MODE: Loss Function for Deep Steganalysis at Low False Positive Rate

1st Jan Butora
UMR 9189 CRIStAL
Univ. Lille, CNRS, Centrale Lille
Lille, France
jan.butora@cnrs.fr

2nd Patrick Bas
UMR 9189 CRIStAL
Univ. Lille, CNRS, Centrale Lille
Lille, France
patrick.bas@cnrs.fr

***Abstract*—Deep steganalysis has been crucial in detecting hidden messages in digital media for nearly a decade. However, its common security evaluation criterion-the probability of error under equal prior-fails to reflect real forensic challenges. In practice, low False Positive (FP) rates matter most but are only adjusted empirically post-training. Standard classifiers, trained with cross-entropy loss, optimize balanced error rates rather than minimizing FPs.**

We propose a framework that integrates the likelihood ratio test into the loss function to optimize deep classifiers for low FP rates. Our method outperforms standard cross-entropy and other modern approaches, as demonstrated on the BOSSBase dataset across FP rates of 10^{-3} to 10^{-1} in both uncompressed and JPEG domains.

***Index Terms*—steganalysis, Neyman-Pearson, false positive rate, deep learning**

I. INTRODUCTION

Steganography [8] covertly embeds messages within digital objects, modifying a cover object into a stego object with undetectable changes. A secret message can be retrieved using the correct stego key. Steganalysis aims to detect such hidden data, primarily in digital images. Due to the absence of robust statistical models for natural images, deep learning classifiers [3], [17], [18] dominate this binary classification task.

While convolutional neural networks (CNNs) outperform feature-based methods [9], [11], they are typically optimized with cross-entropy loss, which balances False Positive (FP) and False Negative (FN) rates. However, forensic applications demand ultra-low FP rates (below 10^{-3}), as false detections can lead to costly investigations and wrongful targeting. Despite this, research on optimizing detectors for low FP rates remains scarce [7], [16], with prior steganalysis efforts focusing on linear classifiers.

Pevny and Ker [15] introduced loss functions (e.g., exponential and logistic loss) for optimizing the $FP50$ metric—FP rate at 50% FN rate—by maximizing feature separation. More recently, alternative evaluation metrics have emerged. The ALASKA [5] challenge introduced the $MD5$ metric (FN at 5% FP rate), but even the winning team [19] selected model post-training rather than optimizing for this criterion. ALASKA 2 [6] introduced weighted AUC (wAUC) to emphasize low FP regions, yet top teams still relied on cross-entropy loss [20].

To bridge this gap, we propose a novel loss function leveraging the Neyman-Pearson lemma [14], inspired by anomaly detection methods such as PatMat [1] and DeepTopPush [13]. Unlike prior work, our approach directly optimizes deep learning detectors for low FP rates, improving forensic reliability while maintaining accuracy.

II. PRELIMINARIES AND PRIOR ART

Let $X \in \mathbb{R}^N$ be an image with N pixels and label $y \in \{0,1\}$ (0 for cover, 1 for stego). Let $\mathcal{C}_\Lambda$, and $\mathcal{S}_\Lambda$ denote the cover and stego images in a set $\Lambda \in$ {train,val,test}. Steganalysis is a binary hypothesis problem:

$$\mathcal{H}_0 : X \text{ is cover}, \quad \mathcal{H}_1 : X \text{ is stego}. \tag{1}$$

Since a purely statistical solution is infeasible, machine learning is used instead. Let $f : \mathbb{R}^N \to \mathbb{R}^2$ be a CNN outputting logits ϕ_c (cover) and ϕ_s (stego). Defining $\phi = \phi_s - \phi_c$, the probability of an image being stego is approximated using the sigmoid function:

$$\hat{y} = \sigma(\phi) = \frac{1}{1+e^{-\phi}}. \tag{2}$$

With this in mind, we assume that a detector outputs only a single logit ϕ. The detector then decides that a given image is a stego image when $\hat{y} > 1/2$, or equivalently when $\phi > 0$. This gives an implicit decision threshold at zero, though it can be adjusted via the ROC curve. The detector is typically optimized for this threshold, balancing FP and FN rates.

To optimize the classifier's weights with gradient-based methods, a binary cross-entropy loss function is used:

$$l(y,\hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y}). \tag{3}$$

For stego images, this minimizes $\log(1+e^{-\phi})$, while for covers, it maximizes the same function. This introduces a logistic transformation:

$$L(x) = \log(1+\exp(-x)), \tag{4}$$

whose derivative $\partial L(x)/\partial x = \sigma(x) - 1$, reduces emphasis on correctly classified samples. This prevents over-optimization on easy cases, enhancing detection of harder samples.

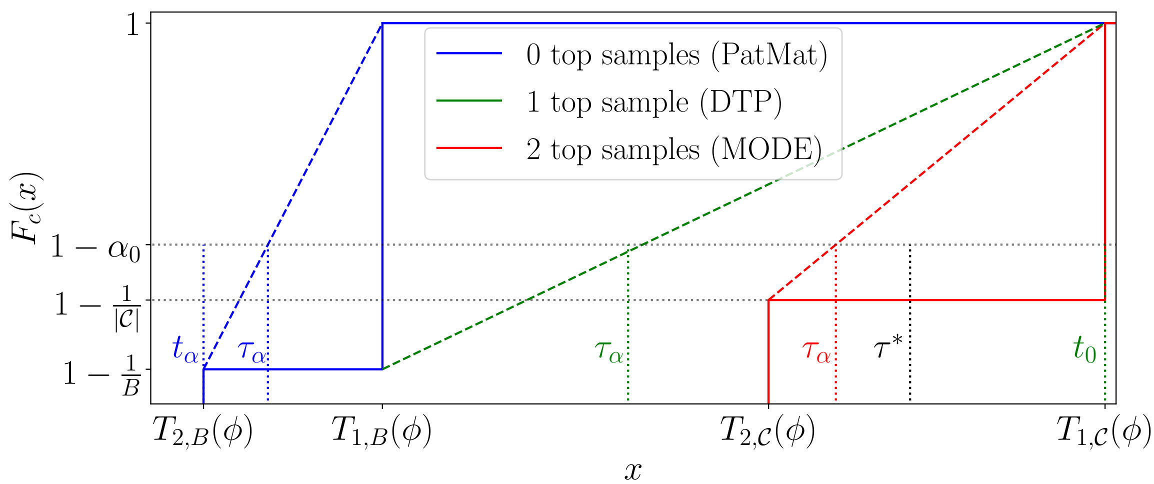

Fig. 1: Proposed estimation of the $(1-\alpha_0)$-quantile from an empirical cdf $F_c(x)$. B denotes the size of a cover minibatch, and $\mathcal{C}$ is the cover training dataset. Solid lines are the empirical cdfs, and dashed lines represent their linear approximations. Blue: empirical cdf estimated from a minibatch of B samples, green: empirical cdf with the top sample added to the minibatch, red: empirical cdf with the two top samples added to the minibatch. The spacing between the green cdf values should be $1/(B+1)$, which is omittted for simplicity. The true $(1-\alpha_0)$-quantile is denoted by τ^*, t_α denotes PatMat's threshold, and t_0 is the top sample used as a threshold for DeepTopPush.

A. Neyman-Pearson lemma

In this work, we build upon previously proposed methods of optimization at low FP rates, both of which are based on the Neyman-Pearson lemma [14]. Let's consider the observed logit ϕ as a realization of a random variable. Under the null (cover) hypothesis, we will denote the cumulative distribution function (cdf) of this variable as $F_c(x)$. Similarly, we will denote $F_s(x)$ the cdf under the stego hypothesis. Let us further denote α_0 the FP rate that we are willing to tolerate. It follows that the decision threshold associated with this FP rate is given by the cover $(1-\alpha_0)$-quantile:

$$\tau^* = F_c^{-1}(1-\alpha_0). \tag{5}$$

To optimize for this threshold, the Neyman-Pearson lemma states that the optimal test is the Likelihood Ratio Test, which, in this context, maximizes the stego distribution above τ^*, which is equivalent to minimizing $F_s(\tau^*)$.

For convenience, let $\phi(x)$ be the output logit of an image x, and define $T_{n,\Sigma}(\phi)$ as the n-th top sample among $\phi(x), x \in \Sigma$:

$$\sum_{x\in\Sigma}[\phi(x) > T_{n,\Sigma}(\phi)] = n. \tag{6}$$

B. Pat&Mat

Pat&Mat [1] (Precision At the Top & Mainly Automated Tuning) is a framework for binary classification focused on maximizing accuracy in low FP rate regions. For a given FP rate α_0, it estimates the threshold τ^* using the nearest cover logit at the $(1-\alpha_0)$-quantile, denoted as t_α. Optimization then maximizes the separation between stego logits and this threshold:

$$\text{minimize} \quad \sum_{s\in\mathcal{S}_{\text{train}}} l(t_\alpha - \phi(s)), \tag{7}$$

$$\text{subject to} \quad t_\alpha = T_{\lceil\alpha_0\cdot|\mathcal{C}_{\text{train}}|\rceil,\mathcal{C}_{\text{train}}}(\phi), \tag{8}$$

where $l(\cdot)$ is a convex surrogate of a 0-1 loss, set as the logistic function (4). While t_α approximates τ^*, it remains the closest cover logit rather than an exact estimate.

A key limitation is the need to estimate the $(1-\alpha_0)$-quantile across the dataset, restricting the original method to linear classifiers. A minibatch-based stochastic gradient descent was proposed, but optimization was limited to FP rates of 10^{-2}. This was improved in [13] by using larger (20k-sample) minibatches for malware detection. However, for CNN steganalyzers, GPU memory constraints significantly reduce batch size, making quantile estimation unreliable for smaller FP rates.

C. DeepTopPush

DeepTopPush [13] addresses the issue of insufficient batch samples for quantile estimation by maximizing the true positive (TP) rate at $\alpha_0 = 0$. The algorithm iteratively tracks the top cover sample, denoted t_0, adding it to every minibatch during training. If a cover image with a higher logit appears, t_0 is updated accordingly for subsequent training. The optimization problem is formulated as:

$$\text{minimize} \quad \sum_{s\in\mathcal{S}_{\text{train}}} l(t_0 - \phi(s)), \tag{9}$$

$$\text{subject to} \quad t_0 = T_{0,\mathcal{C}_{\text{train}}}(\phi), \tag{10}$$

As in Pat&Mat, we use (4) as the convex surrogate of $l(\cdot)$.

III. Proposed Method

A. Motivation

As shown in Section IV, PatMat and DeepTopPush effectively optimize for low FP rates compared to cross-entropy loss (3). However, both have limitations.

PatMat requires large minibatches to estimate thresholds accurately, needing $\sim \frac{10}{\alpha_0}$ images for an FP rate α_0, which is infeasible for $\alpha_0 \leq 10^{-3}$.[1] With a batch size B, it we can only estimate $(1 - s/B)$-quantile, where $s \in \{0, \ldots, B-1\}$ ($B = 32$ in our experiments). Consequently, choosing the nearest sample of τ^* will always select the largest sample in the minibatch, whenever $\alpha_0 < 1/B$. This can be problematic since the probability of randomly choosing a cover image whose logit is above the τ^* is α_0. This means that in a vast majority of optimization steps, the algorithm optimizes, in fact, for much higher FP rates than prescribed.

DeepTopPush avoids this issue by tracking the running maximum across all cover images, making it effective for extremely low FP rates. However, for cases where a small but nonzero FP rate (e.g., 10^{-6}) is acceptable, optimizing for those rates directly may yield better results.

Finally, both methods rely on a single cover sample to compute gradients per optimization step, while the rest are used only to estimate the threshold t_α.

B. The MODE Loss

We propose a new loss function for low-FP-rate optimization with two key improvements over previous methods:

1) Optimization for any given FP rate.
2) Leveraging all images (including covers) to better estimate the threshold for FP rate α_0.

1) Threshold estimation: Like Pat&Mat, we estimate the $(1-\alpha_0)$-quantile τ_α from cover images but introduce two key modifications. First, instead of selecting the nearest sample t_α, we compute a linear approximation of the cover empirical cumulative distribution function (cdf) $\hat{F}_c(x)$. Second, unlike DeepTopPush which tracks only the top cover sample, we track the top two to estimate the cdf slope for very small α_0. Note that we manually set the value of the cdf at the second-highest sample to $1 - 1/|\mathcal{C}_{\text{train}}|$.

Figure 1 illustrates this approach. In blue, we see an empirical cdf from a given minibatch with B cover images, which has $1/B$ spacing between the cdf values. Having the linear approximation (dashed), we find the decision threshold by inverting the cdf $\tau_\alpha = \hat{F}_c^{-1}(1-\alpha_0)$. From the discussion in the previous section, this estimate will in many cases underestimate the actual threshold we are looking for. We therefore add the top cover sample over the whole training dataset to the minibatch and use its linear approximation (green). Finally, a better estimate τ_α of τ^* can be obtained by adding the two top cover samples over the whole training dataset (red).

We want to emphasize that various enhancements to this linear cdf approximation are possible. On one hand, we can consider a non-linear approximations, and on the other hand, we can add more top samples to the minibatch to obtain the correct quantile. To do so, we would need approximately $k \sim \alpha_0 \cdot |\mathcal{C}_{\text{train}}|$ top samples. In our experiments, $|\mathcal{C}| = 7000$, so we would need 7 top samples for $\alpha_0 = 10^{-3}$. For simplicity, we restrict ourselves only to the 2 top samples and linear approximation of the cdf, leaving further improvements for future work.

2) Optimization: Having estimated the threshold τ_α, we formulate the steganalyst's test with the detector's logits as:

$$\begin{aligned} \mathcal{H}_0 &: \quad \phi \leq \tau_\alpha, \\ \mathcal{H}_1 &: \quad \phi > \tau_\alpha. \end{aligned} \tag{11}$$

We then model the classifier's output conditioned on the observation $x = \phi - \tau_\alpha$ with Bernoulli distribution $P(Y = 1|x = \phi - \tau_\alpha) = p$. Employing logistic transformation on p to obtain the shifted logits x, we obtain $p = \sigma(\phi - \tau_\alpha)$.

Maximizing the likelihood $\prod_i P(Y = y_i|X = \phi_i - \tau_\alpha)$ over all images is then equivalent to minimizing the cross-entropy:

$$\text{minimize} \quad \textstyle\sum_i y_i \log p_i + (1-y_i)\log(1-p_i), \tag{12}$$

$$\text{such that} \quad \tau_\alpha = \hat{F}_c^{-1}(1-\alpha), \tag{13}$$

where the minimization (12) can be written as

$$\text{min.} \ \textstyle\sum_i y_i \log(1+e^{\tau_\alpha - \phi_i}) + (1-y_i)\log(1+e^{\phi_i - \tau_\alpha}). \tag{14}$$

This resembles PatMat (7),(8) with the exception that the optimization is done on cover images too, and the threshold τ_α is found from an approximation of the cdf $\hat{F}_c(x)$. In fact, this differs from a cross-entropy minimization (3) that uses a zero threshold, only by considering a different threshold τ_α, computed from the cover samples. We believe the proposed strategy leads to a more stable estimation of the threshold τ^* and thus a better separation of the cover and stego distributions. Indeed, using only the top two samples to estimate τ^* can lead to a very noisy estimate in the case of heavy tail distributions. A case which is circumvent by penalizing the outlier cover images. We name the proposed loss function (14) *MODE*: **M**aximizing **O**ptimal **D**etector's **E**fficiency.

IV. Experimental Results

We now describe the dataset used to generate the datasets, as well as the training strategy of the steganalyzer.

A. Setup

We use 10,000 grayscale uncompressed images (512×512) from BOSSBase [2], split into training (7000), validation (1000), and testing (2000) sets. Cover images are embedded using HILL [12] at 0.3 bpp.

For JPEG images, covers are compressed with Libjpeg[2] at QFs 75, 95, and 100, then embedded with UERD [10] at 0.1 bpnzac.

We choose the JIN-pretrained [4] SRNet [3] as it allows us to only refine the detector on a limited amount of data, instead of training all of its parameters from a random initialization. For every tested algorithm and loss function, we first train this detector for 50 epochs using the Cross-Entropy (CE) loss function (3), with 64 randomly selected images in every mini-batch. The learning rate starts at 10^{-3} and is halved if

[1] Using an NVIDIA V100 GPU (32GB RAM), we can fit up to 64 images of size 512×512 per batch.

[2] http://libjpeg.sourceforge.net/

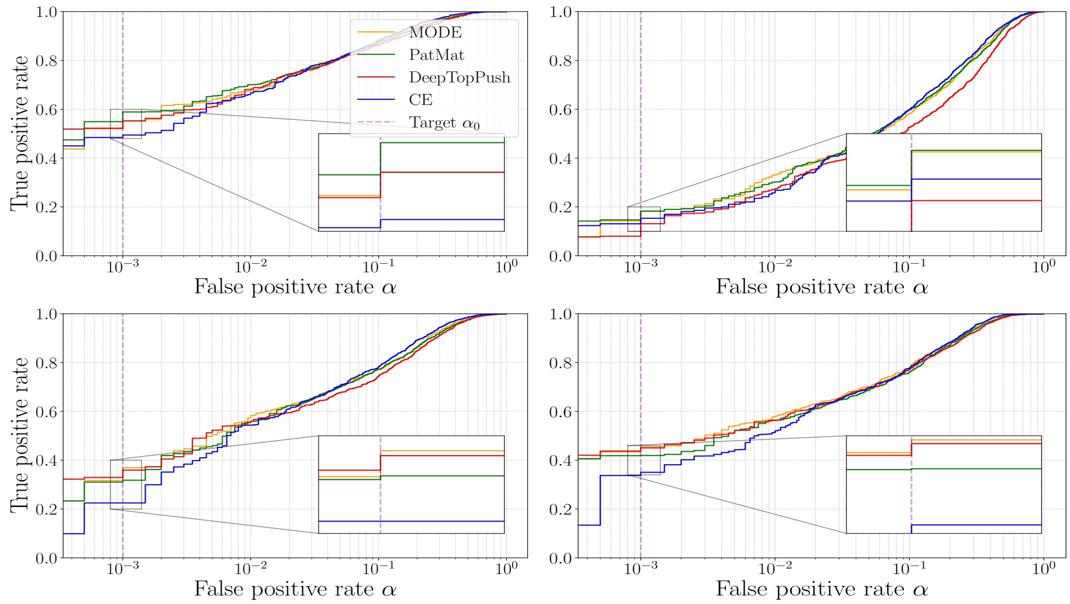

Fig. 2: ROC curves for the tested loss functions with $\alpha_0 = 10^{-3}$. From left to right: UERD, QF 75, QF 95, QF 100; HILL.

Loss	α_0	P_E	$P_D(\alpha)$ 10^{-3}	$P_D(\alpha)$ 10^{-2}	$P_D(\alpha)$ 10^{-1}
CE	-	0.1520	0.3505	0.5205	0.7765
DTP [13]	0	0.1577	0.4505	0.5645	0.7715
Pat&Mat [1]	10^{-2}	0.1617	0.4420	0.5730	0.7595
	10^{-3}	0.1577	0.4195	0.5635	0.7610
MODE	10^{-2}	**0.1483**	0.4475	**0.5815**	**0.7860**
	10^{-3}	0.1500	**0.4550**	0.5810	0.7805

TABLE I: Results for various FP rates. HILL, 0.3 bpp.

Loss	α_0	P_E	$P_D(\alpha)$ 10^{-3}	$P_D(\alpha)$ 10^{-2}	$P_D(\alpha)$ 10^{-1}
CE	-	0.1090	0.4945	0.6675	0.8815
DTP [13]	0	0.1162	0.5525	0.6830	0.8625
Pat&Mat [1]	10^{-2}	0.1095	0.5645	0.6920	0.8770
	10^{-3}	0.1123	**0.5890**	**0.7010**	0.8720
MODE	10^{-2}	0.1085	0.5600	0.6945	0.8805
	10^{-3}	**0.1075**	0.5530	0.6775	**0.8835**

TABLE II: Results for various FP rates. UERD, QF 75.

validation loss stagnates for 3 epochs. We then refine the CE-trained detectors for 50 more epochs with an initial learning rate of 10^{-4}, keepin all other hyperparameters as in [4].

Detection performance is measured by the TP rate $P_D(\alpha)$ (Detection probability), at FP rates $\alpha \in \{10^{-1}, 10^{-2}, 10^{-3}\}$.

B. Optimization at low FP rates

We now compare the proposed MODE with the standard cross-entropy loss function (3), PatMat (7), and DeepTopPush (DTP) (9). As mentioned in Section II, both PatMat and DTP use the logistic surrogate loss function (4). Since we only have 2k testing images, we only target FP rates $\alpha_0 \in \{10^{-3}, 10^{-2}\}$ for PatMat and MODE. This is because we can get quite noisy results for smaller α_0 due to the lack of training/testing data. It is important to point out that we used PatMat with a linear approximation of the cdf, similarly as for MODE, however, the minibatches are not augmented the same way. In practice, we use the threshold τ_α, instead of t_α visualized in blue in Figure 1. Without this approximation, PatMat would behave the same for the two testing FP rates based on the discussion in Section III-A.

The specific values of true positives for given FP rates, as well as P_E are listed in Tables I,II,III, and IV. For uncompressed images embedded with HILL (Table I), and UERD at QF 100 (Table IV), MODE provides the overall best performance with $2-5\%$ improvement over PatMat. Table II shows that for UERD at QF 75, all methods outperform cross-entropy for $\alpha = 10^{-3}$ for at least 6%, while PatMat being overall best by $1-2\%$ over MODE. For QF 95 (Table IV), PatMat and MODE perform very similarly for the smaller FP rate, while MODE gets a 3% boost for $\alpha = 10^{-2}$.

The full ROC curves for $\alpha_0 = 10^{-3}$ are shown in Figure 2 and we can see that DeepTopPush is indeed more optimized for very low FP rates compared to the other methods, with an exception at QF 95, although these values are extremely noisy due to the testing set size of 2k images. A larger-scale experiment will have to be performed in the future to further validate our finding for smaller FP rates.

To summarize, the cross-entropy loss provides the best results on P_E, which for the given problems is quite close to the FP rate at $\alpha = 10^{-1}$. However, the proposed method improves the detection over the cross-entropy at $\alpha = 10^{-3}$ by $10\%, 6\%, 3\%$, and 14%, for HILL and the 3 quality factors with UERD, respectively. For images compressed with QF 75, PatMat provides the best results, while for higher-quality JPEGs and uncompressed images, MODE outperforms all the other methods.

The code used for implementing MODE will be made available on the authors' website upon acceptance of the paper.

V. Conclusions

We have proposed MODE, a new method of optimizing deep steganalyzers for small false positive rates. We add two

Loss	α_0	P_{E}	$P_{\mathrm{D}}(\alpha)$		
			10^{-3}	10^{-2}	10^{-1}
CE	-	$\mathbf{0.2280}$	0.1535	0.2680	$\mathbf{0.5990}$
DTP [13]	0	0.2760	0.1315	0.2745	0.5265
Pat&Mat [1]	10^{-2}	0.2357	0.1555	0.3105	0.5860
	10^{-3}	0.2390	$\mathbf{0.1830}$	0.3020	0.5970
MODE	10^{-2}	0.2395	0.1760	$\mathbf{0.3375}$	0.5955
	10^{-3}	0.2402	0.1815	0.3320	0.5825

TABLE III: Results for various FP rates. UERD, QF 95.

Loss	α_0	P_{E}	$P_{\mathrm{D}}(\alpha)$		
			10^{-3}	10^{-2}	10^{-1}
CE	-	$\mathbf{0.1520}$	0.2250	0.5440	$\mathbf{0.7830}$
DTP [13]	0	0.1710	0.3595	0.5600	0.7445
Pat&Mat [1]	10^{-2}	0.1593	0.3090	0.5690	0.7720
	10^{-3}	0.1607	0.3180	0.5590	0.7730
MODE	10^{-2}	0.1578	0.3295	0.5710	0.7815
	10^{-3}	0.1618	$\mathbf{0.3695}$	$\mathbf{0.5830}$	0.7715

TABLE IV: Results for various FP rates. UERD, QF 100.

top cover samples to each minibatch during the detector's training. The logits from these two samples are used to estimate the right tail of the cover distribution and find a decision threshold given by a desired FP rate. Using a surrogate logistic function, this decision threshold is then used to optimize a shifted cross-entropy loss function, which puts emphasis on the desired FP rate.

We demonstrate on several JPEG quality factors, and on uncompressed images that if the problem requires FP rates close to 0, it seems that using DeepTopPush to optimize the detector is the steganalyst's best choice. However, as soon as we can allow small FP rates, MODE leads to better true positive rates outperforming, although by a small margin, other state-of-the-art methods for deep learning steganalysis. Due to a potential computational overhead on a large dataset, we only considered targeting FP rates of 10^{-3} and 10^{-2}.

Since the experiments were performed on a rather small BOSSBase dataset, we plan to evaluate the method further with more data in order to validate the method for small FPs such as 10^{-5}. Furthermore, the effect of batch size, and the effect of using top $K > 2$ cover samples on the estimate of the quantile will also be studied. Finally, non-linear approximations of the empirical cdf will be investigated.

Our work contributes to the ongoing discussion surrounding steganalysis as a security evaluation metric, highlighting the need for more nuanced approaches that prioritize accuracy over simplicity in real-world forensic scenarios. By exploring novel optimization techniques and empirical evaluations of DL models, we aim to provide actionable insights for researchers seeking to improve their understanding of this complex field.

VI. Acknowledgements

This work received funding from the French Defense & Innovation Agency. This work was also supported by a French government grant managed by the Agence National de la Recherche under the France 2030 program, reference ANR-22-PECY-0011.

References

[1] Lukás Adam, Václav Mácha, Václav Šmídl, and Tomás Pevnỳ. General framework for binary classification on top samples. *Optimization Methods and Software*, 37(5):1636–1667, 2022.

[2] 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.

[3] 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.

[4] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, June 21–25, 2021.

[5] R. Cogranne, Q. Giboulot, and P. Bas. The ALASKA steganalysis challenge: A first step towards steganalysis "Into the wild". 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.

[6] R. Cogranne, Q. Giboulot, and P. Bas. ALASKA–2: Challenging academic research on steganalysis with realistic images. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[7] M.A. Davenport, R.G. Baraniuk, and C.D. Scott. Controlling false alarms with support vector machines. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, 2006.

[8] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.

[9] 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.

[10] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi. Using statistical image model for JPEG steganography: Uniform embedding revisited. *IEEE Transactions on Information Forensics and Security*, 10(12):2669–2680, 2015.

[11] V. Holub and J. Fridrich. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security*, 10(2):219–228, February 2015.

[12] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[13] Václav Mácha, Lukáš Adam, and Václav Šmídl. Deeptoppush: Simple and scalable method for accuracy at the top. *arXiv preprint arXiv:2006.12293*, 2020.

[14] J. Neyman and P. E. Sharpe. On the problem of the most efficient tests of statistical hypotheses. In *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 1933.

[15] T. Pevný and A. D. Ker. Towards dependable steganalysis. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 1501–1514, San Francisco, CA, February 8–12, 2015.

[16] Philippe Rigollet and Xin Tong. Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, 12(86):2831–2855, 2011.

[17] 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.

[18] 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.

[19] 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.

[20] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich. Imagenet pretrained cnns for jpeg steganalysis. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.