The Challenges of Rich Features in Universal Steganalysis

Tomáš Pevný[a] and Andrew D. Ker[b]

[a]Agent Technology Center, Czech Technical University in Prague,
Karlovo naměstí 13, 121 35 Prague 2, Czech Republic.
[b]Oxford University Departtment of Computer Science, Parks Road, Oxford OX1 3QD,
England.

ABSTRACT

Contemporary steganalysis is driven by new steganographic *rich feature* sets, which consist of large numbers of weak features. Although extremely powerful when applied to supervised classification problems, they are not compatible with unsupervised universal steganalysis, because the unsupervised method cannot separate the signal (evidence of steganographic embedding) from the noise (cover content). This work tries to alleviate the problem, by means of feature extraction algorithms. We focus on linear projections informed by embedding methods, and propose a new method which we call *calibrated least squares* with the specific aim of making the projections sensitive to stego content yet insensitive to cover variation. Different projections are evaluated by their application to the anomaly detector from Ref. 1, and we are able to retain both the universality and the robustness of the method, while increasing its performance substantially.

Keywords: Steganalysis, Rich Features, Dimensionality Reduction, Anomaly Detection

1. INTRODUCTION

Recent research in steganalysis has been highly successful in developing new steganalytic features,[2,3] and these have driven amazing advances in the accuracy of detection. When steganalysis is formulated as a binary classification problem — which implies that the steganalyst has complete knowledge of the steganographic channel including the steganographic algorithm used, the source of the cover images, and even the length of the possibly-hidden payload — the state of art trains ensembles of simple classifiers on *rich features*, which extract thousands or tens of thousands of measurements from each image. However, rich features have some drawbacks. First, they seem to be *fragile*, in the sense that classification performance quickly deteriorates if steganalyst is wrong in his assumptions about the channel. Second, they are *large*, with features for a reasonable-sized training set running to tens or hundreds of gigabytes of data.

When we consider a steganalyst with a more realistic scenario, the state of art is our anomaly detection method from Ref. 4. It proposed a new paradigm for steganalysis, where the focus is on the identification of a guilty actor, out of many actors each sending many images, instead of individual stego images. The steganalyst intercepts several images from multiple actors. Assuming most actors to be innocent, they estimates the average mismatch between the different actors' sources. The actor with greatest mismatch to the others is the actor suspected of using steganography, and actors can be ranked by their level of suspicion. This paradigm has two advantages over the classical approach. First, it does not suffer from cover source mismatch, because there is no training: estimation of the average cover mismatch between different actors (which can be thought of as an alternative to training) is performed after images to be analyzed have been acquired. Second, the detector is universal, because it is not trained to detect a particular steganographic method.

Unfortunately, the anomaly detector is not compatible with recently-proposed rich features. To be accurate (and universal), it requires the steganalytic features to be more sensitive to steganographic changes than to image content. Our prior work[1,4] used the 6 years old "PF274" features[5] (274 dimensional), which exhibited good performance against a wide range of steganographic methods even in realistic scenarios.[6] However, a

Further author information:
T. Pevný: E-mail: pevnak@gmail.com, Telephone: +420 22435 7608
A. D. Ker: E-mail: adk@cs.ox.ac.uk, Telephone: +44 1865 283530

embedding algorithm	linear strategy PF274	linear strategy $\mathcal{CF}^*$	greedy strategy PF274	greedy strategy $\mathcal{CF}^*$
F5	14.6	**9.5**	**21.9**	40.3
nsF5	**10.7**	23.1	**27.2**	38.0
JP	**7.8**	16.2	**23.3**	38.4
OG	**1.9**	5.7	**5.0**	26.5
SH	**2.8**	4.7	**7.4**	23.0

(a) total payload 0.1 bits per non-zero coefficient

embedding algorithm	linear strategy PF274	linear strategy $\mathcal{CF}^*$	greedy strategy PF274	greedy strategy $\mathcal{CF}^*$
F5	**4.5**	4.9	**7.4**	19.8
nsF5	**3.6**	5.1	**9.6**	26.3
JP	**4.5**	7.7	**7.1**	12.4
OG	**1.2**	1.6	**1.8**	6.3
SH	**1.4**	1.6	**2.3**	5.6

(b) total payload 0.2 bits per non-zero coefficient

Table 1: Comparison of accuracy of anomaly detector from Ref. 1, when the 274-dimensional PF274 features, and the 7850-dimensional $\mathcal{CF}^*$ features, are used with it. 100 actors each emitted 100 images, with one guilty actor spreading payload by either a linear (spread) or greedy (concentrated) strategy, using one of 5 JPEG embedding algorithms. Each experiment was repeated 100 times. The anomaly detector ranks all 100 actors and the table displays the average rank of the guilty actor: lower is better. For more details see Sections 2.1 and 2.2. The better-performing feature set is highlighted by boldface in each case.

larger number of weak features gives *inferior* performance, because the weak features collectively contain a large amount of noise (caused by cover content). This phenomenon is demonstrated in Table 1, comparing the anomaly detector's performance when it supplied with PF274 and 7850-dimensional $\mathcal{CF}^*$ features.[7] Although $\mathcal{CF}^*$ features are conclusively more sensitive than PF274 for supervised binary classification, their application in the anomaly detector is inferior in almost every case.

This work tries to marry the detection power of rich features with the universal steganalyzer, by means of feature extraction algorithms. The anomaly detector from Ref. 1 is used as a black box, and we find ways to project rich features into much smaller spaces, informed by embedding algorithm, with the aim of retaining both the universality and the robustness of the detector. In section 2 we briefly summarise the anomaly detector; section 3 covers the options for dimensionality reduction include a new method which we call *calibrated least squares*; section 4 compares these methods, and section 5 examines ways to improve their robustness to changes in the steganographic parameters. Section 6 concludes the paper.

2. BACKGROUND

2.1 Unsupervised Universal Steganalysis

The goal of universal steganalysis is to detect *any* steganographic algorithm, without knowledge of it. This problem is not very well-defined, and in practice "any" is relaxed to mean all those algorithms to which a chosen steganographic feature set is sensitive. Universal steganalysis is important for practical applications, since a detector often does not know the embedding algorithm used by the actors whose communications they monitor.

There are two types of universal steganalysis. *Supervised universal steganalysis*[8,9] is somewhat of a modification to the predominant methods of binary classification. It has a distinct training and classification phases: during training, a classifier learns the boundary between events occurring with high probability and those with low probability (see Ref. 10 for a survey). The classifier is trained only on samples from a one class: cover images. It is assumed stego objects will fall into the part of the feature-space with low probability, and therefore recognised as stego objects during the classification phase.

The drawback of the supervised universal steganalysis is the sensitivity to the type of cover images the classifier is trained on. A classifier trained on one type, and used on another, displays decreased accuracy. The extent of the decrease depends both on the classifier and the robustness of features. In this case, robustness means that the features are more sensitive to stego content than to differences between cover sources.

The only approach to *unsupervised universal steganalysis* was proposed by Ker *et al* in Ref. 4. It assumes that the detector obtains multiple images from multiple actors (users), which all are being simultaneously scrutinised for the presence of hidden payload. Unlike the supervised methods above, the goal is not to detect individual

images, but to identify a *guilty* actor, who is using steganography in some (or all) of his images. As a side note, this was also the first practical approach to the *pooled steganalysis* problem posed in Ref. 11.

The essence of the most recent version of this method,[4] which we use in this paper, is to treat the actors (not their individual images) as objects, and perform outlier analysis on them. It works as follows: (a) extract steganalytic features from all images, normalise them by using the principal component transformation*, and group them by the actor; (b) use Maximum Mean Discrepancy[12] (MMD) to calculate a distance between each pair of actors; (c) use the outlier detection algorithm Local Outlier Factor[13] (LOF) to rank each actor's degree of anomaly compared with the others. The actors with highest values of LOF are considered to be guilty. For all details see Refs. 1 and 6.

As with supervised universal steganalysis, the quality of the steganographic features is key to the performance of the method. In order for guilty actors to be more anomalous than the rest, the features must be more sensitive to stego content than to variations between actors' cover sources.

2.2 Experimental Settings

This paper deals with the *condensing* of large-dimensional features into small-dimensional projections. In order to evaluate the quality of a set of projections, we plug them into the anomaly detector and measure its accuracy. We leave it to future work to evaluate the feature sets in other applications. Below, we describe briefly the experimental settings for all experiments presented in this paper, including those in section 1.

We use a set of real-world images from a leading social network,[1] in this case 100 images from each of 3000 different actors (uploaders). These were divided into two sets: the images from 1000 of the actors were used for supervised dimensionality reduction methods described in the paper, and those from the remaining 2000 actors to evaluate the features used. For each experiment we took a random set of 100 of the 2000 "testing" actors, and selected one to be "guilty". The guilty actor is simulated by embedding payload (of which more in a moment) into some or all of their 100 images; the 99 innocent actors did no embedding. Then features from all 100×100 images were fed into the anomaly detector, the actors ranked by level of suspicion, and the rank of the guilty actor noted. Each experiment was repeated 100 times — with a different set of 100 actors and a different guilty actor — and the average rank of the guilty actors was calculated. A lower average rank is better, with an average of 1 indicating that the guilty actor is top of the list on every occasion and an average of 50.5 (out of 100 actors) indicating random guessing.

The embedding of the guilty actor is determined by three parameters: (i) the algorithm used to embed in each image, (ii) the total payload to be embedded, and (iii) the strategy used to allocate payload between their 100 images.

For (i), we use the same algorithms as our prior work:[6] five common and publicly available steganographic algorithms (F5;[14] nsF5;[15] JP Hide&Seek,[16] later abbreviated as JP; Steghide,[17] abbreviated as SH; Outguess,[18] abbreviated as OG). These were selected, like the cover images, to mimic a real-world scenario in which a non-expert uses off-the-shelf steganography software.

For (ii), total length of payload is defined with respect to a proxy for the *total* capacity of the actor's images, the total number of non-zero DCT coefficients (nzDCT) in their 100 images. The advantage of this setting is that the quantity does not depend on the capacity of individual steganographic algorithms. Payload is measured in *bits per non-zero DCT coefficient* (bpnc). In this paper we only display results for 0.1 bpnc and 0.2 bpnc payloads, which illustrate behaviour adequately.

In Ref. 6 we investigated some elementary payload allocation strategies and in this paper, for (iii), we use the two most important methods. The *linear* strategy allocates some payload into every image, in proportion to its capacity (here capacity is measured for the particular embedding algorithm used). The *greedy* (referred to in Ref. 6 as *max-greedy*) strategy uses the fewer number of images possible, by filling images to maximum steganography capacity and starting with images which have greatest capacity. In Ref. 6 it is observed that the greedy strategy is generally less detectable (for reasons which do not concern us here).

*Such whitening has been shown necessary,[1] to prevent a few features' magnitude or correlation from overwhelming the distance calculation.

Finally, when we use training data for the dimensionality reduction, this is done using the 1000 "training" actors; a random selection of 50000 images was selected (50 images from every actor), and a uniformly random-length payload embedded in each one. We emphasise that all training images underwent the embedding process, even if their selected payload is zero: for some algorithms, embedding a zero-length payload is not the same as no embedding at all. This means that the training set does not contain features extracted from true cover images. As will be seen shortly, this has consequences for the quality of the dimension reduction.

2.3 Rich Features

In this work we restrict ourselves to one of the most recent rich feature sets, known as $\mathcal{CF}^*$. These 7850-dimensional features count frequencies of adjacent filtered DCT coefficients, where the adjacency and filtering take place in various inter- and intra-block directions.[7] In Ref. 7 it is shown that $\mathcal{CF}^*$ features produce vastly superior binary classifiers for single-image steganalysis, compared with the PF274 features[†] we used in our prior anomaly detector, but Table 1 has shown that they are inferior for anomaly detector.

2.4 Aims

We use the $\mathcal{CF}^*$ features as a black box, and aim to find ways to project them to smaller dimensions. This is a form of feature reduction. The ultimate goal is improvement in the accuracy of steganalysis: supervised and unsupervised, targeted and universal. We have noted that the essence of the problem is to find features which are sensitive to stego content but insensitive to differences between covers (cover content) and cover source.

In this paper we consider only the application of features to our anomaly detector, another black box for this paper. Our first aim is *accuracy*: to maintain and increase the accuracy of the detector when the features are projected into small (≤ 50) dimensions. Our benchmark is the average rank of guilty actor. The second aim is *robustness with respect to supervised dimension reduction*: even if the dimension reduction is informed by the behaviour of one or more embedding algorithms, it should still produce features which work well when tested against novel embedding algorithms. We measure this by performing dimensionality reduction supervised by one embedding algorithm, and tested against every other.

3. DIMENSIONALITY REDUCTION METHODS

This paper describes methods of *feature condensing*, a type of dimensionality reduction which can be viewed as a search for directions in a high-dimensional space $\mathbb{R}^d$. In supervised dimensionality reduction there is a distinct training phase, in which some information is available to select directions which (in some sense) better find the information we are hoping for; in the case of unsupervised reduction, we can only hope to find structure in the test data we are presented with.

We are restricting our attention to linear projections because we believe that the steganalysis problem is *in one sense* essentially linear. We expect that hiding two bits of steganographic payload should on average involve about twice as much distortion as hiding one bit[‡], and the linearity should hold for all small payloads. Furthermore, we know that a linear relationship *does* exist, because the results of Ref. 19 demonstrated that accurate estimators of payload size can be created as linear functions of feature vectors.

However, we cannot expect that all stego objects move in the same direction, certainly not when created with different stego algorithms, nor that cover objects begin from nearby points if they arise from different sources. The boundary between cover and stego objects (for fixed payload length) may well be nonlinear (the superiority of nonlinear classifiers such as Support Vector Machine and ensemble methods suggests so). So a good set of condensed features will be a collection of different linear projections, which between them capture the behaviour of different cover objects, sources, and embedding algorithms.

The most popular method of *unsupervised* dimensionality reduction is the Principal Component Transformation (also known as the Karhunen-Loève Transformation), which projects the data so that the coordinates are

[†]The comparison is actually with the cartesian-calibrated version of PF274; a direct comparison gives an even stronger result.

[‡]Clever coding tricks can distort the linear relationship, but not between *change rate* and distortion.

not correlated. The other popular linear transformations are Independent Component Analysis and Projection Pursuits, which are not investigated here.

The art on *supervised* dimensionality reduction is surprisingly scarce. In fact, a textbook entirely devoted to feature selection, reduction, and extraction problems[20] describes only one method, which is in its essence a maximum covariance method described below. We will develop a novel supervised reduction method, specifically for steganalysis.

(We mention in passing that we cannot hope to get much help from *feature selection*, because Ref. 3 demonstrated that the rich features provide small amounts of information each, and every part of the rich feature set is necessary for good performance.)

In the rest of this section, we recapitulate some dimensionality reduction methods, including Ordinary Least Squares (OLS) which can be viewed as an extreme case producing a single projection. Our expositions are sometimes unusual, presenting them in a coherent notation and allowing us to draw comparisons and to see how well the methods align with our aim.

Throughout this section, we use the following notation. Features $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ extracted from n images are arranged in a matrix $\mathbf{X} \in \mathbb{R}^{n,d}$. It is assumed that the matrix $\mathbf{X}$ is *centered*: the mean of each columns is zero (this simplifies the notation). For supervised dimensionality reduction we need to distinguish whether features are extracted from cover images or stego images: the notation $\mathbf{X}^c$ / $\mathbf{X}^s$ denotes features extracted exclusively from cover / stego images respectively. The corresponding steganographic change rates (the number of different DCT coefficients between cover and stego, divided by the number of non-zero DCT coefficients in the cover) are stored in a column vector $\mathbf{Y}^s \in \mathbb{R}^{n,1}$.

3.1 Principal Component Transformation

The PCT can be defined as an iterative algorithm, where in k^{th} iteration one seeks a projection vector $w_k \in \mathbb{R}^d$ best explaining the data (maximising the variance) and being orthogonal to all previous projections$\{w_i\}_{i=1}^{k-1}$. This can be formulated as

$$w_k = \arg \max_{\|w\|=1} w^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} w \tag{1}$$

subject to

$$w^{\mathrm{T}} w_i = 0, \ \forall i \in \{1, \ldots, k-1\}.$$

It can be shown that w_k is k^{th} eigenvector of the matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, assuming that the vectors are sorted such that corresponding sequence of eigenvalues λ_k is non-increasing.

In practice, eigenvectors corresponding to small eigenvalues are usually discarded as they are assumed not too carry important information. In all our experiments, we kept only projections with eigenvalues at least 0.01.

3.2 Maximum Covariance Transformation

The weakness of PCT, for our application, is that it does not take into account the objective function. In our case, the objective function is the presence of steganography signalled by the steganographic change rate. This is to some extent resolved by finding a direction maximising the *covariance* between the projected data $\mathbf{X}^s w$ and the dependent variable $\mathbf{Y}^s$. Again, the vector w_k found in k^{th} iteration should be orthogonal to previous projections $\{w_i\}_{i=1}^{k-1}$. This is called the Maximum CoVariance (MCV) method. The problem is formulated as

$$w_k = \arg \max_{\|w\|=1} \operatorname{cov}(\mathbf{X}^s w, \mathbf{Y}^s) = \arg \max_{\|w\|=1} \mathbf{Y}^{s\mathrm{T}} \mathbf{X}^s w \tag{2}$$

subject to

$$w^{\mathrm{T}} w_i = 0, \ \forall i \in \{1, \ldots, k-1\}.$$

The analytical solution is $w_k = \mathbf{Y}^{s\mathrm{T}} \mathbf{X}^s{}_k$, where $\mathbf{X}_k^s = \mathbf{X}_{k-1}^s (\mathbf{I} - w_{k-1} w_{k-1}^{\mathrm{T}})$, and $\mathbf{X}_1^s = \mathbf{X}^s$.

3.3 Ordinary Least Squares Regression

Ordinary Least Square regression (OLS) finds a single direction w minimising the total square error between the projected data $\mathbf{X}^s w$ and the dependent variable $\mathbf{Y}^s$. The optimisation problem is usually formulated as

$$w = \arg\min_{w \in \mathbb{R}^d} \|\mathbf{Y}^s - \mathbf{X}^s w\|^2,$$

but to link the method to previous ones we write it as

$$w = \arg\max_{w \in \mathbb{R}^d} 2\mathbf{Y}^{s\mathrm{T}}\mathbf{X}^s w - w^{\mathrm{T}}\mathbf{X}^{s\mathrm{T}}\mathbf{X}^s w. \tag{3}$$

Unlike PCT and MCV, OLS solves an *unconstrained* optimisation problem. Similarly to MCV, it finds a projection to have high covariance with the dependent variable $\mathbf{Y}^s$ (the first term), but it also tries to reduce the variance of the projection (the second term). The second term can be therefore viewed as a regularisation removing the need for a constraint.

The analytical solution is

$$w = (\mathbf{X}^{s\mathrm{T}}\mathbf{X}^s)^{-1}\mathbf{X}^{s\mathrm{T}}\mathbf{Y}^s,$$

but this assumes that $\mathbf{X}^{s\mathrm{T}}\mathbf{X}^s$ is regular and the inversion is numerically stable. In practice this is not always true, and in fact in our application the $\mathcal{CF}^*$ features always lead to a nearly singular matrix. To alleviate this, a small diagonal matrix is added to prevent non-singularity and increase the stability of the solution:

$$w = (\mathbf{X}^{s\mathrm{T}}\mathbf{X}^s + \lambda\mathbf{I})^{-1}\mathbf{X}^{s\mathrm{T}}\mathbf{Y}^s.$$

This method is called ridge regression[21] and it is the solution of the following problem:

$$w = \arg\max_{w \in \mathbb{R}^d} 2\mathbf{Y}^{s\mathrm{T}}\mathbf{X}^s w - w^{\mathrm{T}}\mathbf{X}^{s\mathrm{T}}\mathbf{X}^s w - \lambda\|w\|^2.$$

The parameter λ acts as a regularisation constraining the complexity of the solutions. To achieve the best results one should make a search for an optimal value of λ, but our only goal here is to prevent inversion of near-singular matrices so we fixed the parameter at $\lambda = 10^{-7}$. In this paper when we mention OLS we in fact mean ridge regression with a very small ridge.

It is indeed possible to turn OLS into an iterative algorithm by putting the same orthogonality constraint as in PCT and MCV, and repeat the process. But doing so does not give us new vectors (except for some numerically-insignificant noise left over from the ridge) since w found in the first iteration already contains all the linear information about $\mathbf{X}^s$ in $\mathbf{Y}^s$.

3.4 Calibrated Least Squares Regression

Recall that good steganographic projections should be *sensitive* to embedding changes yet *insensitive* to the image content. This means that the covariance between the projection of stego features and their embedding change rate should be high, while the variance of projection of cover features should be low. This objective is not optimised in any of the above algorithms.

Therefore we propose to find the projections be iteratively solving the following problem, which we call *calibrated least squares* (CLS). In the k^{th} iteration, the algorithm solves following problem

$$w_k = \arg\max_{w \in \mathbb{R}^d} 2\mathbf{Y}^{s\mathrm{T}}\mathbf{X}^s w - w^{\mathrm{T}}\mathbf{X}^{c\mathrm{T}}\mathbf{X}^c w - \lambda\|w\|^2, \tag{4}$$

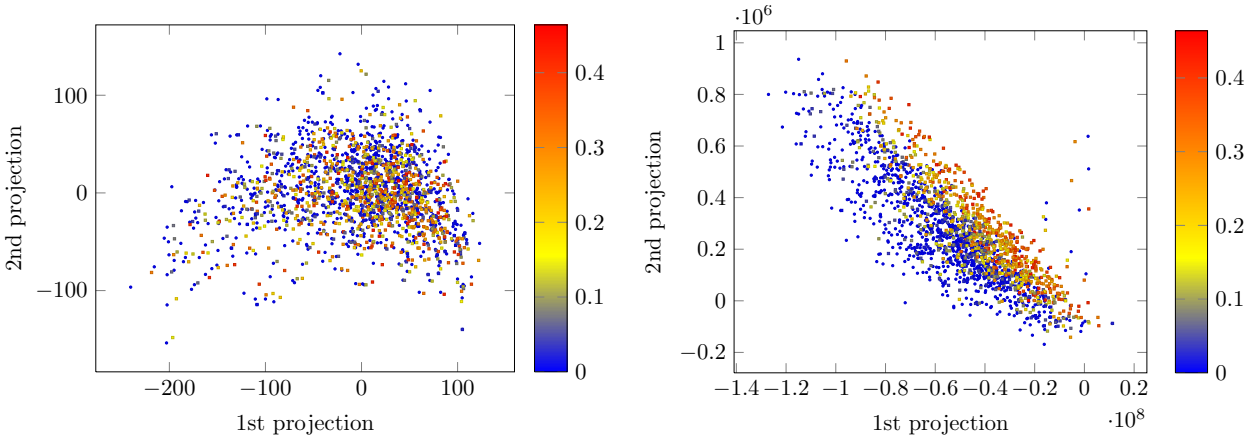
subject to

$$w^{\mathrm{T}} w_i = 0,\ \forall i \in \{1, \ldots, k-1\}.$$

The problem (4) is similar to the one being solved in OLS, but it reflects more closely our goal. As before, we stabilise the numerical inversion with a ridge-like parameter $\lambda = 10^{-7}$. The analytical solution of (4) is equal to

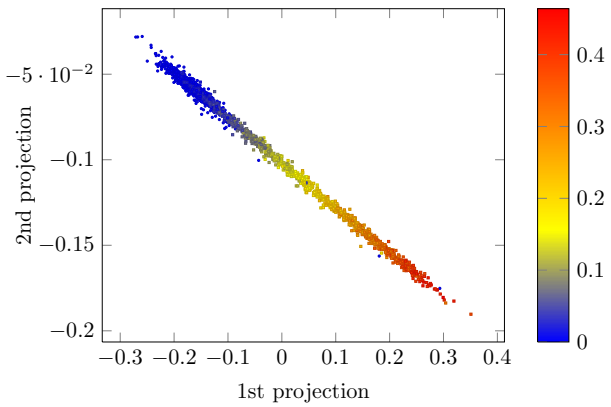
$$w_k = (\mathbf{X}_k^{c\mathrm{T}}\mathbf{X}_k^c + \lambda\mathbf{I})^{-1}\mathbf{X}_k^{s\mathrm{T}}\mathbf{Y}^s, \tag{5}$$

where $\mathbf{X}_k^s = \mathbf{X}_{k-1}^s(\mathbf{I} - w_{k-1}w_{k-1}^{\mathrm{T}})$ with $\mathbf{X}_1^s = \mathbf{X}^s$, and $\mathbf{X}_k^c = \mathbf{X}_{k-1}^c(\mathbf{I} - w_{k-1}w_{k-1}^{\mathrm{T}})$ with $\mathbf{X}_1^c = \mathbf{X}^c$.

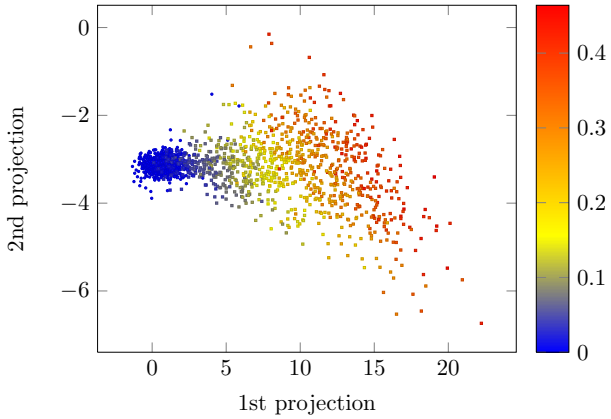


(a) Principal component transformation

(b) Maximum covariance method



(c) Ordinary least squares regression



(d) Calibrated least squares regression

Figure 1: First two condensed features of PCT, MCV, OLS, and CLS methods, from the testing set. The embedding algorithm used for condensing features and creating stego images was nsF5. The colour indicates the steganographic change rate.

4. COMPARISON OF FEATURE CONDENSING METHODS

We have explained the merits of different algorithms for finding projections. We can illustrate their behaviour by taking the nsF5 testing data set, which has uniformly random payload sizes, and projecting it onto the first two components found by PCT, MCV, OLS (in the iterative version which finds more than one projection) and CLS, and these are displayed in Figure 1. The supervised dimensionality reduction was also performed with nsF5 data.

We have noted that PCT (Section 3.1) focuses on the explanation of the variance in the data, which is not exactly on par with our goal, as the variance in features can be dominated by cover image content. In Figure 1(a) we see that the 2-dimensional projection has decorrelated the data, but has not identified the stego signal at all. MCV (Section 3.2) finds directions maximally correlated with the explained variable (change rate). Although this is better than PCT, the image content acting as noise is not suppressed. This is observed in Figure 1(b), which shows a clear correlation between payload size and the feature projection, but still a lot of noise. OLS (Section 3.3) maximises the covariance of a projection with the payload and minimises the variance of the projection. Since the covariance and variance of projections are measured on the same data, it is not clear whether the variance comes from image content or from embedding changes. This ambiguity is removed in the proposed CLS method (Section 3.4). Figure 1(c) and (d) shows that OLS identifies stego content very well, but only 1 dimension of information is available (the 2nd projection mirrors the first almost exactly), whereas CLS has managed to identify two distinct dimensions, both of which correlate with payload. It also shows that a "cluster" of cover images has been identified.

4.1 Accuracy and Robustness

We now test these theories by using condensed feature sets in our anomaly detector. The steganalyst will use the unsupervised universal steganalyser[1] with $\mathcal{CF}^*$ features condensed by PCT (producing approximately 3000 projections with eigenvalue at least 0.01), MCV (50 projections), OLS (one projection), and CLS (50 projections). The detector's accuracy, measured by the average rank of the guilty actor, is shown in Table 2. Columns correspond to the condensing method and the steganographic algorithms used for supervised condensing, while rows correspond to the embedding algorithm used by the guilty actor. Because we did not know the optimal number of condensed features to use, the numbers are those giving the best results on the testing set. Although this would not be possible in real-world applications, at this point we want to compare the methods at their best. The problem of determining a suitable number of condensed features will be addressed in the next subsection.

The results in Table 2 confirm our expectations. The unsupervised PCT method (which corresponds to the best method of Refs. 1 and 6; recall that the same whitening was already used by the anomaly detector) is worst. The MCV method is inferior to OLS in most situations, and CLS is significantly superior to the others. It is also notable that the linear embedding strategy (spread payload between all images) is most detectable by PCT features (which parallels the results of Ref. 6), but this is no longer the case for CLS-condensed features. In Ref. 6 the poorer detection of the greedy embedding strategy (concentrate payload into fewest images) was identified as a weakness of the anomaly detector, and explained by noise in the features. This weakness seems to have been removed by CLS feature condensing.

One of the biggest concerns in the supervised feature condensing is over-fitting a particular embedding algorithm, which would negate the universality of the detector. By condensing on one embedding algorithm, and testing on another, we are able to measure the extent of over-fitting. As one would expect, the fact that OLS only creates a single projection makes it vulnerable to over-fitting, and this is observed in Table 2. To some extent it is also observed in MCV. Features condensed by using training data from Outguess and Steghide (columns captioned "OG" and "SH") produce accurate identification of the guilty actor if they use the same algorithms. But if they use a different algorithm the accuracy is not much better than random guessing. However, CLS-condensed features do not suffer from substantial over-fitting, and the off-diagonal entries in the blocks of Table 2 still show good accuracy. We believe that this robustness stems from the minimisation of variance on cover images.

There is an interesting phenomenon which occurs in the case of F5, with the linearly-spread payload. When F5 is used both for condensing the features, by OLS and MCV, and for testing, the condensed features have

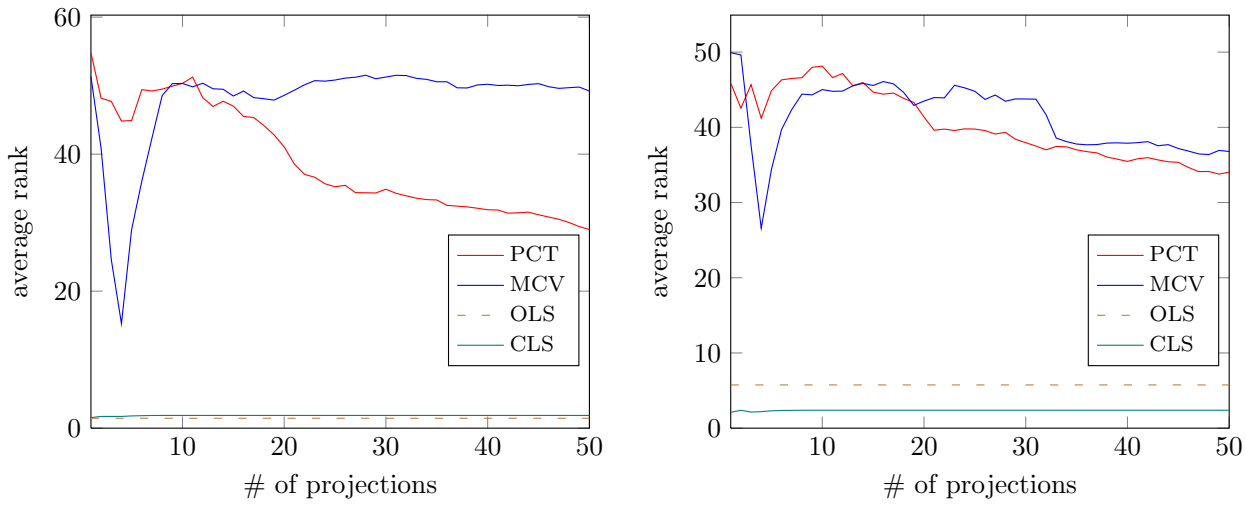
	PCT	MCV trained on					OLS trained on					CLS trained on				
		F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	40.3	23.4 (4)	28.4 (4)	31.6 (5)	30.7 (50)	28.9 (47)	22.2	4.4	28.1	41.0	25.6	**1.6** (1)	**1.9** (1)	**8.8** (1)	**6.6** (4)	**4.5** (3)
nsF5	38.0	25.3 (4)	26.6 (4)	29.8 (5)	31.8 (42)	28.9 (48)	31.6	5.8	30.2	41.2	33.8	**1.8** (1)	**2.1** (1)	**10.1** (1)	**10.9** (4)	**10.5** (3)
JP	38.4	34.7 (4)	36.2 (4)	27.2 (5)	33.8 (50)	34.4 (47)	38.8	35.5	6.9	47.1	47.6	**8.9** (1)	**7.2** (2)	**1.7** (1)	**15.5** (2)	**10.5** (2)
OG	26.5	23.2 (44)	19.2 (50)	18.4 (50)	31.6 (4)	3.2 (6)	26.4	23.0	44.9	2.4	1.3	**3.7** (1)	**3.0** (6)	**11.8** (2)	**1.2** (1)	**1.1** (1)
SH	23.0	21.9 (44)	18.7 (50)	17.5 (50)	30.3 (4)	2.6 (6)	31.4	31.7	45.9	3.5	1.3	**5.2** (1)	**3.2** (6)	**9.1** (2)	**1.2** (1)	**1.1** (1)

(a) The guilty actor embeds total payload of 0.1bpnc, using the greedy strategy.

	PCT	MCV trained on					OLS trained on					CLS trained on				
		F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	9.5	27.1 (50)	22.4 (48)	**20.0** (49)	18.7 (50)	18.1 (50)	55.9	2.4	41.6	48.3	24.0	**1.8** (1)	**1.8** (1)	25.9 (4)	**9.7** (50)	**5.4** (2)
nsF5	23.1	14.8 (4)	15.2 (4)	18.3 (5)	22.4 (46)	23.0 (48)	7.9	**1.4**	8.9	23.4	21.3	**1.4** (1)	1.5 (1)	**2.2** (1)	**5.3** (2)	**4.3** (2)
JP	16.2	33.8 (50)	30.2 (50)	37.4 (22)	30.9 (48)	35.3 (49)	29.3	36.7	21.3	44.1	39.3	**15.6** (3)	**11.4** (5)	**1.5** (4)	**11.2** (50)	**17.6** (44)
OG	5.7	25.8 (50)	18.4 (50)	13.5 (50)	26.9 (3)	1.6 (6)	13.6	18.3	35.4	1.7	1.2	**3.1** (1)	**2.6** (6)	**4.9** (2)	**1.1** (1)	**1.1** (1)
SH	4.7	19.6 (50)	15.0 (50)	11.2 (49)	27.5 (3)	1.5 (6)	16.2	15.6	31.4	1.7	1.3	**3.1** (3)	**2.7** (5)	**2.6** (1)	**1.1** (1)	**1.1** (1)

(b) The guilty actor embeds total payload of 0.1bpnc, using the linear strategy.

Table 2: Average rank of guilty actor out of 100 actors (lower is better; 1.0 represents perfect detection and 50.5 random guessing), when $\mathcal{CF}^*$ features are condensed using PCT, MCV, OLS, and CLS methods. The PCT method is unsupervised and extracts all projections with eigenvalue at least 0.01, and the others create condensed features using data informed by each of the five different embedding algorithms separately. Each row corresponds to the guilty actor using a different embedding algorithm; the experimental settings are described in Section 2.2. For the MCV and CLS methods, the number of projections selected is the value up to 50 which gives the best testing accuracy and is displayed in parentheses below the average rank; OLS always creates one projection. The PCT, MCV, OLS, and CLS methods are compared for every combination of training/embedding algorithm, and the one with best performance highlighted in boldface.



(a) Guilty actor embeds 0.1bpnc using the linear strategy. (b) Guilty actor embeds 0.1bpnc using the greedy strategy.

Figure 2: Average rank of guilty actor, where up to 50 features are condensed by PCT, MCV, OLS, and CLS: the training and testing sets were created using the nsF5 embedding algorithm. Although OLS creates only one feature, it is plotted as a horizontal line.

worse performance than PCT, even as bad as random guessing in the case of OLS. This is counterintuitive, since condensing provides additional information to the steganalyzer. Indeed, it is better to train on the nsF5 algorithm instead of the F5 algorithm, even when the steganographer is using F5. The cause of this strange behaviour is the implementation of the F5 algorithm, which causes stego images to have different characteristics from cover images. This behaviour is explained in detail in Appendix A. Notice that CLS is immune to this because it uses cover features as well as stego features during the condensing process.

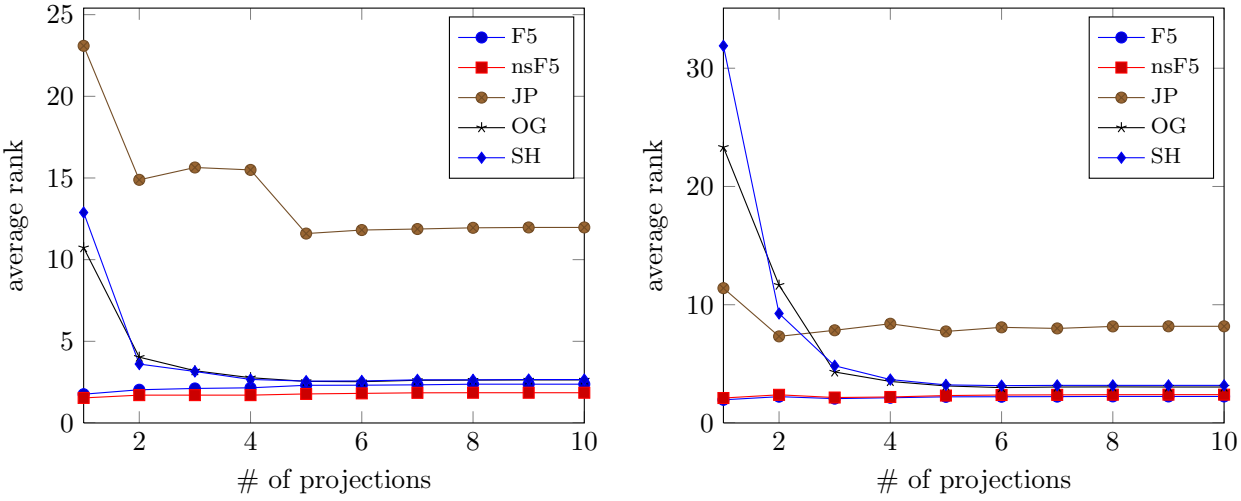
Finally, comparison of Tables 1 and 2 shows that CLS-condensed $\mathcal{CF}^*$ features far outstrip the PF274 features in accuracy for anomaly detection (including, in almost every case, when they were created using the wrong algorithm). Thus we have indeed been able to marry the power of rich features with unsupervised universal steganalysis.

4.2 Number of Condensed Features

The results shown in Table 2 were achieved using the *optimal* number of projections determined on the testing set (up to 50). Obviously, this "cheating" cannot be used in practice, but we will now show that the performance is, in fact, rather insensitive to the number of projections.

The anomaly detector was run with each number of features from 1 to 50, and its accuracy — the average rank of guilty actor — is plotted as the number of condensed features varies in Figure 2, for both embedding strategies. In this plot the embedding algorithm used for condensing, and by the guilty actor, were always nsF5. We do not display graphs for other embedding methods to avoid cluttering the paper, but they were of similar shapes. For PCT, the graph shows that more projections are better. This was already discussed in Ref. 6, and the phenomenon is caused by the lack of information about the steganographic changes being concentrated in any particular projection. On the other hand, MCV achieves the best performance when using the first few projections only, and then its performance decreases. This is most likely caused by the over-fitting, as MCV does not have any regularisation. The accuracy of CLS almost does not change after the first projection. This is caused by the fact that guilty actor used the each same embedding algorithm as was used for the condensing of the features, and the optimum of (4) is achieved on the first projection.

Figure 3 allows us to study effects of mismatch between algorithm used for condensing (again nsF5) and algorithms used by the guilty actor. The graph is shown only for the CLS, because it is superior to all other



(a) Guilty actor embeds 0.1bpnc using the linear strategy. (b) Guilty actor embeds 0.1bpnc using the greedy strategy.

Figure 3: Average rank of guilty actor where the training data is mismatched with the testing data, where up to 50 features are condensed by CLS: the training set was created using the nsF5 embedding algorithm, and the five different embedding algorithms are used for the testing set.

feature condensing methods. The first projection is sufficient when there is no mismatch between the algorithms, but in the case of mismatch more projections improve performance. Five projections represents performance close to the optimum in each case (including examples not shown here to avoid cluttering the paper). The important property of CLS is that using a few more projections than optimal does not decrease the performance of the universal steganalyzer.

5. IMPROVING ROBUSTNESS

The conclusion of the previous section is that calibrated least squares dominates, followed by ordinary least squares regression. We now investigate whether the accuracy of universal steganalysis can be improved by increasing diversity among the projections.

From the literature we have identified two complementary strategies to do so. *Bagging*[22] creates diversity by repeating the condensing algorithm multiple times, each time on a random subset of available samples, and concatenating the projections from each run. We have modified bagging to work on the level of actors rather than on individual images: every training set contained 50 images from 256 randomly selected actors out of the 1000 actors in the training data.

The second approach, *ensemble*, creates diversity by repeating the condensing algorithm using different subsets of the full feature set, and again concatenating the results.[23] We randomly choose 2048 out of 7850 features each time. This approach has been already successful in targeted steganalysis.[7]

We have tested the effect of both approaches by using CLS and OLS. Each time we created 50 projections by (a) executing OLS fifty times, or (b) executing CLS ten times and taking the first five projections of each run. As in the previous section, we chose the initial sequence of these 50 projections with the best performance on the testing set. The evaluation conditions were also exactly the same as in the previous section. Table 3 shows the average ranks of the guilty actor, hiding 0.1bpnc using the greedy embedding strategy. We do not show results for the linear strategy, since they are essentially the same.

The results show that neither bagging nor ensemble helps significantly when used with the CLS method, at least with the parameters we tested. We should not exclude the possibility that they might boost performance with different parameters (number of projections per run, number of actors/features used) but we performed

	ensemble projections trained on					bagged projections trained on					standard CLS trained on				
	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	1.8	2.1	**6.1**	8.0	6.7	1.7	2.4	9.6	7.4	8.2	**1.6**	**1.9**	8.8	**6.6**	**4.5**
	(1)	(1)	(2)	(33)	(21)	(1)	(2)	(1)	(1)	(25)	(1)	(1)	(1)	(4)	(3)
nsF5	2.0	2.2	**7.3**	**10.4**	**9.2**	2.3	3.2	9.9	17.8	21.7	**1.8**	**2.1**	10.1	10.9	10.5
	(1)	(1)	(2)	(45)	(21)	(1)	(23)	(1)	(47)	(27)	(1)	(1)	(1)	(4)	(3)
JP	**7.0**	9.9	1.8	19.1	10.7	12.4	10.0	2.0	27.4	16.2	8.9	**7.2**	**1.7**	**15.5**	**10.5**
	(1)	(12)	(1)	(11)	(1)	(1)	(30)	(3)	(50)	(1)	(1)	(2)	(1)	(2)	(2)
OG	2.9	3.9	**6.3**	1.2	1.1	**2.5**	5.1	16.7	1.4	1.2	3.7	**3.0**	11.8	**1.2**	**1.1**
	(4)	(43)	(40)	(1)	(1)	(22)	(21)	(10)	(3)	(1)	(1)	(6)	(2)	(1)	(1)
SH	2.9	3.6	**5.2**	1.2	1.1	**2.8**	5.9	12.5	1.4	1.2	5.2	**3.2**	9.1	**1.2**	**1.1**
	(33)	(41)	(40)	(1)	(1)	(23)	(21)	(10)	(2)	(3)	(1)	(6)	(2)	(1)	(1)

(a) Bagging and ensemble versions of the CLS method (10 repetitions of 5 projections)

	ensemble projections trained on					bagged projections trained on					standard OLS trained on				
	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH	F5	nsF5	JP	OG	SH
F5	18.9	**4.0**	**19.4**	**24.1**	**21.0**	**14.3**	4.3	22.5	32.1	23.9	22.2	4.4	28.1	41.0	25.6
	(10)	(1)	(8)	(2)	(29)	(2)	(1)	(6)	(6)	(18)	(1)	(1)	(1)	(1)	(1)
nsF5	25.8	**4.9**	**19.0**	**23.2**	**25.4**	**22.8**	6.8	26.7	33.7	32.9	31.6	5.8	30.2	41.2	33.8
	(9)	(1)	(11)	(2)	(2)	(7)	(1)	(27)	(6)	(18)	(1)	(1)	(1)	(1)	(1)
JP	45.4	**30.0**	**4.9**	44.1	44.0	39.5	31.7	5.4	**40.3**	**33.9**	**38.8**	35.5	6.9	47.1	47.6
	(15)	(12)	(1)	(12)	(21)	(44)	(22)	(1)	(25)	(25)	(1)	(1)	(1)	(1)	(1)
OG	**9.7**	**9.1**	**8.8**	**2.3**	1.3	18.0	13.3	15.8	3.5	1.3	26.4	23.0	44.9	2.4	1.3
	(41)	(22)	(36)	(1)	(1)	(21)	(46)	(47)	(2)	(1)	(1)	(1)	(1)	(1)	(1)
SH	**9.4**	**9.2**	**8.4**	**3.2**	1.3	18.2	12.4	16.9	4.5	**1.3**	31.4	31.7	45.9	3.5	1.3
	(50)	(46)	(37)	(1)	(1)	(44)	(46)	(42)	(5)	(1)	(1)	(1)	(1)	(1)	(1)

(b) Bagging and ensemble versions of the OLS method (50 single projections)

Table 3: Average rank of guilty actor when the $\mathcal{CF}^*$ features are condensed using standard, bagging, and ensemble versions of CLS and OLS. The guilty actor embeds total payload of 0.1bpnc, using the greedy strategy.

some exploratory experiments in this direction, with little success. In some scenarios in Table 3(a) the bagging method achieves a slight increase in robustness to condensing supervised by the wrong embedding method. But in most cases the projections found by a single execution of the CLS method, on all available training data, gives better performance. CLS projections are generally already quite robust.

Bagging (but not ensemble) does improve the accuracy when the condensing algorithm is OLS (Table 3 (b)). However, the accuracy still does not match that of condensed features found by a single execution of the CLS algorithm.

From the results in Tables 2 and 3, one might conclude that the most secure embedding algorithm, out of the tools we tested, is JPHide&Seek because the guilty actor is less often ranked among the most suspicious. However, we do not agree: its detectability is similar to the other methods when the feature condensing is trained with the same algorithm, and the apparent "security" must be due to an embedding mechanism which is different to the other algorithms. A suitably-crafted training set (perhaps bagging using different embedding methods) will alleviate this.

6. CONCLUSIONS

The paradigm of *rich features* — very large numbers of features each conveying weak information about steganographic payload — has excelled in the context of supervised steganalysis, where the information can be focused during the training stage. But it is not suitable for unsupervised methods, unless the signal-to-noise ratio can be improved. We have posed this as a feature reduction problem, which selects projections to condense the steganographic information in large feature sets down to a much smaller number. We have suggested that this must be achieved by a supervised technique. The challenges include obtaining sufficient variety in the projections to make them robust to testing on embedding algorithms other than where they were trained, and to variation between cover sources.

Our experimental results are limited to the anomaly detector of Ref. 1, where we have been able to create detectors based on reduction of $\mathcal{CF}^*$ features to just a handful of important projections, showing substantial improvement over prior art.

Using the results of supervised feature reduction method means that the complete anomaly detector is no longer unsupervised. However, we have demonstrated that the features show a high level of robustness to varying the embedding algorithm, meaning that the anomaly detector retains its universal behaviour. Finding the balance between supervision (to focus information) and universality (for robustness) is an important challenge for steganalysis in general. Another approach would be to find "general" embedding operations which simulate the behaviour of a wide range of embedding algorithms (which, for the most part, only increment or decrement DCT coefficients or pixels).

There are other dimensionality reduction methods which we did not test, but we demonstrated that *calibrated least squares* is very closely aligned with our objective of enhancing steganographic information whilst suppressing cover noise. The field of unsupervised dimensionality reduction has recently seen some advances driven by "big data"[24] and it may be worthwhile to pursue similar methods in steganalysis. But we are convinced that steganalysis features have some practically-linear properties which make simple methods such as CLS and OLS bagging a more promising line of research.

ACKNOWLEDGMENTS

The work on this paper was supported by European Office of Aerospace Research and Development under the research grant numbers FA8655-11-3035 and FA8655-13-1-3020. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of EOARD or the U.S. Government.

The work of T. Pevný was also supported by the Grant Agency of Czech Republic under the project P103/12/P514.

REFERENCES

[1] Ker, A. D. and Pevný, T., "Identifying a steganographer in realistic and heterogeneous data sets," in [*Media Watermarking, Security, and Forensics XIV*], Memon, N., Alattar, A., and Delp III, E., eds., *Proc. SPIE* **8303**, 0N01–0N13, SPIE (2012).

[2] Fridrich, J. and Kodovsky, J., "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security* **7**(3), 868–882 (2012).

[3] Kodovsky, J. and Fridrich, J., "Steganalysis of JPEG images using rich models," in [*Media Watermarking, Security, and Forensics XIV*], Memon, N., Alattar, A., and Delp III, E., eds., *Proc. SPIE* **8303** (2012).

[4] Ker, A. D. and Pevný, T., "A new paradigm for steganalysis via clustering," in [*Media Watermarking, Security, and Forensics XIII*], Memon, N., Dittmann, J., Alattar, A., and Delp III, E., eds., *Proc. SPIE* **7880**, 0U01–0U13, SPIE (2011).

[5] Pevný, T. and Fridrich, J., "Merging Markov and DCT features for multi-class JPEG steganalysis," in [*Media Watermarking, Security, and Forensics IX*], Delp III, E. J. and Wong, P. W., eds., *Proc. SPIE* **6505**, 03–14, SPIE (2007).

[6] Ker, A. D. and Pevný, T., "Batch steganography in the real world," in [*Proceedings of the 14th ACM Multimedia & Security Workshop*], 1–10, ACM (2012).

[7] Kodovsky, J., Fridrich, J., and Holub, V., "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security* **7**(2), 432–444 (2012).

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

[9] Pevný, T. and Fridrich, J., "Novelty detection in blind steganalysis," in [*Proceedings of the 10th ACM Multimedia & Security Workshop*], Ker, A. D., Dittmann, J., and Fridrich, J., eds., 167–176 (2008).

[10] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)* **41**(3), 1–58 (2009).

[11] Ker, A. D., "Batch steganography and pooled steganalysis," in [*Proceedings of the 8th Information Hiding Workshop*], Camenisch, J. L., Collberg, C. S., Johnson, N. F., and Sallee, P., eds., *LNCS* **4437**, 265–281, Springer (2006).

[12] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J., "A kernel method for the two-sample problem," in [*Advances in Neural Information Processing Systems 19*], 513–520, MIT Press (2007).

[13] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., "LOF: Identifying density-based local outliers," in [*Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*], 93–104, ACM (2000).

[14] Westfeld, A., "F5-a steganographic algorithm," in [*Proceedings of the 4th Information Hiding Workshop*], Moskowitz, I., ed., *LNCS* **2137**, 289–302, Springer (2001).

[15] Fridrich, J., Pevný, T., and Kodovský, J., "Statistically undetectable JPEG steganography: dead ends challenges, and opportunities," in [*Proceedings of the 9th ACM Multimedia & Security Workshop*], 3–14, ACM (2007).

[16] Latham, A., "Implementation of the JPHide and JPSeek algorithms ver 0.3 (released August 1999)." `http://linux01.gwdg.de/~alatham/stego.html` (last accessed April 2012).

[17] Hetzl, S. and Mutzel, P., "A graph-theoretic approach to steganography," in [*Proceedings of the 9th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security*], *LNCS* **3677**, 119–128, Springer (2005).

[18] Provos, N., "Defending against statistical steganalysis," in [*Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*], 323–335, USENIX Association (2001).

[19] Pevný, T., Fridrich, J., and Ker, A. D., "From blind to quantitative steganalysis," *IEEE Transactions on Information Forensics and Security* **7**(2), 445–454 (2012).

[20] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A., [*Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*], Springer (2006).

[21] Hoerl, A. E. and Kennard, R. W., "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics* **12**(1), 55–67 (1970).

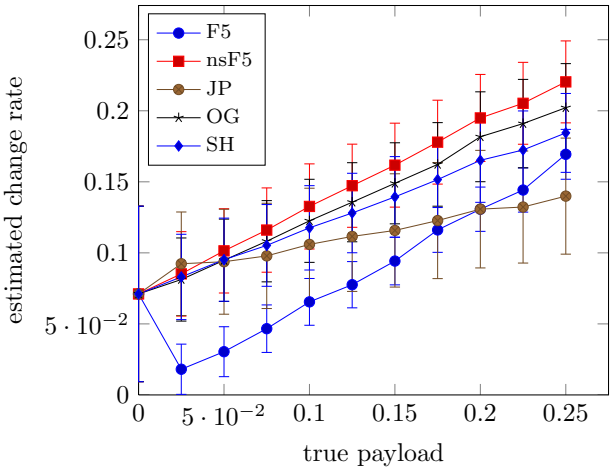


Figure 4: The graphs show median and interquartile range of the change rate estimates using an OLS estimator trained on F5 examples. Five different embedding algorithms are tested, each at different change rates.

[22] Breiman, L., "Bagging predictors," *Machine Learning* **24**(2), 123–140 (1996).

[23] Lazarevic, A. and Kumar, V., "Feature bagging for outlier detection," in [*Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*], *KDD '05*, 157–166 (2005).

[24] Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y., "Building high-level features using large scale unsupervised learning," in [*Proceedings of the 29th International Conference on Machine Learning*], Omnipress (2012).

[25] Westfeld, A., "Implementation of the F5 steganographic algorithm (released May 2011)." `http://code.google.com/p/f5-steganography/` (last accessed April 2012).

APPENDIX A. ARTIFACTS OF F5 IMPLEMENTATION

We investigate the anomalous behaviour of the F5 algorithm, which is exemplified by the behaviour of an OLS quantitative estimator similar to that in Ref. 19. Figure 4 shows the median and interquartile range of estimated change rates, plotted against the true payload, when the estimator is trained on F5 examples.

The estimator works well in the sense that the estimated change rate increases linearly with the payload, when testing all five steganographic algorithms. Notice, though, that the estimator has a positive bias of approximately 0.05, which resolves for the F5 method (but not the others) at above-zero embedding rates. The cause of this bias is the unusual implementation of JPEG compression used by the F5 software[25] (written in Java). Even at zero payload, there is a decompression step followed by recompression (using different code) which introduces artefacts into the objects.

The OLS estimator was trained exclusively on images embedded by the F5 algorithm (and the same happens when we condense features using F5 training examples) so it has learned the source of stego images only. In testing the cover images, used to estimate accuracy at payload 0, have different characteristics. So this rather weird behaviour is actually an example of cover source mismatch. It is also responsible for the poor performance of features condensed by OLS on samples of the F5 algorithm (see Table 2). We emphasise that CLS is robust against this problem, because it uses cover genuine samples for regularisation.