Analyzing Quantitative Detectors for Content-Adaptive Steganography

Edgar Kaziakhmedov, Eli Dworetzky, and Jessica Fridrich, Department of ECE, SUNY Binghamton, NY, USA, {ekaziak1, edworet1, fridrich}@binghamton.edu

Abstract

In this article, we study the properties of quantitative steganography detectors (estimators of the payload size) for content-adaptive steganography. In contrast to non-adaptive embedding, the estimator's bias as well as variance strongly depend on the true payload size. Initially, and depending on the image content, the estimator may not react to embedding. With increased payload size, it starts responding as the embedding changes begin to "spill" into regions where their detection is more reliable. We quantify this behavior with the concepts of reactive and estimable payloads. To better understand how the payload estimate and its bias depend on image content, we study a maximum likelihood estimator derived for the MiPOD model of the cover image. This model correctly predicts trends observed in outputs of a state-of-the-art deep learning payload regressor. Moreover, we use the model to demonstrate that the cover bias can be caused by a small number of "outlier" pixels in the cover image. This is also confirmed for the deep learning regressor on a dataset of artificial images via attribution maps.

Introduction

The main objective of steganalysis is to detect the use of steganography. This requires algorithms that can identify statistical anomalies introduced by steganographic algorithms. Quantitative steganalysis is a term used for detectors designed to estimate the length of the embedded message. The first detectors of this type were designed for least significant bit replacement (LSBR). The RS steganalysis [14] laid the ground for a direction called structural steganalysis further developed by Dumitrescu et al. [12, 11] and Ker [19, 20]. A different class of quantitative attacks on LSBR is the Weighted Stego-image attack [13, 23, 22, 3], which was later (re)derived as a likelihood ratio test [40] whose expectation is the embedding change rate. Structural steganalysis of LSBR in JPEG images employing a zero message hypothesis [39], embedding invariants, and the maximum likelihood estimation was introduced in [24, 31]. Approaching quantitative steganalysis with machine learning by training detectors as payload regressors (e. g., support vector regressors) on steganalysis "features" [27] allowed constructing quantitative detectors in a fully automatized fashion for all embedding schemes in both the spatial and JPEG domain. By making the features "aware of parity", further advancements in quantitative steganalysis of LSBR became possible [16]. With high-dimensional "rich" models [15] data-driven regressors became even more accurate [25, 32, 9]. Today, state-of-the-art quantitative detectors are built with deep convolutional neural networks (CNNs) trained as payload regressors [29, 8].

The first rigorous analysis of the error of quantitative steganalyzers appeared in [4]. The authors presented evidence that the estimation error consists of two components – the highly non-Gaussian between-image error or cover bias and the Gaussian within-image error due to the randomness in embedding (pixel visitation). The right tail of the distribution of the between-image error was experimentally shown to be well modeled with the Student's t-distribution, which has power tails that affect the false-alarm rate should the quantitative steganalyzer be used as a binary detector of steganography. The statistical distribution of the between-image error was derived by Ker for LSB replacement and least squares quantitative steganalysis [21]. To the best knowledge of the authors, the error of data-driven quantitative steganalyzers for content-adaptive steganography has not been studied before.

After defining our notation and acronyms, in Section "Detector response curves" we introduce the key concept studied in this paper – the detector response curve defined as the detector's expected soft output as a function of the embedded message length. We also introduce two novel critical payloads for evaluating quantitative steganalyzers, which are the reactive and estimable payloads. In Section "Quantitative steganalyzers" we describe the quantitative steganalyzer studied in this paper, which is a novel end-to-end trained deep learning regressor; its performance is briefly evaluated against state of the art. To obtain insight into how the response curves of the data-driven regressor depend on the embedded message length and image content, in Section "MLE of payload size" we study the response curves of a maximum likelihood estimator of payload size derived for the MiPOD model of the cover image. In Section "Explaining trends ..." a rather tight match is observed between the outputs of this MLE and a data-driven regressor, which helps us understand the observed trends and behavior of the data-driven regressor. Moreover, in Section "Analyzing cover bias" we use our cover model and the MLE to show that the estimator's bias on covers is often caused by a small number of "outlier pixels" in the cover image. This inspired us to verify the existence of such "influential pixels" for the data-driven detector via attribution maps. To this end, we work with a deep learning detector trained on an artificial version of our dataset containing natural-looking images that follow our cover model. The paper is concluded in the last Section "Conclusions."

Notation

In this section, we introduce basic notational conventions. Vectors and matrices will be typed in boldface. Their elements will be denoted with the corresponding non-boldface letter. Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The ternary entropy function is $H_3(x) = -2x \log x - (1-2x)\log(1-2x)$. In this paper, we work solely with 8-bit grayscale images with pixels from the set $\{0, \dots, 255\} \triangleq \mathcal{I}$.

The following abbreviations are used throughout this paper: DLR for deep learning regressor, SVR for support vector regressor, RC for the response curve of a quantitative detector, CNN for convolutional neural network, LSB for least significant bit, MLE for maximum likelihood estimation, MSE and MAE for the mean square (absolute) error, and TTA for test time augmentation.

We strictly use the symbols $\mathbf{c} \in \mathcal{I}^{M \times N}$ and $\mathbf{s} \in \mathcal{I}^{M \times N}$ for cover and stego images with $n = M \times N$ pixels, respectively, while $\mathbf{x}$ and $\mathbf{y}$, both from $\mathbb{R}^{M \times N}$, are reserved for model representation of images (their noise residuals). Data-driven detectors will be trained and tested on examples of $\mathbf{c}$ and $\mathbf{s}$ while the MLE will work with $\mathbf{x}$ and $\mathbf{y}$.

Detector response curves

This section introduces the key concept studied in this paper, which is the detector response curve (RC). It is defined as the output of a quantitative detector as a function of payload size for a given cover image and embedding technique. We will be interested in the trends exhibited by RCs based on the content of the cover image.

Given a cover image with n pixels $\mathbf{c} = \{c_i\}_{i=1}^n$, $c_i \in \mathcal{I}$, its stego version after embedding relative payload α bpp, $\mathbf{s}(\alpha)$, is a random variable with the same domain. In particular, for a ternary embedding scheme with symmetric costs $\boldsymbol{\rho} = \{\rho_i\}$ that embeds at the rate-distortion bound (ignoring the values on the boundary of the dynamic range $\mathcal{I}$), $s_i(\alpha) = c_i + \eta_i$, with the stego signal PMF

$$\begin{aligned} \mathbb{P}(\eta_i = -1) = \mathbb{P}(\eta_i = 1) = \beta_i \\ \mathbb{P}(\eta_i = 0) = 1 - 2\beta_i, \end{aligned} \tag{1}$$

where

$$\beta_i(\lambda) = \frac{e^{-\lambda \rho_i}}{1 + 2e^{-\lambda \rho_i}}, \tag{2}$$

with the Lagrange multiplier $\lambda > 0$ determined from the payload constraint

$$\alpha n = \sum_{i=1}^{n} H_3(\beta_i(\lambda)). \tag{3}$$

When presenting the detector with stego images obtained with different stego keys k, the detector's output $d(\mathbf{s}(\alpha; k))$ is a random variable due to the within-image error, which is well modeled with a Gaussian distribution [4].

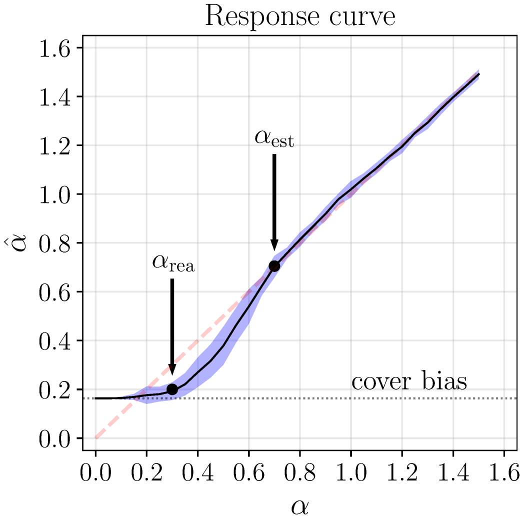

Figure 1. *An example of a typical response curve $\varrho(\alpha; \mathbf{c})$ (black solid line). The shading shows the standard deviation of the response across stego images of the same length. Initially, the detector does not respond to embedding because content-adaptive steganography makes changes in areas of the image where they are hard to detect. With increased payload, the detector starts responding, and it eventually outputs a good estimate of the payload size. We quantify this with the concept of reactive and estimable payloads α_{rea} and α_{est}. See the text for a formal definition.*

The expectation

$$\varrho(\alpha; \mathbf{c}) = \mathbb{E}_k[d(\mathbf{s}(\alpha; k))] \tag{4}$$

is called the detector's *response curve* for cover image $\mathbf{c}$. The estimator bias is defined as

$$b(\alpha; \mathbf{c}) = \varrho(\alpha; \mathbf{c}) - \alpha, 0 \le \alpha \le \alpha_{\max}, \tag{5}$$

where $\alpha_{\max} \le \log_2 3$ is the maximal relative payload embeddable in the image. Note that $\alpha_{\max}$ may be smaller than $\log_2 3$ because many steganographic schemes for example avoid making changes to saturated pixels. To declutter the notation, sometimes we will omit the parameter in the RC and write simply $\varrho(\alpha)$ as it should be clear from the context that the RC is for a specific cover.

The reader is advised to follow Figure 1, which shows a "typical" RC of a quantitative steganalyzer. The value $b(0; \mathbf{c}) = \varrho(0; \mathbf{c})$ is called the between-image error [4]. In this paper, we call it the *cover bias* instead for the following reason. For non-adaptive embedding schemes, the estimator bias $b(\alpha; \mathbf{c}) = \varrho(\alpha; \mathbf{c}) - \alpha$ is approximately independent of α, $b(\alpha; \mathbf{c}) = b(0; \mathbf{c})$. As will be seen below, for content-adaptive embedding the bias is far from being constant. This is because for small payloads the pixels that are likely to be modified by embedding are constrained to such areas of the cover image where detection is the most difficult. This is why initially the RC appears "flat." When the payload becomes sufficiently large, the embedding changes start "spilling" into detectable regions of the image and the detector output starts to increase. With

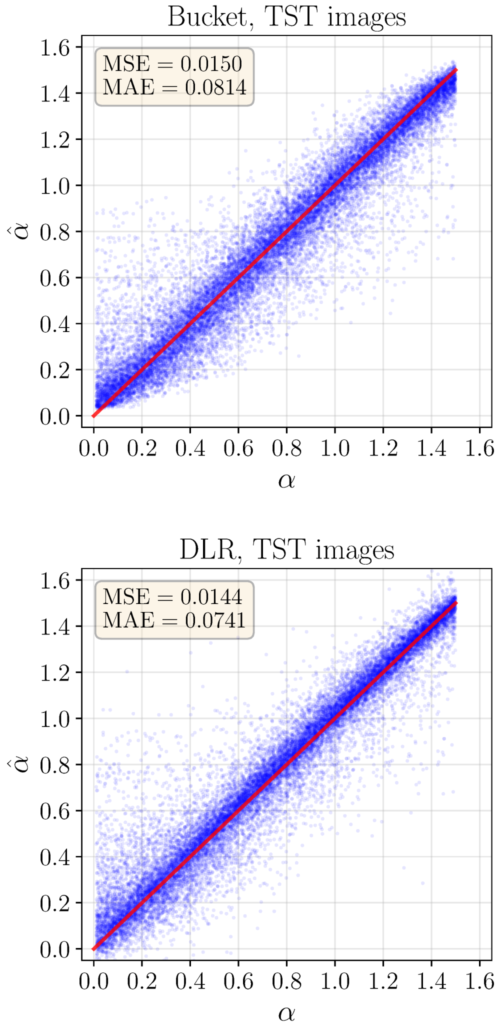

Figure 2. *Scatter plots showing the estimated payload versus the true payload for the Bucket Estimator (top) and DLR (bottom) for S-UNIWARD on the TST set.*

enough embedding changes in detectable regions, the output of the detector grows with a slope approximately equal to 1. This behavior motivated us to define the following critical payloads to numerically characterize RCs.

Reactive payload α_{rea} is the smallest payload α for which $\varrho(\alpha)$ is "different" than the cover bias $\varrho(0)$:

$$\alpha_{\text{rea}}(\tau_{\text{rea}}) = \min_{\alpha}\{\alpha|\varrho(\alpha;\mathbf{c}) - \varrho(0;\mathbf{c}) > \tau_{\text{rea}}\} \qquad (6)$$

where $\tau_{\text{rea}} > 0$ is a suitably chosen fixed threshold.

Estimable payload is defined as the smallest payload for which the RC starts showing a unit slope: $\varrho(\alpha+\Delta\alpha) \approx \varrho(\alpha)+\Delta\alpha$. The definition of estimable payload mimics the definition of reactive payload but with respect to the line with slope 1 instead of a line parallel to the x axis:

$$\alpha_{\text{est}}(\tau_{\text{est}}) = \max_{\alpha}\Big\{\alpha\Big|\,\big||\varrho(\alpha;\mathbf{c}) - \alpha| - |\varrho(\alpha_{\max};\mathbf{c}) - \alpha_{\max}|\big| > \tau_{\text{est}}\Big\}, \qquad (7)$$

where $\alpha_{\max}$ is the maximal embeddable payload in cover $\mathbf{c}$.

Note that both critical payloads are essentially defined by the payloads for which the RC starts "peeling away" from a line. It is the horizontal line with intercept equal to the cover bias for the reactive payload and the diagonal line with slope 1 for the estimable payload. Also note that we allow the estimable payload to not match the true payload (we allow a bias) as long as the detector response starts increasing linearly with payload with slope 1.

In general, the smaller the reactive and estimable payloads are the better the quantitative detector is. Thus, both concepts offer an additional way of benchmarking quantitative detectors.

Quantitative steganalyzers

In this section, we describe the quantitative detector that will be used for our study and briefly benchmark it against the previously proposed Bucket Estimator.

The main bulk of experiments were conducted on a dataset of 20,000 images, which is the union of the BOSSBase 1.01 [1] and BOWS2 [2], each with 10,000 grayscale images resized to 256×256 pixels with `imresize` in Matlab using default parameters. We refer to this merged dataset as BB for brevity. This dataset is a popular choice for designing detectors with deep learning because small images are more suitable for training deep architectures [34, 5, 35, 36, 33, 38].

The BB was split into three disjoint sets for training deep learning quantitative steganalyzers that will be analyzed in this paper. The training set (TRN) contains all 10,000 BOWS2 images along with 7,000 randomly selected images from BOSSbase. The remaining images from BOSSbase were randomly partitioned to create the validation set (VAL) and the testing set (TST) with 1,000 and 2,000 images, respectively.

Deep learning detector (DLR)

In this paper, we describe a novel computationally efficient quantitative detector built as an SRNet [5] trained in an end-to-end fashion to regress the relative payload expressed in bits per pixel (bpp). We abbreviate this detector as DLR, Deep Learning Regressor. The decision to work with the DLR rather than the previously proposed Bucket Estimator proposed by Chen et al. [8] was made to avoid dealing with the excessive complexity and GPU memory requirements of the Bucket Estimator when regressing payloads in the full payload range $[0, 1.5]$ bpp. Since, to the best knowledge of the authors, the DLR has not been previously studied, in Section "Experimental benchmark" we provide a brief comparison with the Bucket Estimator on S-UNIWARD to show that the DLR is indeed the better detector.

To prepare the stego images for training the DLR, each image from BB TRN and VAL set was embedded 10 times with payloads uniformly and randomly pre-sampled from the range [0.01, $\min(\alpha_{\max}, 1.5)$] bpp, where $\alpha_{\max}$ is the maximal payload for each image. During training, the network is exposed to all 10 payloads every epoch. Images from the testing set were embedded 50 times from the same range.

Bucket estimator

The Bucket Estimator was built as described in [8] with a few differences, which we point out. First, the TRN set is randomly split into two disjoint subsets, which we call TRN1 and TRN2, each with 14,000 and 3,000 images. Then, on TRN1 we trained 15 binary detectors (SRNets) to distinguish the class of cover images and stego images embedded with a fixed relative payload $\alpha \in \mathcal{A} = \{0.1, 0.2, \ldots, 1.5\}$ bpp. Note that we use the VAL set for evaluating binary detectors. The SRNet detectors are used as "feature extractors" that map an input image to a 512-dimensional vector – the output of the last convolutional layer before the IP layer. The actual bucket estimator was trained as a multi-layered perceptron (MLP) on the union of these features on TRN2. For MLP training, each image from TRN2 is embedded 10 times in the same way as for DLR. Since the concatenated features have a dimensionality of $d = 512 \times 15 = 7{,}680$, to keep the number of parameters within the memory of our GPUs (and in contrast to [8]), we used a MLP with three hidden layers with 7,680, 960, and 120 nodes. The first two hidden layers are also followed by batch normalization and the ReLU activation function. The weight decay parameter was set to 1 to prevent the MLP from overfitting.

All deep learning detectors were initialized with JIN-SRNet [7], which is the SRNet [5] pre-trained on ImageNet [10] and its stego version embedded with J-UNIWARD [17] with payloads uniformly randomly selected from the interval [0.4, 0.6] bpnzac. The networks were then refined for a given steganalysis task via transfer learning as described in [7] with cross-entropy loss for binary steganalysis and L_2 loss for quantitative steganalysis (payload regression or DLR).

Experimental benchmark

We now benchmark the performance of the DLR against the Bucket Estimator on S-UNIWARD. For this purpose, we use the standard mean square error (MSE) and mean absolute error (MAE) across all images in the TST set as well as the newly introduced critical payloads. Figure 2 shows the scatter plots of the estimated payload vs. true payload for the Bucket Estimator (top) and for DLR (bottom). Table 1 shows the MSE, MAE, and average reactive and estimable payloads across the TST set. The DLR outperforms the Bucket Estimator with all four performance measures. We experimented with a range of different choices for the thresholds τ_{rea} and τ_{est} and eventually settled on $\tau_{\text{rea}} = 0.05$ and $\tau_{\text{est}} = 0.03$, which seemed to visually capture the concept of these two critical payloads the best.

Regressor	MSE	MAE	$\overline{\alpha}_{\text{rea}}(0.05)$	$\overline{\alpha}_{\text{est}}(0.03)$
Bucket	0.0150	0.0814	0.216	1.115
DLR	0.0144	0.0741	0.191	0.754

Table 1. Comparison between the Bucket Estimator and DLR through the MSE and MAE, and the reactive and estimable payloads averaged over the TST set for S-UNIWARD. Note that a smaller reactive (estimable) payload implies better performance.

Additionally, in Figure 3 we plot the histogram of reactive and estimable payloads for the DLR and for the Bucket Estimator. In agreement with the evidence provided by the conventional performance measures, the MSE and MAE (Table 1), the histograms confirm that DLR's performance improves upon the Bucket Estimator. We notice that both estimators are rather close in terms of starting to respond to embedding, $\hat{\alpha}_{\text{rea}}$, but the DLR exhibits a linear RC with slope 1 much sooner than the Bucket Estimator.

MLE of payload size

To better understand the trends observed in RCs across images, we use the MiPOD image model [28] and study the RCs of a MLE of the payload size within this model. In MiPOD, cover pixels are modeled as independent Gaussian variables, $\mathbf{c} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. The mean $\boldsymbol{\mu}$ is set to zero in MiPOD because it has no effect on the stego algorithm as the embedding "costs" (Fisher information) only depend on the variance. The variance estimator was designed to minimize MiPOD's empirical detectability of steganography (see Section V in [28]). In this section, the mean has no effect on the MLE because it is known, hence we also make the assumption that $\boldsymbol{\mu} = \mathbf{0}$. Finally, we wish to emphasize that the MLE described here is for a general ternary embedding algorithm and not necessarily for MiPOD itself. MiPOD's variance estimator is merely used for estimating the cover image model.

Given a steganographic scheme with symmetric costs ρ_i computed from the cover image $\mathbf{c}$ and ignoring the effect of quantization, the stego image pixels $\mathbf{y} = \{y_i\}$ follow a Gaussian mixture

$$y_i \sim \beta_i \mathcal{N}(-1, \sigma_i^2) + \beta_i \mathcal{N}(1, \sigma_i^2) + (1 - 2\beta_i)\mathcal{N}(0, \sigma_i^2), \quad (8)$$

where the pixel change rates β_i (2) satisfy the payload constraint (3).

Note that the stego image $\mathbf{y}$ is the result of a doubly stochastic process. First, the cover $\mathbf{x}$ is sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and then modified to $\mathbf{y} = \mathbf{x} + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is the stego signal (1). With this model, one can estimate the message length using the maximum likelihood estimator

$$\hat{\alpha}^{(\text{MLE})}(\mathbf{x}, \mathbf{y}, \alpha) = \arg\max_{\alpha} \sum_{i=1}^{n} \log f(\mathbf{y}|\alpha), \quad (9)$$

where $f(\mathbf{y}|\alpha)$ is the likelihood (8) of observing the stego image residuals (8) when payload of length α is embedded in $\mathbf{x}$. In practice, for better numerical stability we first estimate the Lagrange multiplier λ using MLE, which is

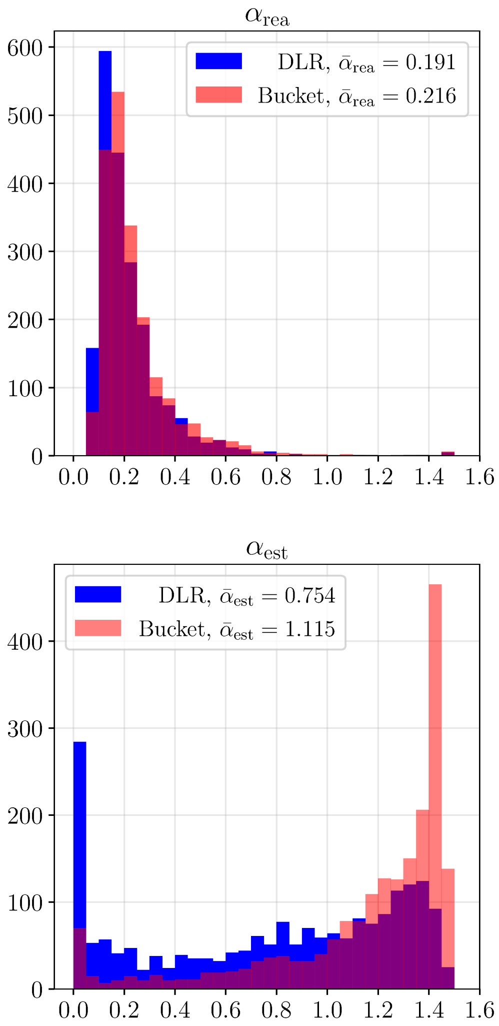

Figure 3. *Histogram of reactive and estimable payloads for S-UNIWARD across the TST set. The DLR (blue) consistently starts reacting and estimating for smaller payloads than the Bucket Estimator (coral).*

then substituted into (2) and (3) to obtain the estimate of the relative payload α for a given stego realization $\mathbf{y}$.

To study the RC for a given cover image $\mathbf{c}$ from BB, we first compute its embedding costs ρ_i and then estimate the variances σ_i^2 using MiPOD's variance estimator. Once the costs and the model parameters are estimated, we compute one realization of the cover $x_i \sim \mathcal{N}(0, \sigma_i^2)$, $i = 1, \dots, n$. This is the equivalent of a "cover image" within the MiPOD model. Then, for each payload α we add the stego signal $\eta_i(\alpha) \in \{-1, 0, 1\}$, $y_i(\alpha) = x_i + \eta_i(\alpha)$, to simulate embedding. This is repeated N_p times (p as in payload) for the same x_i and α but a different stego signal $\boldsymbol{\eta}(\alpha)$ in order to estimate the expectation of the MLE outputs over embeddings

$$\varrho^{(\text{MLE})}(\alpha; \mathbf{x}) = \mathbb{E}_{\mathbf{y}}[\hat{\alpha}^{(\text{MLE})}(\mathbf{x}, \mathbf{y}, \alpha)], \tag{10}$$

which is the MLE's RC for a specific cover realization $\mathbf{x}$.

Explaining trends in DLR's RCs

Equipped with an image model and a MLE of the payload, our goal is to explain the trends observed in RCs of the DLR, including the cover bias.

Figure 5 shows the RCs for one BB image shown on the right. The middle figure contains eight RCs $\varrho(\alpha; \mathbf{x})$ for eight cover realizations $\mathbf{x}$ derived from a MiPOD model of the same cover image $\mathbf{c}$. Note that the cover bias varies significantly across cover realizations. To see if the DLR exhibits a similar behavior, we would need multiple "acquisitions" of the cover image. However, for most datasets there is only one realization of the cover available. Instead, we feed the DLR with eight different versions of the cover image (and its stego versions) obtained by rotating it by integer multiples of 90 degrees and mirroring (the D4 group of augmentations). While this is technically different than supplying actual acquisitions, it is at least similar in spirit. As Figure 5 left shows, DLR's RCs also exhibit a large statistical spread of the cover bias as the MLE for the image model. In the next section, we argue that this statistical spread is due to a small number of outlier pixels in the cover.

Given that the cover bias is quite sensitive to a few outlier pixels, to obtain a more meaningful comparison between the RCs of the DLR and the MLE, we average the RCs across all eight cover realizations (D4 transformations).[1] In Figure 5, we show examples of such averaged RCs for 12 randomly selected images from BB. Note that the MLE RCs seem to match the trends exhibited by the DLR in terms of the initial "flat" region and the region where the regressor begins outputting a reasonably accurate estimate of the payload size.

Analyzing cover bias

Intrigued by the observed sensitivity of the cover bias w.r.t. cover realizations (Figure 4), in this section we analyze this phenomenon for the MLE within our model and then contrast our findings on experiments with the DLR. In summary, the cover bias is due to a small number of outlier pixels in the cover image that the regressors mistaken for a sign of embedding.

Influential pixels (MLE)

Given a cover image $\mathbf{x}$, ignoring the constant term and the term that only depends on $\mathbf{x}$ but not on payload (λ), the log likelihood for the mixture (8) is

$$f(\mathbf{y}|\lambda) = \sum_{i=1}^{n} \ell_i(x_i, \lambda), \tag{11}$$

where

$$\ell_i(x_i, \lambda) = \ln(1 - 2\beta_i + \beta_i a_i^+ + \beta_i a_i^-) \tag{12}$$

[1] Averaging network outputs over the D4 group at test time, which is recognized as test time augmentation (TTA), is commonly done in machine learning to improve performance (see Sec. 3.7.2 in [37]).

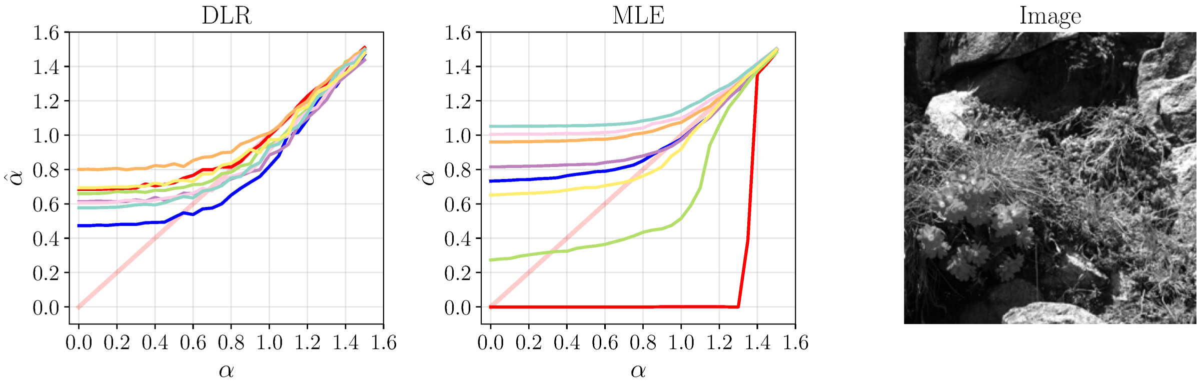

Figure 4. *Left: response curves of the DLR across eight D4 transformations for the same image. Middle: Response curves $\varrho(\alpha;\mathbf{x})$ for eight cover realizations $\mathbf{x}$ derived from a MiPOD model of one BB image shown on the right. Note that both detectors show a wide spread of the cover bias.*

with $\beta_i(\lambda)$ (2) and

$$a_i^+ = \exp\left(-\frac{1+2x_i}{2\sigma_i^2}\right) \tag{13}$$

$$a_i^- = \exp\left(-\frac{1-2x_i}{2\sigma_i^2}\right). \tag{14}$$

As λ goes from $\lambda = 0$ (maximal payload) to $\lambda \to +\infty$ (zero payload), ℓ_i's contribute to $f(\mathbf{x}|\lambda)$ to a different degree, depending on a_i^+, a_i^-, and on β_i, which in turn depend on the exact value of the cover x_i, on σ_i, and on the cost ρ_i. For large σ_i, both a_i^+ and a_i^- are likely to be small, hence the contribution of $\ell(x_i)$ to the log likelihood is not so insignificant. For small[2] σ_i, however, one or both a_i^+ and a_i^- can be very large. For a $\kappa\sigma_i$ outlier $x_i = \kappa\sigma_i$, $\kappa > 0$, $a_i^- = \exp\left(\frac{2\kappa\sigma_i - 1}{2\sigma_i^2}\right)$. For example, a 4σ outlier $x_i = 4\sigma_i$ for $\sigma_i = 0.2$ leads to $a_i^- = e^{7.5} \approx 1808$. Such pixels have the potential to strongly affect the MLE output. As seen in (12), for a pixel to be "influential" we need to consider the magnitude of the products $\beta_i a_i^+$ and $\beta_i a_i^-$ or, in other words, the effect of embedding costs. Indeed, a pixel's impact may be attenuated by small β_i (large cost). Thus, we identify candidates for influential pixels based on their "outlierness" computed from the logarithms of the products $\beta_i a_i^+$ and $\beta_i a_i^-$:

$$\begin{aligned} o_i(x_i) &= \max\left\{\ln \beta_i a_i^+, \ln \beta_i a_i^-\right\} \\ &= \ln \beta_i + \max\left\{-\frac{1+2x_i}{2\sigma_i^2}, -\frac{1-2x_i}{2\sigma_i^2}\right\}. \end{aligned} \tag{15}$$

Notice that a small change rate β_i decreases $o_i(x_i)$. In all experiments below, we computed β_i for payload 0.5 bpp.

To test how well the outlierness (15) captures a pixel's effect on the cover bias, the following experiment was conducted with the MLE. We selected an image that exhibited a wide range of the cover bias across cover acquisitions $\mathbf{x}$ (see the response curves in Figure 6 left). Then, we selected one acquisition with a large cover bias and another one with a very small bias.

For the image with a negligible bias, we carried out the following insertion experiment. After sorting the residuals by $o_i(4\sigma_i)$, we replaced k values x_i with the largest $o_i(4\sigma_i)$ with $4\sigma_i$ for $k = 1, 2, \ldots$ and observed the effect on the response curves. As Figure 6 middle shows, inserting only a small number of outlier pixels rapidly increases the cover bias. In fact, replacing even a single pixel can introduce a noticeable bias. In contrast, inserting 4σ outliers at randomly selected pixels has a much less pronounced effect (Figure 6 right). This experiment also justifies using o_i (15) as a measure for identifying influential pixels.

For the image with a large bias, we carried out the following complementary deletion experiment to see if removing pixels with large o_i can decrease the cover bias. The values x_i with the k largest o_i were replaced with $x_i = 0$ while gradually increasing $k = 1, 2, \ldots$. Figure 7 top confirms that the cover bias rapidly decreases when deleting top k influential pixels (pixels with large o_i). In contrast, deleting randomly selected pixels (the figure on the bottom) has no effect on the cover bias.

To confirm that the above case study is a typical case for images from BB, we executed the deletion and insertion experiments for all BB images from the testing set. For the insertion test, we selected 1,174 images with cover bias less than 0.01 and inserted outlier pixels to increase the bias to at least 0.2. Our goal was achieved for 81% of images with the average amount of inserted outlier pixels equal to 20. For the deletion test, we selected 177 images with cover bias larger than 0.2. Subsequently, we applied the deletion procedure to decrease the cover bias below 0.01. On average, we needed to delete 11 pixels to achieve our goal across all 177 images.

In summary, we demonstrated that within our model setup only a small number of influential pixels with large o_i have a major effect on the MLE's cover bias. In the next section, we study whether this same phenomenon is also observed with data-driven detectors (DLR).

[2] MiPOD's estimator floors the estimated standard deviations to $\sigma_i \geq 0.1$.

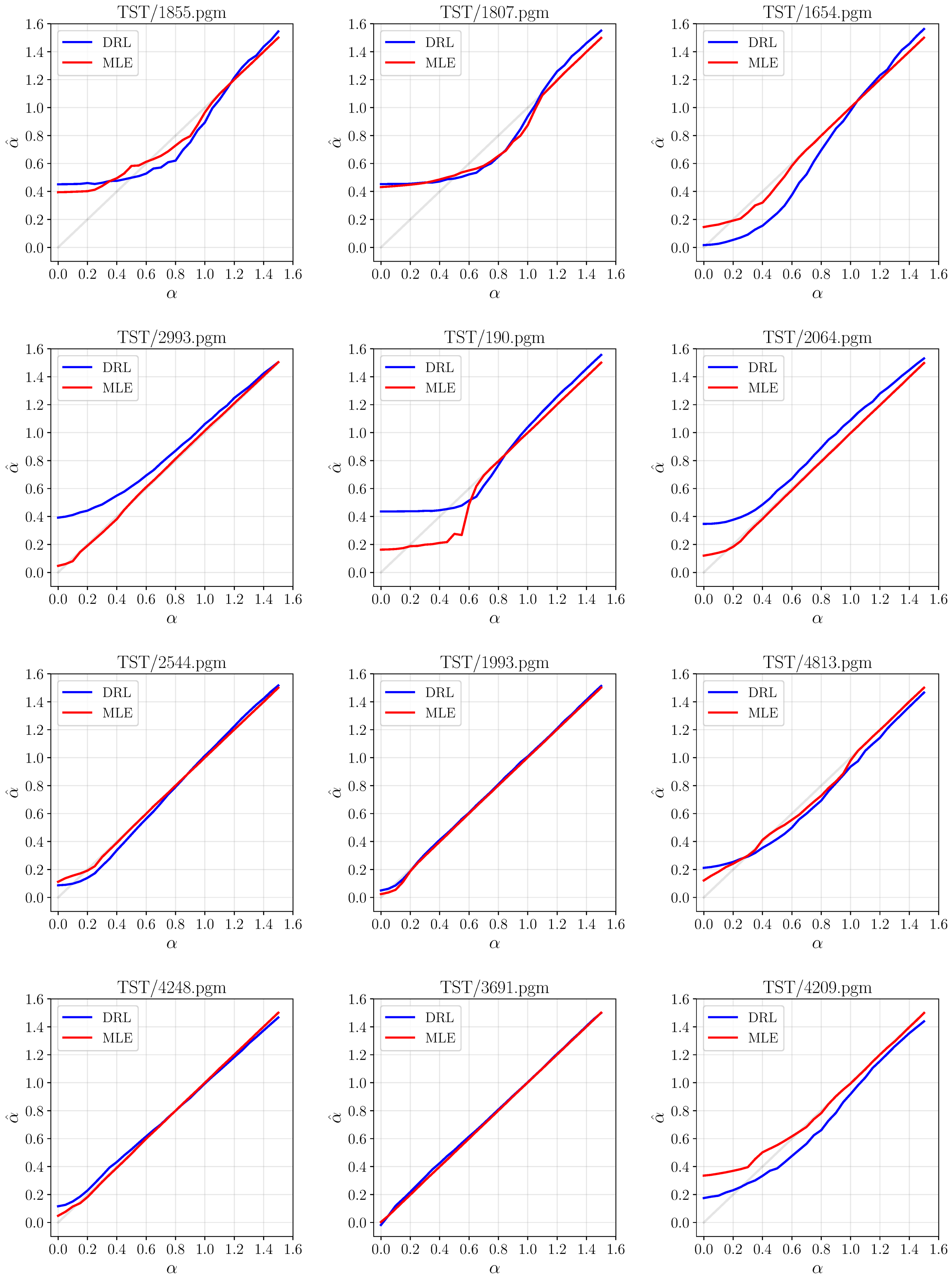

Figure 5. *Examples of RCs of the DLR for 12 randomly selected images from the TST set (blue) and the RCs of the MLE for their residual models (red). The RCs were averaged over eight D4 rotations and flips for the DLR and over eight realizations of covers for the MLE.*

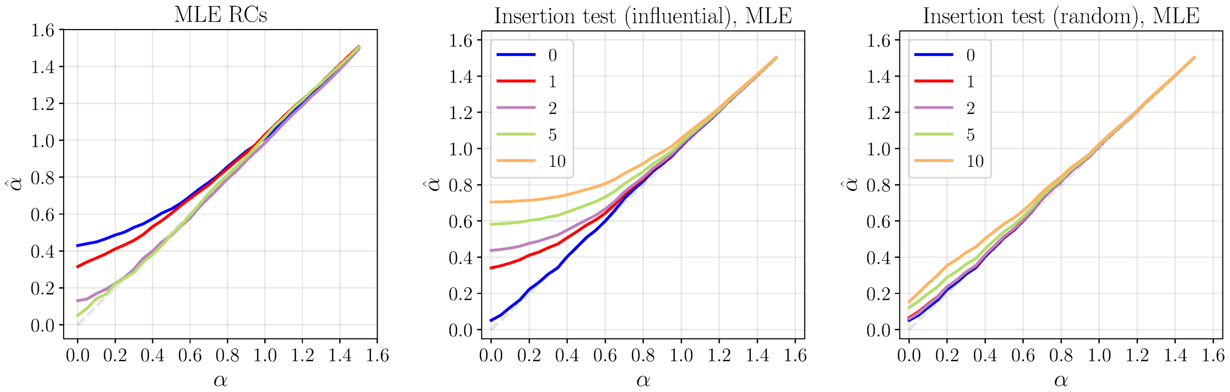

Figure 6. *MLE's response curves across acquisitions for one BB image (left), after inserting 4σ outliers at pixels with large $o_i(4\sigma_i)$ (middle), and after inserting 4σ outliers at randomly selected pixels (right).*

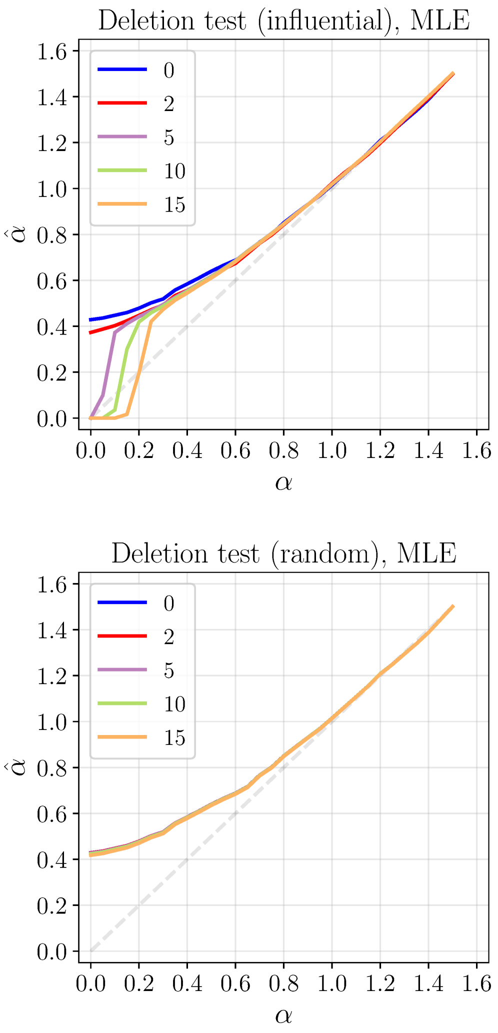

Figure 7. *MLE's response curves after deleting pixels with the largest o_i (top) and randomly selected pixels (bottom) for one BB image.*

Influential pixels (DLR)

To identify pixels in cover images that most contribute to the output of the DLR, we used the Integrated Gradients (IG) attribution technique [30]. However, we could not find a good base image for the cover for the attribution to work. To resolve this problem and in order to relate the attribution to a statistical model, we switched to a dataset of artificial but realistic looking images where pixels follow a known model but kept the regressor in the form of a trained network (DLR).

The artificial image dataset was prepared in a similar manner as in [6, 18]. BB images were simply denoised and then independent Gaussian noise with unequal variances was added to force a known cover model. In more detail, for each image $\mathbf{c}$ from BB, first we computed the pixel variances σ_i^2 using MiPOD's variance estimator. Then, the image was denoised using the wavelet denoising filter [26] with $\sigma_{\text{Den}} = 10$. Rounding the pixel values to integers, we denote the resulting denoised image with n pixels $\boldsymbol{\mu}(\mathbf{c}) \in \{0, \ldots, 255\}^n$. After denoising, the image dynamic range was narrowed to $[15, 240]$ to make sure the pixel values after noisification fit within the [0,255] range with high probability. The pixels in the artificial version of the same image follow a multi-variate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}(\mathbf{c}), \boldsymbol{\Sigma}(\mathbf{c}))$ with a diagonal covariance $\boldsymbol{\Sigma} = \text{diag}\left(\frac{1}{4}\sigma_1^2, \ldots, \frac{1}{4}\sigma_n^2\right)$. The pixels are finally rounded to integers and clipped to $[1, 254]$. We abbreviate this artificial version of BB as BB1/2, since the noisification is done with $\sigma_i/2$.

For images from BB1/2, there exists a natural base image for the IGs – the denoised image $\boldsymbol{\mu}$. Notice that the images from BB1/2 exactly follow the model within which we study the MLE.

The deletion experiment was executed by replacing pixels with the largest attribution with pixels from the denoised image $\boldsymbol{\mu}$. In some images, the cover bias decreased significantly even after removing only a few pixels. To scale up this experiment, we selected 388 images from the BB1/2 testing set with cover bias larger than 0.3. Figure 8 left shows the cover bias averaged over all 388 images after deleting top k pixels with the largest attribution. In contrast, deleting k randomly chosen pixels has no effect on the cover bias. We verified that pixels with the largest attribution indeed have large outlierness o_i (15) by selecting 20 pixels with the largest attribution from each image (20×388 pixels in total) and plotting the histogram of their outlierness o_i. Figure 8 middle shows that this histogram has a much thicker right tail when compared to o_i of 20 randomly selected pixels from each image.

Next, we studied whether a large positive cover bias can be introduced by inserting 4σ outliers at pixels with large $o_i(4\sigma_i)$ (the insertion experiment). To this end, we selected 288 images from BB1/2 with cover bias in the interval $[0.01, 0.05]$. Figure 8 right shows the average cover bias after replacing k pixels with the largest $o_i(4\sigma_i)$ when averaged over all 288 images. We observe that the DLR starts exhibiting a positive bias, which increases when adding more outlier pixels. In contrast, the cover bias is virtually unchanged when the pixels were chosen randomly.

Note that in both experiments, it takes more pixels to alter DLR's output that for the MLE. This is to be anticipated since the DLR is a data-driven estimator and thus is not completely aware of the source model unlike the MLE. To confirm the effect of influential pixels as identified by attribution and outlierness o_i, we verified that altering DLR's output is not easily achieved by modifying randomly selected pixels.

Conclusions

Quantitative steganography detectors output an estimate of the embedded secret message length. Originally conceived of and analyzed for detection of LSB replacement, quantitative detectors can be constructed for any embedding scheme using machine learning. The estimation error of such detectors for content-adaptive stego schemes exhibits different properties than for non-adaptive steganography. The estimator bias strongly depends on the payload and the distribution of outputs no longer satisfies the shift hypothesis. This is because content-adaptive embedding preferably modifies pixels where the embedding changes are the hardest to detect. The detector output as a function of payload for a fixed cover image thus naturally strongly depends on content.

In this paper, we study this dependence both experimentally for a deep learning payload regressor and theoretically from a model of the cover image and a maximum likelihood estimator. The MLE exhibits trends w.r.t. payload that remarkably closely match the trends observed for payload regressors built with deep learning, which allows us to better understand the estimator error, and the cover bias in particular, as a function of true payload. The MLE reveals that a large positive cover bias is often due to only a small number of "outlier" cover pixels that the estimator mistakens for evidence of embedding. The same kind of outlier pixel values affects the deep learning regressor. This was shown on an artificial dataset by analyzing pixels with the largest attributions. Similarly, we demonstrated that an image with a very small cover bias can be perturbed to exhibit a large cover bias by changing only a small number of carefully selected pixels. This effect was first established for the MLE and then also for a deep learning detector.

Additionally, we introduced two new concepts for benchmarking quantitative detectors, the reactive and estimable payloads, which both depend on cover image content. These quantities join global error measures, such as MSE and MAE, for comparing quantitative detectors in practice.

Future work will be directed towards analyzing quantitative detectors in the JPEG domain. Moreover, we intend to generalize the concept of influential cover pixels to binary detectors of steganography.

Acknowledgments

The work on this paper was supported by NSF grant No. 2028119.

References

[1] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[2] P. Bas and T. Furon. BOWS-2. `http://bows2.ec-lille.fr`, July 2007.

[3] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[4] R. Böhme and A. D. Ker. A two-factor error model for quantitative steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 59–74, San Jose, CA, January 16–19, 2006.

[5] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[6] M. Boroumand, J. Fridrich, and R. Cogranne. Are we there yet? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2019*, San Francisco, CA, January 26–30, 2019.

[7] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In D. Borghys and P. Bas, editors, *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, 2021. ACM Press.

[8] M. Chen, M. Boroumand, and J. Fridrich. Deep learning regressors for quantitative steganalysis. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

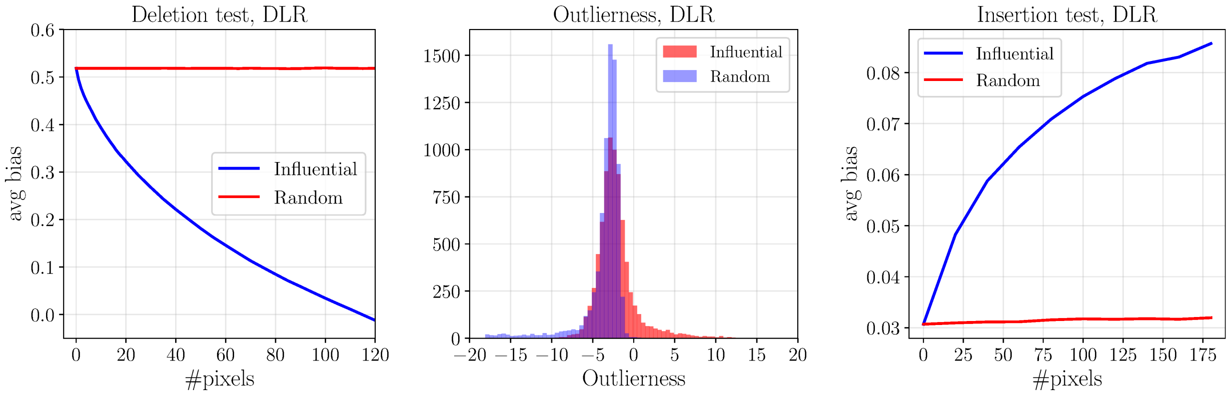

Figure 8. *Left: Average cover bias after deleting k pixels with the largest attribution for the DLR (left) and the histogram of their outlierness o_i (middle) contrasted with the outlierness of randomly selected pixels. Right: Average cover bias after inserting 4σ outliers at pixels with the largest $o_i(4\sigma_i)$ contrasted when inserting the same number of outliers at randomly selected pixels. The deletion (insertion) experiments were conducted over 388 (288) images from BB1/2 with the largest (smallest) cover bias.*

[9] S. Chutani and A. Goyal. Improved universal quantitative steganalysis in spatial domain using ELM ensemble. *Multimedia Tools and Applications*, 77:7447–7468, 2018.

[10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, June 20–25, 2009.

[11] S. Dumitrescu and X. Wu. LSB steganalysis based on higher-order statistics. In A. M. Eskicioglu, J. Fridrich, and J. Dittmann, editors, *Proceedings of the 7th ACM Multimedia & Security Workshop*, pages 25–32, New York, NY, August 1–2, 2005.

[12] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via Sample Pairs Analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[13] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.

[14] J. Fridrich, M. Goljan, and R. Du. Reliable detection of LSB steganography in grayscale and color images. In J. Dittmann, K. Nahrstedt, and P. Wohlmacher, editors, *Proceedings of the ACM, Special Session on Multimedia Security and Watermarking*, pages 27–30, Ottawa, Canada, October 5, 2001.

[15] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[16] J. Fridrich and J. Kodovský. Steganalysis of LSB replacement using parity-aware features. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 31–45, Berkeley, California, May 15–18, 2012.

[17] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[18] E. Kaziakhmedov, E. Dworetzky, and J. Fridrich. Limits of data driven steganography detectors. In A. Bharati, D. Henriques Moreira, and Y. Yousfi, editors, *The 11th ACM Workshop on Information Hiding and Multimedia Security*, Chicago, IL, 2023. ACM Press.

[19] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of Lecture Notes in Computer Science, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[20] A. D. Ker. Fourth-order structural steganalysis and analysis of cover assumptions. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 25–38, San Jose, CA, January 16–19, 2006.

[21] A. D. Ker. Derivation of error distribution in least squares steganalysis. *IEEE Transactions on Information Forensics and Security*, 2:140–148, 2007.

[22] A. D. Ker. A weighted stego image detector for sequential LSB replacement. In *Proc. International Workshop on Data Hiding for Information and Multimedia Security (part of IAS 2007)*, pages 453–456. IEEE Computer Society, 2007.

[23] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W.

Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[24] J. Kodovský and J. Fridrich. Quantitative structural steganalysis of Jsteg. *IEEE Transactions on Information Forensics and Security*, 5(4):681–693, December 2010.

[25] J. Kodovský and J. Fridrich. Quantitative steganalysis using rich models. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2013*, volume 8665, pages O 1–11, San Francisco, CA, February 5–7, 2013.

[26] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.

[27] T. Pevný, J. Fridrich, and A. D. Ker. From blind to quantitative steganalysis. *IEEE Transactions on Information Forensics and Security*, 7(2):445–454, 2011.

[28] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[29] Y. Sun and T. Li. A method for quantitative steganalysis based on deep learning. In *2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pages 302–309, September 28-30, 2019.

[30] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML. JMLR.org, 2017.

[31] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *Image Processing, IEEE Transactions on*, 23(5):1–14, May 2014.

[32] S.T. Veena and S. Arivazhagan. Quantitative steganalysis of spatial LSB based stego images using reduced instances and features. *Pattern Recognition Letters*, 105(C):39–49, April 2018.

[33] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[34] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[35] M. Yedroudj, M. Chaumont, and F. Comby. How to augment a small learning set for improving the performances of a CNN-based steganalyzer? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[36] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudj-net: An efficient CNN for spatial steganalysis. In *IEEE ICASSP*, pages 2092–2096, Alberta, Canada, April 15–20, 2018.

[37] Y. Yousfi, J. Butora, Q. Giboulot, and J. Fridrich. Breaking ALASKA: Color separation for steganalysis in JPEG domain. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[38] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, May 2018.

[39] T. Zhang and X. Ping. A fast and effective steganalytic technique against Jsteg-like algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 307–311, Melbourne, FL, March 9–12, 2003.

[40] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical decision methods in hidden information detection. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 163–177, Prague, Czech Republic, May 18–20, 2011.

Author Biography

Edgar Kaziakhmedov received M.S. degree in Applied Mathematics and Physics from Moscow Institute of Physics and Technology, Moscow, in 2020. He is currently pursuing PhD degree in electrical engineering at Binghamton University. His research areas lie within digital image steganalysis and steganography, neural network based image processing and digital media forensics.

Eli Dworetzky is currently pursuing his PhD in electrical and computer engineering at Binghamton University. His research currently focuses on image steganography and steganalysis. He received an MS in computer engineering from Binghamton University in 2021.

Jessica Fridrich is Distinguished Professor of Electrical and Computer Engineering at Binghamton University. She received her PhD in Systems Science from Binghamton University in 1995 and MS in Applied Mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, and digital image forensics. Since 1995, she has received 23 research grants totaling over $13 mil that lead to more than 230 papers and 7 US patents.