higher embeddings rates, a proper choice of C and γ seems to be not as crucial. There exists one exception (interquantile normalization, 25 % embedding rate, LSB) where standard cross-validation leads to a higher true positive rate than paired cross-validation, however, it is by far not significant: The difference is only 0.1 percentage points. For high embedding rates and when using the complete feature vector, standard cross-validation can compete, but still is somewhat lower at least. For full details, see in Tab. 1 and Tab. 2. From these results, we could not identify an obvious mutual interaction between normalization and cross-validation methods.

A comparison of the two normalization approaches shows that the prediction accuracies for standard normalization are higher for most embedding rates. However, for feature vectors of reduced sizes—containing only error residuals or even a subset thereof—interquantile normalization outperforms standard normalization.

We also tested our algorithm for ±1 embeddings, which conserve parity properties of the images [9]. The results are even better. In this case, for embedding rates of 50 % and 25 %, 76.8 % and 37.1 % of the manipulated images can be revealed for standard normalization, and 81.9 % and 20.7 % for interquantile normalization.

5. CONCLUSIONS

In this paper, we presented improvements to automated model selection and feature preparation for blind steganalyzers: an adapted version of cross-validation that preserves

Table 1: Results computed with the presented asymmetrical classifier for cross-validation accuracies ("CV") and true positive rates ("True+") at a fixed rate of 99 % of true negatives. Paired cross-validation and standard cross-validation as well as standard and interquantile normalization are compared for two embedding methods (LSB replacement, ±1 embedding) and different embedding rates (10 %–50 %). "err" (no marginal statistics) and "err32" (no marginal statistics and only $3 \cdot 32 = 96$ error moments of the finest scales) indicate that only a subset of the components of the feature vectors was employed by the SVM. $\overline{\sigma} = \sigma/\sqrt{n}$ is the standard deviation of the mean over n computations, σ the standard deviation of a single realization. It is $n_1 = 4$ for the cross-validation accuracies and $n_2 = 100$ for the true positive rates.

	LSB replacement embeddings												±1 embeddings			
	50 %		50 % err		50 % err32		25 %		25 % err		10 %		50 %		25 %	
(%)	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std
CV	72.4	65.2	64.4	58.1	68.9	61.8	34.1	25.9	35.2	23.8	14.2	7.1	83.3	83.0	39.7	35.5
$\overline{\sigma}_{\mathrm{CV}}$	3.0	3.2	3.3	1.6	3.3	1.8	1.8	0.8	0.8	1.9	1.0	1.2	1.3	1.5	1.8	1.8
True+	69.5	56.5	61.0	55.1	61.7	58.0	30.8	30.4	34.5	11.6	12.6	3.5	76.8	74.6	37.1	27.1
$\overline{\sigma}_{+}$	0.9	1.9	0.9	1.3	0.9	1.32	0.7	0.7	0.5	0.6	0.2	0.1	1.4	1.5	0.6	1.2

Table 2: Same as above, but for interquantile normalization

	LSB replacement embeddings												±1 embeddings			
	50 %		50 % err		50 % err32		25 %		25 % err		10 %		50 %		25 %	
(%)	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std	pair	std
CV	74.2	63.7	70.7	59.7	70.5	66.0	25.3	20.0	35.7	27.2	9.8	5.1	87.1	80.8	30.2	29.3
$\overline{\sigma}_{\mathrm{CV}}$	2.2	3.4	2.6	3.6	2.3	2.3	0.9	1.5	0.5	0.5	0.4	0.7	1.3	3.5	1.3	2.6
True+	65.7	63.2	69.8	58.8	71.8	65.2	14.4	14.5	34.0	26.9	8.5	2.8	81.9	77.3	20.7	20.9
$\overline{\sigma}_{+}$	1.4	1.3	0.7	1.3	0.9	1.2	0.7	0.7	0.7	0.9	0.2	0.1	0.4	0.8	1.0	0.9

natural image pairs and two normalization approaches. In our experiments with the steganalyzer of Lyu and Farid, it turned out that the paired cross-validation leads to higher values of cross-validation accuracy as expected from theoretical perspective. The values of C and γ picked at this maximum of accuracy allow for a higher steganalysis performance compared to that found with standard cross-validation, especially for lower embedding rates. Furthermore, our analysis shows that normalization methods have a large influence on cross-validation accuracy and steganalysis performance. Mostly, standard normalization seems to be an appropriate choice for feature preparation. However, for a smaller dimensionality of the feature vectors, interquantile normalization is a promising alternative.

6. ACKNOWLEDGMENTS

This work was partially supported by the grant "Communication and Automation" of the Baden-Württemberg Foundation and by the German National Academic Foundation. We would like to thank J. Keppler for his help in the project, P. H. D. Le for the fruitful collaboration, and H. Schwigon of the German Federal Office for Information Security for providing us useful data and discussions.

7. REFERENCES

[1] I. Avcibas, N. D. Memon, and B. Sankur. Steganalysis using image quality metrics. *IEEE Transactions on Image Processing*, 12(2):221–229, February 2003.

[2] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12):1688–1701, December 1999.

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

[4] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, June 1975.

[5] S. Geisser and W. F. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160, March 1979.

[6] M. Goljan, J. J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. *Security, Steganography, and Watermarking of Multimedia Contents VIII*, 6072(1):1–13, February 2006.

[7] T. Holotyak, J. J. Fridrich, and D. Soukal. Stochastic approach to secret message length estimation in ±k embedding steganography. In E. J. Delp and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents*, volume 5681 of *Proceedings of SPIE*, pages 673–684, San Jose, CA, USA, 2005. International Society for Optical Engineering, SPIE.

[8] T. Holotyak, J. J. Fridrich, and S. Voloshynovskiy. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security*, volume 3677 of *Lecture Notes in Computer Science*, pages

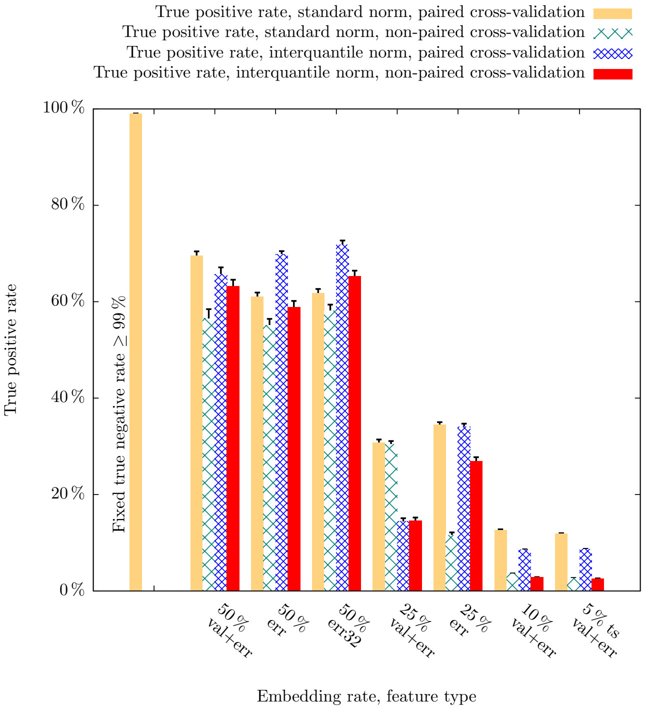

Figure 4: Comparison of classification accuracies for four distinct embedding rates (50 %, 25 %, 10 %, 5 %), for two normalization methods (interquantile, standard). The used embedding scheme is least significant bit replacement, except for "ts", which indicates ternary embedding. "err+val" denotes the usage of marginal and log error statistics, "err" denotes the utilization of log error statistics only, "err32" the utilization of the log error statistics of the 32 moments originating form the finest scales.

273–274, Berlin, Germany, September 2005. Springer-Verlag.

[9] A. D. Ker. Improved detection of LSB steganography in grayscale images. In J. Fridrich, editor, *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 97–115, Berlin, Germany, December 2004. Springer-Verlag.

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

[11] C. E. Rasmussen. Gaussian processes in machine learning. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 63–71, Berlin, Germany, October 2004. Springer-Verlag.

[12] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, MA, USA, January 2006.

[13] B. Schölkopf and A. J. Smola. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2002.

[14] E. P. Simoncelli and E. H. Adelson. Subband transforms. In J. W. Woods, editor, *Subband Image Coding.* Kluwer Academic Publishers, Norwell, MA, USA, 1990.

[15] M. Vetterli and J. Kovačević. *Wavelets and subband coding.* Prentice-Hall, Upper Saddle River, NJ, USA, 1995.