GENERALIZED TRANSFER COMPONENT ANALYSIS FOR MISMATCHED JPEG STEGANALYSIS

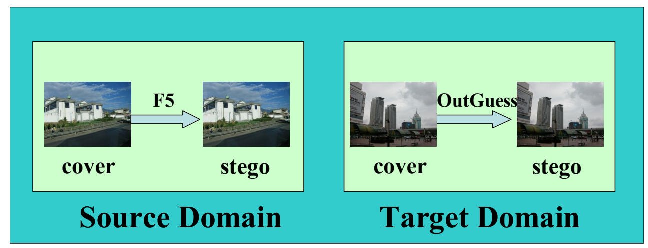
Xiaofeng Li, Xiangwei Kong, Bo Wang, Yanqing Guo, Xingang You

School of Information and Communication Engineering
Dalian University of Technology, Dalian, 116024, China

ABSTRACT

Most universal JPEG steganalysis approaches rely on the assumption that training and testing samples come from the same distribution. They fail when training set and testing set are mismatched. In this paper, we propose generalized transfer component analysis for mismatched JPEG steganalysis to derive new representations from original features for training and testing samples to correct the mismatches. We first apply domain alignment to transform source domain (training set) to an intermediate domain closer to target domain (testing set). Then a set of common transfer components are learnt across two domains by minimizing the distribution distance between them. In the space spanned by these transfer components, two domains manifest similar characteristics and preserve enough discrimination to different categories. Extensive experiments demonstrate our method performs well in mismatched JPEG steganalysis.

Index Terms— steganalysis, mismatch, domain alignment, transfer component analysis



(a) Two image sets embedded with different steganographic schemes:F5 [5], OutGuess [6]. Actually, the two sets are *B-F5* and *B-OutGuess* in Section 3.

Train	Test	JRM [1]	PF-274 [2]
Source	Source	100%	98.5%
Target	Target	99.5%	97.3%
Source	**Target**	**73.5%**	**70.5%**

(b) Detecting performance of two universal JPEG steganalysis.

Fig. 1. Effect of mismatched training and testing sets on JPEG steganalysis.

1. INTRODUCTION

Many universal JPEG steganalysis approaches have been proposed [1, 2, 3]. Most such approaches rely on the assumption that training and testing samples come from the same distribution. However in practical application, as a steganographer would not supply training set to the steganalyst, the steganalysis may be mismatched [4], i.e. training samples and testing samples come from different distributions. For example, when training stego images and testing stego images are embedded with different steganographic schemes, or training images and testing images are quantified with different quantization tables or other factors, their distributions differ much from each other and the performance of approaches above degrade greatly. As shown in Figure 1, two universal JPEG steganalysis methods (JPEG Rich Model (JRM)[1], Pevny and Fridrich's 274-dimensional feature vector (PF-274) [2])perform well when training stego images and testing stego images are embedded with the same steganographic scheme, but the performance degrades when they don't.

Only a few papers have tested steganalysis in a mismatched scenario [4, 7, 8, 9, 10, 11, 12]. In [4], Lubenko and Ker proposed to use the "Large Data "approach combined with simple classifiers for mismatched steganalysis, which required the training data set as large and diverse as possible. But it costs much labor to collect images for training set. If we only have a limited number of training samples not diverse enough, how can we train a classifier that is robust to samples coming from a different distribution from training samples?

Recently some approaches have been proposed to fix the problem of mismatched training and testing sets in other areas: computer vision tasks [13, 14], speech and language processing [15, 16], text classification [17]. In these papers, the mismatched training and testing set are named as source domain and target domain respectively. These approaches have been focusing on deriving new representations from original features for two domains. With the new representations, two domains manifest similar characteristics and preserve enough discrimination to differentiate categories. In object recognition across different domains, Gopalan et al proposed to sample geodesic flow to derive a sequence of intermediate subspaces from two domains to remove the domain shift[13]. In mismatched text classification, Pan et al proposed transfer component analysis (TCA) to find a space in which the marginal probability distribution difference between two domains decreases [17]. Compared with these areas, the mismatched conditions in our problem sometimes are worse. For example, different steganographic schemes would make their stego image statistics change a lot. Thus there may exist no such discriminative enough features common to two domains. If applying these methods directly to our problem, we can't get a satisfactory result.

To address this challenge, we propose generalized transfer component analysis (GTCA) for mismatched JPEG steganalysis. Before deriving the discriminate features common to two domains, we apply the domain alignment to transform the source domain to an intermediate domain that is much closer to the target domain. Then the representations learnt from intermediate domain and target domain would be more discriminate.

2. GENERALIZED TRANSFER COMPONENT ANALYSIS

Our proposed approach, which is applied in JPEG steganalysis system, includes four steps: (1) applying domain alignment to transform source domain to intermediate domain; (2) learning a shared feature space spanned by common transfer components between intermediate domain and target domain; (3) mapping samples to the shared feature space; (4) training a classifier on the mapped training labeled samples and using it to classify the mapped testing samples. We will explain each step in detail in the following subsection.

Suppose we have a set of n_s samples $S = \{s_1, s_2......s_{n_s}\} \in \mathbb{R}^d$ in training set (source domain) and a set of n_t samples $T = \{t_1, t_2......t_{n_s}\} \in \mathbb{R}^d$ in testing set (target domain), where d is the dimension of the feature vector. The two domains are mismatched. We have labels in the source domain, y_i is the label of s_i, where $y_i \in \{-1, 1\}, i = 1, 2......n_s$.

2.1. Domain Alignment

For ease of notation, $E(s^j, y), \sigma(s^j, y)$ are defined to represent the expectation and the standard deviation of the jth dimension feature of samples with label equal to y in source domain, where $y \in \{-1, 1\}, j = 1, 2......d$. Similarly, we define $E(t^j, y), \sigma(t^j, y)$ for target domain.

The aim of domain alignment is to learn a transformation $\varphi(\cdot)$for source domain samples to transform them to an intermediate domain M. We expect that the intra-class expectation and standard deviation of the intermediate domain equal those of target domain:

$$\begin{aligned} E(\varphi(s^j), y) = E(t^j, y) \\ \sigma(\varphi(s^j), y) = \sigma(t^j, y) \end{aligned} \tag{1}$$

where $y \in \{-1, 1\}, j = 1, 2......d$. To reach the equation(1), we define the transformation $\varphi(\cdot)$ for each dimension of each samples in source domain as:

$$\varphi(s_i^j) = (s_i^j - E(s^j, y_i))\frac{\sigma(t^j, y_i)}{\sigma(s^j, y_i)} + E(t^j, y_i) \tag{2}$$

where $j = 1, 2......d, i = 1, 2......n_s$.

The problem with this is that we don't have any label in target domain and therefore we can't estimate $E(t^j, y)$, $\sigma(t^j, y)$.

To solve the problem, we adopt the idea of an iterative fashion similar to [18]. We transform the source domain samples using a similar transformation as formula (2):

$$\psi(s_i^j) = (s_i^j - E(s^j))\frac{\sigma(t^j)}{\sigma(s^j)} + E(t^j) \tag{3}$$

to reach $E(\psi(s^j)) = E(t^j), \sigma(\psi(s^j)) = \sigma(t^j)$. We train a classifier model on the transformed source domain samples and use it to get the joint estimate $p(y|t_i)$ on the unlabeled target domain samples. Hence an approximate $E(t^j, y), \sigma(t^j, y)$ can be computed:

$$E(t^j, y) \approx \frac{1}{\sum_{i=1}^{n_t} p(y|t_i)} \sum_{i=1}^{n_t} t_i^j p(y|t_i) \tag{4}$$

$$\sigma_(t^j, y) \approx \sqrt{\frac{1}{\sum_{i=1}^{n_t} p(y|t_i)} \sum_{i=1}^{n_t} (t_i^j - E(t^j, y))^2 p(y|t_i)} \tag{5}$$

where $j = 1, 2......d$. After we get the approximate $E(t^j, y)$, $\sigma(t^j, y)$, we transform source domain to intermediate domain using the formula (2).

2.2. learning Shared Feature Space

The aim of step (2) is to find a shared space in which intermediate domain and target domain manifest similar characteristics i.e. similar feature distribution. We use a nonparametric distance measure called Maximum Mean Discrepancy (MMD) [19] to measure the distribution difference between the two domains. It is given by:

$$MMD(M, T) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(s_i^{'}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(t_j) \| \tag{6}$$

where $s_i^{'}$ represents the transformed samples s_i in intermediate domain, $\phi(\cdot)$ is kernel-induced feature map to the Reproducing Kernel Hilbert Space (RKHS), $\| \cdot \|_H$ is the squared norm computed in RKHS. This quantity is noting but the squared distance between sample means in RKHS and approaches zero when the two distributions tend to be exactly the same. Taking advantage of kernel trick, formula (6) can rewrite as:

$$MMD(M, T) = tr(\mathbf{KL}) \tag{7}$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{MM} & \mathbf{K}_{MT} \\ \mathbf{K}_{TM} & \mathbf{K}_{TT} \end{pmatrix} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)} \tag{8}$$

with $\mathbf{K}_{MM}, \mathbf{K}_{TT}, \mathbf{K}_{MT}, \mathbf{K}_{TM}$ being the kernel matrices (of element $K_{ij} = \phi(x_i)^T\phi(x_j)$)of data from intermediate domain, target domain and across domains respectively. Moreover, $L_{ij} = 1/n_s^2$ if $x_i, x_j \in M$, $L_{ij} = 1/n_t^2$ else if $x_i, x_j \in T$, and otherwise $L_{ij} = -1/n_t n_s$.

We adopt the idea, transfer component analysis, in [17] to find a non-liner mapping, a $(n_s + n_t) \times m$ kernel feature extraction matrix $\mathbf{W}$ ($m < (n_s + n_t)$), to transform both domains to a new feature space. Thus, the samples mapped in the space are achieved by:

$$\mathbf{X}_{new} = \mathbf{K}\mathbf{W} \in \mathbb{R}^{(n_s+n_t)\times m} \tag{9}$$

The corresponding kernel matrix is $\mathbf{K}_{new} = \mathbf{X}_{new}\mathbf{X}_{new}^T$,and the corresponding MMD between the two mapped domains is:

$$\begin{aligned} MMD(M^{'}, T^{'}) &= tr(\mathbf{K}_{new}\mathbf{L}) \\ &= tr(\mathbf{K}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{K}\mathbf{L}) \end{aligned} \tag{10}$$

The distance between the two domains can be decreased by minimizing the formula. Besides a constraint is added to avoid the trivial solution ($W = 0$), which can preserve (or maximize) the initial data variance in the new space. The constraint is given by:

$$\mathbf{W}^{\mathbf{T}}\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I}_m \tag{11}$$

where $\mathbf{H} = (\mathbf{I}_{(n_s+n_t)} - (1/(n_s+n_t)\mathbf{1}\mathbf{1}^{\mathbf{T}}))^2$, $\mathbf{I}$ is an identity matrix, $\mathbf{1} \in \mathbb{R}^{(n_s+n_t)\times 1}$ with all ones. It means that the covariance matrix of the mapped data in the new space should be an identity matrix.

The final kernel learning problem is then set up as:

$$\begin{aligned} \min_{W} \quad & tr(\mathbf{W}^{\mathbf{T}}\mathbf{W}) + \mu tr(\mathbf{K}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{K}\mathbf{L}) \\ s.t. \quad & \mathbf{W}^{\mathbf{T}}\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I}_m \end{aligned} \tag{12}$$

where the regularization term $tr(\mathbf{W}^T\mathbf{W})$ controls the complexity of $\mathbf{W}$ and μ is a trade-off parameter. Such an optimization problem can be reformulated as a trace maximization problem. The solution of $\mathbf{W}$ is eigenvectors corresponding to the m leading eigenvectors of $(\mathbf{I} + \mu\mathbf{K}\mathbf{L}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}\mathbf{K}$. More detailed development can be found in [17]. In our experiment, we set the dimensionality of subspaces m to 20 and μ to 0.1 empirically.

Once $\mathbf{W}$ is available, we can map the samples of intermediate and target domain to the new space by formula (9). In the new space, we can train a classifier on the mapped source labeled samples and use it to classify the target samples in the same space. The classifier we use is the support vector machine (lib-SVM [20]).

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 2 & 4 & 4 & 3 \\ 1 & 1 & 3 & 4 & 4 & 6 & 6 & 4 \\ 1 & 2 & 3 & 3 & 4 & 5 & 6 & 5 \\ 2 & 3 & 4 & 4 & 5 & 6 & 6 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \end{bmatrix} \quad \begin{bmatrix} 9 & 6 & 5 & 9 & 13 & 22 & 29 & 35 \\ 6 & 6 & 8 & 11 & 15 & 33 & 34 & 30 \\ 8 & 7 & 9 & 13 & 22 & 33 & 39 & 31 \\ 8 & 9 & 12 & 16 & 28 & 49 & 45 & 34 \\ 10 & 12 & 21 & 32 & 39 & 61 & 58 & 42 \\ 13 & 19 & 31 & 36 & 45 & 58 & 63 & 51 \\ 28 & 36 & 44 & 49 & 58 & 68 & 66 & 55 \\ 41 & 52 & 54 & 55 & 62 & 56 & 57 & 54 \end{bmatrix}$$

Fig. 2. Quantization tables of images. The left is luminance quantization table of images from Set A, and the right is that of images from Set B.

3. EXPERIMENT

To evaluate the performance of our method, we choose three common and seriously mismatched conditions in JPEG steganalysis as our experiment scenarios : (1) different quantization tables for cover images; (2) different steganographic schemes;(3) different quantization tables for cover images and different steganographic schemes.

3.1. Setup

For our experiments, we use two sets of JPEG color images(e.g., Set A and Set B)of size of 1600×1200, obtained from a camera(Canon Power Shot pro1). Each set includes 1800 JPEG images. These images contain a wide range of indoor/outdoor, daylight/night scenes, providing a real and challenging environment for a steganalysis problem. The two sets can be distinguished from each other, for they are quantified with different quantization tables as shown in Figure 2.

We classify each set into four groups and use four common JPEG steganographic schemes (F5 [5], OutGuess [6], MBS [21], Jsteg [22]) to embed the message into each group respectively. The message length is set to 10% of the maximum capacity. Thus we get eight domains: *A-F5, A-OutGuess, A-MBS, A-Jsteg, B-F5, B-OutGuess, B-MBS, B-Jsteg*. Each domain includes cover images and stego images.

In our experiment, we use the PF-274 features[2] as the original feature [1]for GTCA. We compare our method (GTCA) with the following methods:(1) **Orig-Fea**. We use the original features, PF-274, without any transformation and use lib-SVM as classifier. (2) **OEAP**. In this case, we use JRM features as image feature and use Online Ensemble Average Perceptron proposed in [4] as classifier. (3)**TCA**. In this case, we use the new representations learnt from original features, PF-274, with the method proposed in [17] as image features and lib-SVM as classifier.

For each pair of source and target domains, we conduct experiments in 5 random trials. In each trial, we randomly

[1]The reason that we choose PF-274 features instead of JRM features [1] is that the dimension of JRM features (22,510) is so high that it is time consuming to apply GTCA to them.

sample 300 labeled data per category in the source domain as training examples, and 300 unlabeled data per category in the target domain as testing examples. We report the averaged classification accuracies.

3.2. Result

To evaluate the effectiveness of our method, we conduct experiments on training and testing sets under three different mismatched conditions using the eight domains we build(*A-F: A-F5, A-O: A-OutGuess, A-B: A-MBS, A-J: A-Jsteg, B-F: B-F5, B-O: B-OutGuess, B-M: B-MBS, B-J: B-Jsteg*). As space is limited, we only give the partial results.

Mismatched Experiment 1 – Different Quantization Tables:

Assume that cover images quantified with different quantization tables are embedded with the same steganographic schemes in training and testing sets. The results are shown in Table 1.

Table 1. Results of Mismatched Experiment 1

Train→Test	A-F→B-F	A-J→B-J	A-M→B-M	A-O→B-O
Orig-Fea	0.505	0.515	0.515	0.505
OEAP [4]	0.500	0.515	0.523	0.515
TCA [17]	0.500	0.505	0.549	0.827
GTCA	0.884	0.965	0.931	0.944
Train→Test	B-F→A-F	B-J→A-J	B-M→A-M	B-O→A-O
Orig-Fea	0.505	0.545	0.535	0.515
OEAP [4]	0.525	0.505	0.505	0.545
TCA [17]	0.525	0.825	0.899	0.515
GTCA	0.787	0.975	0.951	0.865

Mismatched Experiment 2 – Different Steganographic Schemes:

Assume that cover images quantified with the same quantization tables are embedded with different steganographic schemes in training and testing sets. The results are shown in Table 2.

Table 2. Results of Mismatched Experiment 2

Train→Test	B-F→B-M	B-F→B-O	B-J→B-M	B-J→B-O
Orig-Fea	0.695	0.705	0.533	0.515
OEAP [4]	0.833	0.755	0.553	0.535
TCA [17]	0.865	0.785	0.655	0.602
GTCA	0.885	0.875	0.835	0.845
Train→Test	B-J→B-F	B-M→B-F	B-M→B-O	B-O→B-F
Orig-Fea	0.515	0.635	0.870	0.541
OEAP [4]	0.572	0.653	0.905	0.835
TCA [17]	0.755	0.845	0.855	0.775
GTCA	0.745	0.885	0.983	0.877

Mismatched Experiment 3 – Different Steganographic Schemes and Quantization Tables:

Assume that cover images quantified with different quantization tables are embedded with different steganographic schemes in training and testing sets. The results are shown in Table 3.

Table 3. Results of Mismatched Experiment 3

Train→Test	A-F→B-M	A-F→B-O	A-J→B-M	A-J→B-O
Orig-Fea	0.495	0.510	0.500	0.500
OEAP [4]	0.535	0.545	0.523	0.515
TCA [17]	0.505	0.500	0.500	0.502
GTCA	0.805	0.837	0.785	0.753
Train→Test	A-J→B-F	A-M→B-F	A-M→B-O	A-O→B-F
Orig-Fea	0.500	0.510	0.535	0.512
OEAP [4]	0.523	0.515	0.550	0.515
TCA [17]	0.504	0.530	0.559	0.575
GTCA	0.733	0.735	0.922	0.807

From the experimental results above, we can find out that our method performs the best in three mismatched JPEG steganalysis. In each case, GTCA performs better than method proposed in [4], which confirms the superiority of our method when the training set is small and not diverse enough. In most cases, GTCA outperforms TCA[17], which verifies that applying domain alignment can contribute to TCA deriving more discriminate features between two domains. Only in Mismatched Experiment 2, the performance of GTCA is worse than TCA for B-J to B-F. We think the reason is that formulae (4),(5) in Domain Alignment are approximate. In most cases, the two approximate formulae are reasonable. Only when the difference between two domains were huge and the precision of the joint estimate were poor, the error of the approximate formulae would increase. In Mismatched Experiment 3, the results are not as good as those of the former ones. The reason is that the mismatched conditions for experiment 3 are much worse.

4. CONCLUSION

In this paper, we propose generalized transfer component analysis for mismatched JPEG steganalysis. We first apply the domain alignment to transform source domain to an intermediate domain that is much closer to target domain. Then we derive the shared feature space spanned by common transfer components between intermediate domain and target domain in RKHS. Compared with previous approaches, the representations we derive are more discriminate. We demonstrate the effectiveness of our approach under three different mismatched conditions for training and testing sets. The results show that our method can decrease the mismatches in JPEG steganalysis very well.

5. ACKONWLEDGEMENT

This work has been supported in part by grant from the Chinese National Nature Science Foundation under contract NO. 70890083 and NO. 60971095.

6. REFERENCES

[1] Jan Kodovsky and Jessica Fridrich, "Steganalysis of jpeg images using rich models," in *Proc. of SPIE Media Watermarking, Security, and Forensics*, 2012, vol. 8303, pp. 1–13.

[2] Tomas Pevny and Jessica Fridrich, "Merging markov and dct features for multi-class jpeg steganalysis," in *Proc. of SPIE Security,Steganography,and Watermarking of Multimedia Contents*, 2007, vol. 6505, pp. 1–13.

[3] Chunhua Chen and Y.Q. Shi, "Jpeg image steganalysis utilizing both intrablock and interblock correlations," in *Proc. of International Symposium on Circuits and Systems*, 2008, pp. 18–21.

[4] Ivans Lubenko and Andrew D. Ker, "Steganalysis with mismatched covers: Do simple classifiers help?," in *Proc. of ACM Multimedia and Security Workshop*, 2012, pp. 11–18.

[5] Andreas Westfeld, "F5-a steganographic algorithm: High capacity despite better steganalysis," in *Proc. of International Workshop on Information Hiding*, 2001, pp. 289–302.

[6] K. Solanki, N. Jacobsen, U. Madhow, B.S. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding based on erasure and error correction," *IEEE Transactions on Image Processing*, vol. 13, pp. 1627–1639, December 2004.

[7] M. Barni, G. Cancelli, and A. Esposito, "Forensics aided steganalysis of heterogeneous images," in *Proc. of IEEE Conference on Acoustics Speech and Signal Processing*, 2010, pp. 1690–1693.

[8] G. Cancelli, G. Doërr, I. Cox, and M. Barni, "A comparative study of ± 1 steganalyzers," in *Proc. of IEEE Multimedia Signal Processing Workshop*, 2008, pp. 791–794.

[9] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, "Steganalysis of content-adaptive steganography in spatial domain," in *Proc. of Information Hiding Workshop*, 2011, pp. 102–117.

[10] G. Gül and F. Kurugöllü, "Steganalysis of content-adaptive steganography in spatial domain," in *Proc. of Information Hiding Workshop*, 2011, pp. 102–117.

[11] A. D. Ker and I. Lubenko, "Feature reduction and payload location with wam steganalysis," in *Proc. of SPIE Media Forensics and Security*, 2009, vol. 7254, p-p. 0A01–0A13.

[12] P. Bas T. Pevný and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proc. of ACM Multimedia and Security Workshop*, 2009, pp. 75–84.

[13] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. of International Conference on Computer Vision*, 2011, pp. 999–1006.

[14] B.Q Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[15] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 120–128.

[16] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 440–447.

[17] S. Pan, I. Tsang, J. Kwok, and Q.Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, pp. 199–210, Febuary 2011.

[18] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Proc. of International Conference on Data Mining Workshops*, 2007, pp. 77–82.

[19] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, pp. e49–e57, July 2006.

[20] "Libsvm," `http://www.csie.ntu.edu.tw/cjlin/libsvm/`.

[21] Phil Sallee, "Model-based steganography," in *Proc. of International Workshop on Digital Watermarking*, 2004, pp. 154–167.

[22] "Jpeg-jsteg-v4," `http://www.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz`.