Quantitative and Binary Steganalysis in JPEG: A Comparative Study

Ahmad ZAKARIA
LIRMM, Univ Montpellier, CNRS
Montpellier, France
ahmad.zakaria@lirmm.fr

Marc CHAUMONT
LIRMM, Univ Nîmes, CNRS
Montpellier, France
marc.chaumont@lirmm.fr

Gérard SUBSOL
LIRMM, Univ Montpellier, CNRS
Montpellier, France
gerard.subsol@lirmm.fr

***Abstract*—We consider the problem of steganalysis, in which Eve (the steganalyst) aims to identify a steganographer, Alice who sends images through a network. We can also hypothesise that Eve does not know how many bits Alice embed in an image.**

In this paper, we investigate two different steganalysis scenarios: Binary Steganalysis and Quantitative Steganalysis. We compare two classical steganalysis algorithms from the state-of-the-art: the QS algorithm and the GLRT-Ensemble Classifier, with features extracted from JPEG images obtained from BOSSbase 1.01. As their outputs are different, we propose a methodology to compare them.

Numerical results with a state-of-the-art Content Adaptive Embedding Scheme and a Rich Model show that the approach of the GLRT-ensemble is better than the QS approach when doing Binary Steganalysis but worse when doing Quantitative Steganalysis.

***Index Terms*—Steganography, Quantitative Steganalysis, Binary Steganalysis, Multi-class Steganalysis, JPEG**

I. Introduction

Steganography alters innocuously looking cover objects to communicate in concealment. The science of detection of hidden data is called steganalysis. The research proposed in this paper focuses on steganalysis in digital images, probably the most studied cover objects. More precisely, we use images coded in JPEG which is the most common format.

Recent years have seen remarkable progress in steganography and steganalysis in JPEG images. Syndrome Trellis Codes [1] gave birth to numerous modern, Content-Adaptive data hiding algorithms in JPEG domain [2], where the embedding changes concentrate in textured and noisy regions which modifications are hard to detect.

Among the different steganalysis scenarios, two of them are interesting us: Binary Steganalysis and Quantitative Steganalysis. The first aims to make a binary decision whether there is or not a hidden message in an image, and has been remarkably improved over the past few years with the introduction of Rich Media models [3]–[6], and Ensemble Classifier [7]–[9]. The second scenario, which aims to estimate the payload (a null payload corresponds to a cover image), was introduced in [10] and has not been so much studied over the recent years. Researchers achieved the last significant improvement in [11] by using the recently proposed Rich Models.

In this paper, we propose to analyse and compare the performance of these two scenarios for JPEG steganalysis, without any assumption on the payload. For each scenario, we use a state-of-the-art algorithm. One difficulty is to compare scenarios which have different outputs: Binary or Quantitative (i.e. real) values.

After briefly summarizing the tested steganalysis algorithms (§ II), we present the comparison procedure (§ III) followed by the design of our experiments in § IV. Results and discussion are presented in § V and some conclusions are given in § VI.

II. Presentation of the algorithms

The first scenario is Binary Steganalysis which is based on the GLRT-ensemble Classifier (GLRT) [9]. This algorithm leverages the advantages of Optimal Detectors and Steganalysis machine learning approaches to employ an accurate statistical model for the base learners' projections in an Ensemble classifier [7]. Each base learner is a Fisher Linear Discriminant (FLD) classifier trained on a uniformly randomly selected subset of features, and then, its projection $\mathbf{v} \in \mathbb{R}^L$ is cast within hypothesis testing theory. The statistical hypothesis test here is a mapping $\delta : \mathbb{R}^L \mapsto \{H_0, H_1\}$ such that hypothesis H_i is accepted if $\delta(\mathbf{v}) = H_i$. Notice that this algorithm works without any assumption on the payload.

The second scenario is Quantitative Steganalysis. Notice that payload estimation p can be continuous or discrete. For this scenario, we selected the algorithm proposed in [11] (QS algorithm). It is a machine learning regression framework that assembles, via the process of gradient boosting, a large number of simpler base learners built on random subspaces of the original high-dimensional feature space. Each base learner is a Regression Tree adapted to reflect the specific nature of high-dimensional feature spaces in Steganalysis.

It is worth mentioning that in our experiments, both GLRT and QS algorithms use the same feature vectors for training, as depicted in *Fig*. 1. This condition is crucial for obtaining meaningful results.

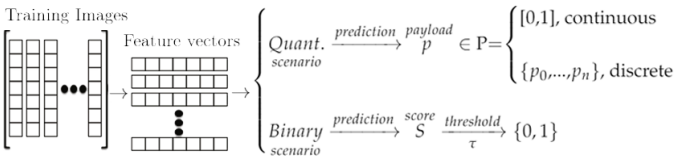

Fig. 1. Schematic representation summarising the binary and quantitative Steganalysis scenarios.

III. Comparison procedure

The fact that the results of the two algorithms are in different forms obliges us to post-process them before comparing them. A decision has been made to follow two different scenarios in our experiments:

-A **binary scenario**, where we construct a Binary Steganalysis algorithm from the QS regressor to compare its results with the original GLRT classifier. We name the QS regressor the *QS-binary*.

-A **quantitative scenario**, where we construct two quantitative algorithms, the GLRT-multiclass and the GLRT-regression, to compare their results to the original QS algorithm.

A. Binary scenario

In this scenario, we have to transform estimated payloads given by the QS algorithm into a binary decision. For this, we propose to construct the QS-binary algorithm.

QS-binary algorithm: The QS-binary algorithm uses thresholding to transform the n estimated payloads given by the QS algorithm into a binary decision (0=cover / 1=stego):

$$\begin{cases} 1 & p_i > p_\tau \\ 0 & p_i \leq p_\tau \end{cases} \tag{1}$$

where for the *ith* image vector, $p_i \in P = [0,1]$ is the payload predicted from the original QS regressor. p_τ is a fixed threshold over P calculated in the validation phase and optimized to minimize the probability of error $Pe = {}^1/_2\,(probab.\ of\ false\ alarm + probab.\ of\ missed\ detection)$ when there is the same number of cover and stego image vectors.

This way, from a list of payloads, we create a binary (cover / stego) decision comparable to the results of the GLRT classifier. To measure the performance, we calculate the probability of error Pe.

To compare GLRT and QS-binary, we also draw the Receiver Operating Characteristic (ROC) curves for both.

B. Quantitative scenario

To obtain quantitative results from the GLRT classifier, we construct two quantitative algorithms, the GLRT-multiclass and the GLRT-regression, to compare their results to the original QS algorithm. Below, we explain how to build the GLRT-regression and the GLRT-multiclass algorithms.

GLRT-regression algorithm: The GLRT-regression algorithm is a piecewise linear regression model, trained on a set of scores $S \in \mathbb{R}^L$ given from the GLRT classifier, to estimate the payloads $p \in P$, with $P = [p_0, p_n]$:

$$p = \begin{cases} p_0 & s \leq p_0 \\ a \times s & p_0 < s \leq p_n \\ p_n & p_n < s \end{cases} \tag{2}$$

Where $s \in S$ is the score for an image vector obtained from an image vector by the GLRT classifier before the thresholding . All predictions belongs to the interval $[p_0, p_n]$. $a \in \mathbb{R}^*$ is a slope of a linear function, $p = a \times s$, whose construction is based on the assumption that the scores follow a standardised Gaussian distribution under cover and all different payload sizes hypotheses, and the "shift hypothesis" (the shift is proportional to the payloads). That is why the regression function goes through $p_0 = 0$. Finally, we calculate the scores from the testing data, the same way as we did for the training data, to use them for predictions of payloads using our regression model of *Eq.* (2).

GLRT-multiclass algorithm: The GLRT-multiclass algorithm can be created, in the case we have a discrete range of payloads $P = \{p_0, \ldots, p_n\}$. We thus use a one-vs-one multi-class classifier which predicts a class by calculating the maximum of votes given by applying the GLRT between each couple of classes. We formalize it as follows: For n classes of payloads, let $i, j, k \in [0, n]$. Let I be an image vector. Let c_i be a class for payload p_i. Let $\zeta_{i,j}$ be a binary classifier between c_i and c_j such that $i < j$. These are $(n-1)n/2$ classifiers. Let V be the vector of votes where $V[k]$ contains the votes for c_k . We train all $\zeta_{i,j}$ then we test them on our testing data. The final decision for I is $c_k[I]$. It is calculated as follows: For all $\zeta_{i,j}[I]$, if $\zeta_{i,j}[I]$ is equal to c_i then $V[i] = V[i] + 1$, else $V[j] = V[j] + 1$. Finally, $k = \underset{k}{argmax}\ V[k]$.

IV. Experimental protocol

A. Algorithm parameters

To precisely examine the Steganalysis algorithms, we use them with their optimal parameters. In the training phase, the GLRT classifier searches for the optimal value of feature space dimensionality d_{sub} and automatically determines the number of base learners L [9]. For the QS algorithm, the hyper-parameters are set manually, and they are experimentally optimised, we use the same values as fixed in the original paper [11].

The two Steganalysis algorithms, the J-UNIWARD embedding algorithm and the GFR feature extractor, are implemented in MATLAB. Their implementations are

available for download at the research code web page of the Binghamton University[1].

In the quantitative scenario, we use the Sklearn Python package to calculate the Root Mean Squared Error and the Mean Absolute Error. Also, the Matplotlib Python package is used to plot the regression function.

B. JPEG feature vector construction

The first step in our experimental protocol is the preparation of the images data. We convert 10000 512 × 512 grey-scale spatial images from BOSSbase into JPEG images, using the MATLAB's command imwrite, with quality factors 75 and 95. Then we use the advanced adaptive steganographic scheme J-UNIWARD to generate stego images with different embedding rates { 0.1, 0.2, 0.3, 0.4, 0.5} bits per nonzero AC DCT coefficient (bpnzAC). We restrict our work to this range following the same scale used in [9].

Next, we use the 17,000-dimensional JPEG domain Rich Model (GFR), proposed in [5], to extract the feature vectors from the cover and stego images.

Next, we clean the feature vectors from NaN values (it occurs when the feature values are constant over images) and from constant values, to obtain 16750-dimensional feature vectors. Finally, we normalise the data using an algorithm proposed in [12]. There are 10000 feature vectors for cover images and 10000 feature vectors for each payload which gives a total of 60000 feature vectors.

C. Database construction

For GLRT, GLRT-regression and GLRT-multiclass, we use 10000 covers and 10000 stegos such that each five different payload sizes are equally distributed. A ratio of 1/5 is respected when selecting stegos; we choose the first 10% of stegos with payload 0.1 bpnzAC that corresponds to the first 10% of covers, the second 10% of stegos with payload 0.2 bpnzAC that corresponds to the second 10% of covers, and so on. . . .

Each time we randomly split the data into two equal parts, 50% for the training and validation phase and 50% for the testing phase.

For QS and QS-binary, the image vectors are with payloads 0, 0.1, 0.2, 0.3, 0.4, 0.5 bpnzAC such that the total number of features vectors is 20000. We prepare them respecting a ratio of 1/6 for each payload. Additionally, we split the input data between training, validation or testing phases, with 8400 vectors for training, 2100 for validation and 9500 for testing.

D. Comparison procedures

We explained in § III, the binary scenario and the quantitative scenario, where we construct the algorithms. In this section, we explain how we apply these algorithms to our data.

[1] HTTP://dde.binghamton.edu/download/

1) Binary scenario: Below we explain how to use the QS-binary algorithm to compare its results with the original GLRT classifier.

QS-binary: First, we apply the QS algorithm (training/testing) on the data that we already prepared as in § IV-C and obtain the predicted payloads. Next, as explained in § III, we calculate p_τ that minimize the probability of error Pe, with precision degree 10^{-4}, in the validation phase. Finally, we classify the predicted payloads in the testing phase using p_τ and $Eq.$ (1). This way, from a list of payloads {0, 0.1, 0.2, 0.3, 0.4, 0.5}, we create a binary (cover/stego) data comparable to the results of the GLRT classifier.

2) Quantitative scenario: Here we compare the algorithms in the quantitative scenario on $P = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, by applying the GLRT-regression and the GLRT-multiclass algorithms as explained in § III.

GLRT-regression: We train the regression model explained in § III on scores obtained from the GLRT classifier to predict a payload $p \in P$, as in $Eq.$ (2). The parameters p_0 is 0 and p_n is 0.5 as shown in $Fig.$ 2. Note that, in $Fig.$ 2 the regression is given for QF 75, and that the regression slope is a little bit more steep for QF 95. To train the model, we use the following procedure:

First, we apply the GLRT training algorithm which finds the optimal values of d_{sub} and L parameter and trains each FLD base learner.

Next, we compute the projection onto all base learners for training samples themselves (the regular use of the algorithm is to calculate the projection onto all base learners for testing samples then to continue into the testing phase). The projections are under $H0$ for training covers, and under $H1$ for training stegos, they all will be normalised by the covariance under $H0$ and by subtracting the mean value under $H0$.

Next, we compute the Generalised Likelihood Ratio (GLR) test which is given by the projection onto the vector of the mean projections under $H1$ normalised by the norm to ensure that the GLR follows a standardised Gaussian distribution under $H0$. Further details are available in [9]. Next, we train a regression model, $Eq.$ (2) on the obtained training GLRs to predict payloads.

Finally, we calculate the GLRs from the testing data, the same way as we did for the training data, then we use them for predictions of payloads using our regression model.

GLRT-multiclass: Our numerical range of payloads to be predicted is discrete, which is close to the multiclass classification problem, so we apply the GLRT-multiclass classifier explained in § III:

This will be a one-vs-one classifier that uses the GLRT classifier to do the binary classification between each couple of classes. We represent the classes by numbers between 0 and 5 instead of using their real values for a simpler use, hence we get a list of votes $V = \{ V_{0,1}, V_{0,2}, V_{0,3}, V_{0,4}, V_{0,5}, V_{1,2}, V_{1,3}, V_{1,4}, V_{1,5}, V_{2,3}, V_{2,4}, V_{2,5},$

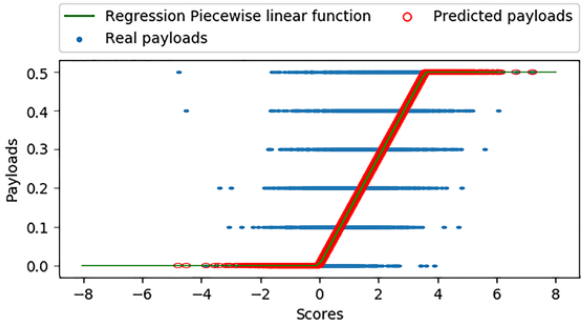

Fig. 2. Quantitative scenario: schematic description for the regression piecewise linear function for quality factor 75.

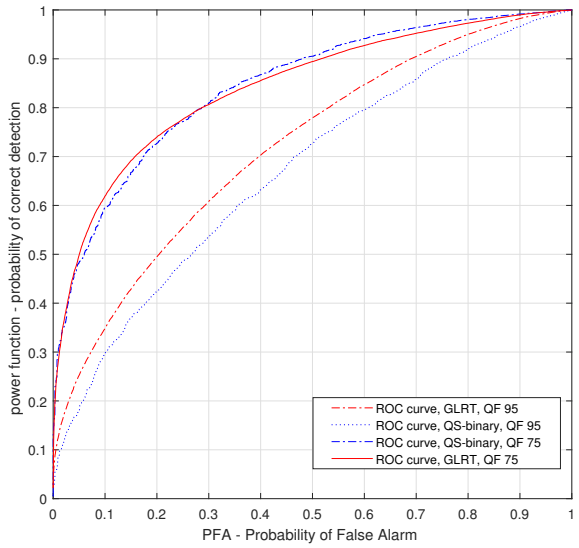

Fig. 3. Binary scenario: empirical ROC curves for the QS-binary and the GLRT algorithms, for quality factor 75 and 95.

$V_{3,4}$, $V_{3,5}$, $V_{4,5}$ } of binary decisions calculated from the binary classifiers for each image. These will be used to calculate the final decision as explained in § III.

V. Results & Discussion

A. Binary scenario

In the case of binary scenario, *Tab*. I shows a small superiority of the GLRT classifier with a difference in *Pe* of about 2% for quality factor 75 and about 4% for quality factor 95.

TABLE I
BINARY SCENARIO: PROB. OF ERROR *Pe*, OF GLRT AND QS-BINARY APPROACHES, FOR QF 75 ($p_\tau = 0.1482$) AND 95 ($p_\tau = 0.2647$) .

	QS-binary	GLRT
QF 75	0.2479	0.2275
QF 95	0.3795	0.3438

TABLE II
BINARY SCENARIO: DETECTION POWER OF GLRT AND QS-BINARY APPRAOCHES FOR $\alpha_0 = 0.055$, FOR QF 75 AND 95.

Quality factor	Payload	Clairvoyant, GLRT	Payload Mixture, GLRT	Trained for R=0.5, GLRT	Payload Mixture, QS-binary
75	0.5	0.9367	0.9151	0.9348	0.8829
	0.4	0.8091	0.7806	0.7878	0.6853
	0.3	0.5467	0.5258	0.5015	0.3739
	0.2	0.2665	0.2524	0.2165	0.1691
	0.1	0.1068	0.1032	0.0925	0.0823
95	0.5	0.5526	0.5487	0.5583	0.4364
	0.4	0.3741	0.3635	0.3303	0.2779
	0.3	0.2158	0.1974	0.1746	0.1629
	0.2	0.1188	0.1071	0.0981	0.0943
	0.1	0.0710	0.0679	0.0649	0.0655

TABLE III
BINARY SCENARIO: PROBABILITY OF ERROR *Pe*, FOR GLRT AND QS-BINARY APPROACHES, FOR QF 75 AND 95 IN THE CASE OF DIFFERENT TRAINING SCENARIOS.

Quality factor	Payload	Clairvoyant, GLRT	Payload Mixture, GLRT	Trained for R=0.5, GLRT	Payload Mixture, QS-binary
75	0.5	0.0585	0.0684	0.0596	0.2128
	0.4	0.1059	0.1198	0.1137	0.2203
	0.3	0.1842	0.1975	0.2006	0.2502
	0.2	0.2932	0.3075	0.3180	0.3253
	0.1	0.4059	0.4188	0.4277	0.4333
95	0.5	0.1975	0.2115	0.1954	0.2511
	0.4	0.2707	0.2802	0.2774	0.3305
	0.3	0.3490	0.3555	0.3544	0.4017
	0.2	0.4185	0.4272	0.4247	0.4565
	0.1	0.4714	0.4740	0.4740	0.4849

TABLE IV
QUANTITATIVE SCENARIO: AVERAGE PREDICTED ERROR (AVG), ROOT MEAN SQUARED ERROR (RMSE) AND MEAN ABSOLUTE ERROR (MAE) FOR GLRT-REGRESSION, QS AND GLRT-MULTICLASS APPROACHES, FOR QF 75 AND 95.

Payload	GLRT-regression			GLRT-multiclass			QS		
	QF 75								
	AVG	RMSE	MAE	AVG	RMSE	MAE	AVG	RMSE	MAE
0	0.0541	0.0960	0.0541	0.0692	0.1298	0.0692	0.1312	0.1568	0.1312
0.1	0.1334	0.1229	0.0989	0.1197	0.1309	0.1017	0.1645	0.1094	0.0812
0.2	0.1614	0.1355	0.1141	0.1876	0.1359	0.1070	0.2182	0.0919	0.0749
0.3	0.2292	0.1544	0.1290	0.2868	0.1331	0.0980	0.2883	0.0909	0.0745
0.4	0.2826	0.1858	0.1495	0.3797	0.1148	0.0809	0.3623	0.0919	0.0704
0.5	0.3524	0.2103	0.1477	0.4548	0.0949	0.0452	0.4251	0.1021	0.0759
All		0.1508			0.1232			0.1071	
	QF 95								
	AVG	RMSE	MAE	AVG	RMSE	MAE	AVG	RMSE	MAE
0	0.0908	0.1498	0.0908	0.1494	0.2362	0.1494	0.2413	0.2506	0.2413
0.1	0.1431	0.1566	0.1224	0.1627	0.1925	0.1527	0.2478	0.1625	0.1478
0.2	0.1393	0.1466	0.1266	0.2084	0.1886	0.1646	0.2613	0.0916	0.0736
0.3	0.1826	0.1967	0.1703	0.2619	0.1896	0.1589	0.2816	0.0731	0.0599
0.4	0.2700	0.2200	0.1796	0.3420	0.1838	0.1368	0.3096	0.1166	0.0986
0.5	0.2821	0.2795	0.2180	0.3993	0.1874	0.1007	0.3422	0.1747	0.1580
All		0.1915			0.1963			0.1448	

Tab. II and *Tab*. III present respectively, the detection power (i.e. the probability of detection of a stego image within all the examined stego images) for a probability of false alarm of $\alpha_0 = 0.055$ and the minimal total probability of error Pe.

We obtain results for the GLRT approach according to 3 different training scenarios:

- in the clairvoyant test, the embedding rate is known, i.e. training and testing are performed with the same payload.
- training is performed on a uniform mixture of payloads.
- training is performed with a fixed payload $R = 0.5$.

Results for the QS-binary approach are obtained by training on a uniform mixture of payloads.

We can conclude that the detection power is better for GLRT approach whatever the training scenario (clairvoyant, payload mixture or fixed payload) compared to the QS-binary approach.

B. Quantitative scenario

In *Tab*. IV, we compare the performance of the QS, the GLRT-regression, and the GLRT-multiclass approaches, all implemented with GFR features [5].

Unlike the binary scenario, here, the QS approach provides better results than the GLRT-regression and the GLRT-multiclass ones, with in average about 4% smaller RMSE than the GLRT-regression and about 2% lower RMSE than the GLRT-multiclass for quality factor 75. For quality factor 95, QS approach gives about 4% smaller RMSE than the others. But this is only true for high payloads. For small payloads, the QS approach gives less good results.

Note from *Fig*. 3 and from *Tab*. IV that in both quantitative and binary scenarios, the results are better for quality factor 75 than quality factor 95, especially for small payloads. This is due to image compression that makes the embedding changes more straightforward to detect [5].

VI. Conclusions and perspectives

In this paper, we investigated two state-of-the-art steganalysis algorithms, QS and GLRT. The goal was to compare them and find the best to use in our future work in pooled steganalysis.

Numerical results based on Content Adaptive Embedding Scheme and Rich Model show that the GLRT approach is slightly better than the QS one when doing Binary Steganalysis and that GLRT approach is marginally worse than the QS one when doing the Quantitative Steganalysis. GLRT approach seems more accurate to estimate payload when it is small; This may be due to the accuracy of the original GLRT classifier which is good for small payload [9].

Despite the broad difference between the binary and the quantitative scenarios, using an algorithm specially developed for one scenario in the other scenario context gives competitive since the difference is between 2-4% in Pe or $RMSE$. It may be interesting to use this way of comparing algorithms developed for different scenarios to analyse new algorithms as [13] where scores given by a classifier are used to train a regression model for payload estimation.

In conclusion, these experiments open doors for multiple questions and possible uses of machine learning algorithms for Binary and Quantitative Steganalysis.

References

[1] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[2] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[3] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[4] V. Holub and J. Fridrich, "Phase-aware projection model for steganalysis of JPEG images," in *Media Watermarking, Security, and Forensics 2015*, vol. 9409. International Society for Optics and Photonics, 2015, p. 94 090T.

[5] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," in *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 2015, pp. 15–23.

[6] C. Xia, Q. Guan, X. Zhao, Z. Xu, and Y. Ma, "Improving GFR Steganalysis Features by Using Gabor Symmetry and Weighted Histograms," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017, pp. 55–66.

[7] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[8] R. Cogranne, V. Sedighi, J. Fridrich, and T. Pevnỳ, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?" in *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[9] R. Cogranne and J. Fridrich, "Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2627–2642, 2015.

[10] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis of digital images: estimating the secret message length," *Multimedia systems*, vol. 9, no. 3, pp. 288–302, 2003.

[11] J. Kodovskỳ and J. Fridrich, "Quantitative steganalysis using rich models," in *Media Watermarking, Security, and Forensics 2013*, vol. 8665. International Society for Optics and Photonics, 2013, p. 86650O.

[12] M. Boroumand and J. Fridrich, "Nonlinear Feature Normalization in Steganalysis," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017, pp. 45–54.

[13] M. Chen, M. Boroumand, and J. Fridrich, "Deep Learning Regressors for Quantitative Steganalysis," *Proc. IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, February, 2018.