Adversarial Examples Against Deep Neural Network based Steganalysis

Yiwei Zhang
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
zywvvd@mail.ustc.edu.cn

Weiming Zhang*
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
zhangwm@ustc.edu.cn

Kejiang Chen
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
chenkj@mail.ustc.edu.cn

Jiayang Liu
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
1229370169@qq.com

Yujia Liu
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
yjcaihon@mail.ustc.edu.cn

Nenghai Yu
CAS Key Laboratory of Electromagnetic Space Information
University of Science and Technology of China
Hefei, Anhui, China
ynh@ustc.edu.cn

ABSTRACT

Deep neural network based steganalysis has developed rapidly in recent years, which poses a challenge to the security of steganography. However, there is no steganography method that can effectively resist the neural networks for steganalysis at present. In this paper, we propose a new strategy that constructs enhanced covers against neural networks with the technique of adversarial examples. The enhanced covers and their corresponding stegos are most likely to be judged as covers by the networks. Besides, we use both deep neural network based steganalysis and high-dimensional feature classifiers to evaluate the performance of steganography and propose a new comprehensive security criterion. We also make a tradeoff between the two analysis systems and improve the comprehensive security. The effectiveness of the proposed scheme is verified with the evidence obtained from the experiments on the BOSSbase using the steganography algorithm of WOW and popular steganalyzers with rich models and three state-of-the-art neural networks.

KEYWORDS

Steganography; adversarial examples; deep neural network; steganalysis; security

*Corresponding author

IH&MMSec'18, June 20–22, 2018, Innsbruck, Austria

ACM ISBN 978-1-4503-5625-1/18/06…$15.00
https://doi.org/10.1145/3206004.3206012

ACM Reference Format:
Yiwei Zhang, Weiming Zhang, Kejiang Chen, Jiayang Liu, Yujia Liu, and Nenghai Yu. 2018. Adversarial Examples Against Deep Neural Network based Steganalysis. In *Proceedings of 6th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'18).* ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3206004.3206012

1 INTRODUCTION

In recent years, information hiding researchers have proposed many advanced steganographic algorithms to hide secret information into a cover image. Most of the schemes embed secret messages in spatial domain or frequency domain, such as HUGO [16], WOW [7], S-UNIWARD [8], HILL [14], J-UNIWARAD [8] and UERD [6]. These methods can minimize a heuristically-defined embedding distortion while hiding secrets into a given image to lower the statistical detectability. And based on an oracle used to calculate the detectability map, a new steganography called ASO [13] is proposed which can preserve both cover image and sender's database distributions during the embedding process.

In order to detect whether there is hidden information in an image, the traditional method of steganalysis is divided into two steps, high-dimensional feature extraction and machine learning classifier training. An excellent steganalyzer is the Rich Model (RM), which is usually used in the first step. There are several versions of Rich Models such as Spacial Rich Model (SRM) [4] and its variants [3, 19] in spatial domain and JPEG-SRM (J-SRM) [10] in frequency domain. The most common choice of machine learning classifier is Ensemble Classifier (EC) [11]. The combination of SRM and EC has achieved excellent detection performance.

In the past two years, steganalysis based on Convolutional Neural Network (CNN) models has made a tremendous progress. Compared with the traditional methods, CNN-based steganalysis uses various network structures to learn the effective features of images to distinguish cover images and stego images. Qian [17] used a CNN architecture with Gaussian activations function to construct

a model for steganalysis. Xu [23] designed another CNN structure with *tanh* activation function. Wu [20–22] proposed a CNN model made full use of the advantage of residual network for image steganalysis. Ye [24] proposed a CNN model whose first layer was initialled with the high-pass filter set used in SRM and introduced a novel activation function TLU and a selection-channel-aware scheme for CNN-based steganalysis. The performances of Ye and Wu exceed that of SRM+EC.

Therefore, deep learning steganalysis has become a severe challenge to steganography. In fact, the tasks of neural networks for steganalysis and networks for object classification are very similar. The difference between them are the structure of networks and the number of classification targets. Steganalysis is a binary classification problem while object classification has multiple category labels. Adversarial example is a technique that adds carefully crafted small adversarial noise to the input to cheat the object classification network producing incorrect outputs. Szegedy [18] and Goodfellow [5] made a seminal work and put forward a method of adversarial example construction based on neural network gradient. Then a lot of related outstanding works came up [1, 12, 15].

In a steganographic process, let C be the cover image, S be the stego and we use $\boldsymbol{m}$ to denote the secret message. An intuitive way to combine the technique of adversarial examples with steganography is to add adversarial noise $\boldsymbol{n}_{ad}$ to S in order to turn S into an adversarial example $S_{ad} = S + \boldsymbol{n}_{ad}$. We assume that the object network will judge S as cover successfully, but $\boldsymbol{n}_{ad}$ will prevent the receiver from getting the message $\boldsymbol{m}$ correctly. In a word, there is some difficulty in applying technique of adversarial examples directly to steganography.

In this paper, we propose an adversarial example construction method suitable for steganography, that is, reverse the order of adding adversarial noise and embedding messages. We first add the adversarial noise to C to construct a robust enhanced cover C', then embed the message m into C' to get S. In this way, the receiver can extract $\boldsymbol{m}$ from S successfully. Our method can make the adversarial example C' robust enough to withstand the influence of the message embedding process, so S will still be misjudged by the deep learning classifier as cover. We have also considered how to generate adversarial examples against multiple neural networks for steganalysis. However, in the process of constructing the adversarial examples, modification would be introduced to the images unavoidably. Actually, the larger modification would be likely exposed to SRM+EC. We have analyzed how to control the noise intensity to obtain a reasonable trade-off, so that the overall security of steganography is improved.

The rest of this paper is structured as follows: In Section 2 we review a method to construct adversarial examples and describe details of our adversarial training methods. Section 3 describes our experiment settings and results. Conclusions are drawn in Section 4.

2 STEGANOGRAPHY BASED ON ADVERSARIAL EXAMPLES

In this section, we introduce the technique of adversarial examples and describe our method of training robust adversarial cover images for a given network. We also illustrate ways to construct adversarial examples for multiple networks.

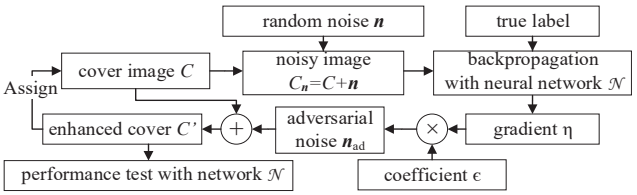

Figure 1: Diagram of the process of constructing enhanced covers using the technique of adversarial examples.

2.1 Adversarial Examples

Similar to the classic "fast gradient sign method"[5], our method to construct an adversarial example is as follows. For a given neural network $\mathcal{N}$, we use θ to denote its parameters. Let C be the input image, y be the target label associated with C (when we are constructing adversarial examples against neural network for steganalysis, the target would always be set to 0 which means cover, while 1 presents stego) and $L_{\mathcal{N}}(\theta, C, y)$ be the loss of the network. Fix θ and compute the gradient of $L_{\mathcal{N}}$ as η:

$$\eta = \nabla_{\boldsymbol{C}} L_{\mathcal{N}}(\theta, C, y), \tag{1}$$

A perturbation that coincides with η will make C more easily be judged as cover by $\mathcal{N}$. Note that the η could be computed efficiently by back propagation in the network. We can simply multiply the gradient η by a coefficient ϵ and add it to C to get the adversarial example C':

$$C' = C + \epsilon\eta. \tag{2}$$

ϵ is related to the learning rate of deep learning. By choosing appropriate ϵ, C' can mislead $\mathcal{N}$ successfully. The above are the construction procedures of adversarial examples. However, just misleading the networks is far from enough. What we need are robust enhanced covers that can withstand the message embedding processing in steganography. We will discuss how to construct the enhanced covers in Section 2.2.

2.2 Adversarial Examples Against Neural Network for Steganalysis

In order to let cover images resist steganographic noise, we introduce a model which can iteratively construct enhanced covers. The model is shown in Figure 1. The process starts at cover image C. We define a noise vector, $\boldsymbol{n}$, which has a considerable strength relative to the steganographic noise that we are going to resist. We add noise $\boldsymbol{n}$ to C to get $C_{\boldsymbol{n}}$. Then a gradient η is obtained by back propagation of network $\mathcal{N}$. Multiply η by the coefficient ϵ to obtain the adversarial noise $\boldsymbol{n}_{ad}$. The sum of $\boldsymbol{n}_{ad}$ and C is the enhanced cover C'. Performance test on C' with $\mathcal{N}$ is the last step in this iteration. If C' passes the test at this time, it will be used as an enhanced cover for steganography, otherwise we will assign C' to C and start the next iteration.

2.2.1 Construction Process. In order to construct C' successfully and ensure the robustness, we usually need to go through several loops to construct an enhanced cover. We set the number of required training loops to q. To ensure the stability and strength of $\boldsymbol{n}_{ad}$, there would be k iterations in each loop. Given a cover image C (C_0) and

a neural network $\mathcal{N}$ with parameters θ in the i-th ($1 \le i \le k$) iteration of a loop, we simulate the steganography with random noise $\boldsymbol{n}_i$ on C_{i-1} (after $i-1$ rounds of iterative calculation of C) to get the noisy image $C_{\boldsymbol{n}_i}$ and then feed it to $\mathcal{N}$. After that, we can use the method described in Section 2.1 to calculate the i-th gradient η_i. Here we add the adversarial noise $\boldsymbol{n}_{ad_i} = \epsilon\eta_i$ calculated on $C_{\boldsymbol{n}_i}$ to C_{i-1} to complete the i-th update iteration, which means $C_i = C_{i-1} + \boldsymbol{n}_{ad_i}$. The modification vector Δ obtained from k iterations can be expressed as:

$$\Delta = \sum_{i=1}^{k} (\epsilon \nabla_{C+\Delta_i} L_{\mathcal{N}}(\theta, C + \Delta_i + \boldsymbol{n}_i, y)), \tag{3}$$

where Δ_i is the cumulative modification vector after i iterations of training, that is $\Delta_i = \sum_{j=1}^{i} \boldsymbol{n}_{ad_j}$. As i increases, $C + \Delta_i$ will gradually adapt to the noise interference and η_i will convergence to a vector close to zero.

After k iterations, the pixel values would be rounded to integers and bounded to 0 to 255 so that the pixels are saved as integers and the overflows/underflows caused by changes are avoided. Then we will test the performance of C' (C_k). The testing process is as follows. Using the specific steganographic method and relative payload, we embed v group random messages on C' to obtain v stego images $\{S_1, S_2, ..., S_v\}$. When the probability that S_i ($1 \le i \le v$) is judged as a cover by $\mathcal{N}$ is greater than that of as a stego, S_i misleads the network $\mathcal{N}$. C' pass the test of $\mathcal{N}$ only if the corresponding stegos can mislead $\mathcal{N}$ with a probability greater than a threshold denote as Th. Otherwise C_k will be assigned to C and begin the next loop of k iterations. In this way, C' will converge to a stable, robust enhanced cover. Finally, the message m is embedded into the enhanced cover C' to generate the stego object S, which will be send to the receiver.

2.2.2 Intensity Control. Using this method, we can implement the white-box attack [9] on current neural networks for steganalysis. Unfortunately, the noise introduced during the adversarial example construction process will be exposed to SRM+EC.

Because the $\mathbf{L}_2$-norm would control the number and magnitude of image pixel modifications and it could be easily utilized by the network to calculate the gradient, we choose the $\mathbf{L}_2$-norm of the modified vector as a regularizer by adding the $\mathbf{L}_2$-loss to the loss function. Let T denote the number of pixels that need to be modified. We want to control the modification of the image when $\mathbf{L}_2$-loss exceeds T, so we use a threshold bounded loss as the regularizer which is denoted as L_2^T. The process of calculating Δ will be changed as follows:

$$\Delta = \sum_{i=1}^{k} (\epsilon \nabla_{C+\Delta_i} (L_{\mathcal{N}}(\theta, C + \Delta_i + \boldsymbol{n}_i, y) + \varepsilon L_2^T)), \tag{4}$$

$$L_2^T = max(\|\Delta_i\|_2 - T, 0), \tag{5}$$

where ε is the coefficient of L_2^T, which is used to control the strength of the regularization. With loss function (4), we can now construct robust adversarial examples for a specific neural network and control the intensity of the modification vectors.

As we already know, adversarial examples can mislead the network to recognize an input image as an object of another target category. Because of the fact that adversarial perturbations are highly aligned with the weight vectors of a model and different models learn the similar functions in the stage of training to perform the same task, adversarial examples in object classification networks have got a generalization across different models [5]. However, steganalysis is a binary classification based on image residual feature extraction. It abandons the semantic information of images, so the generalization of adversarial examples between different models is not strong. This conclusion has been confirmed in our follow-up experiments. Then, how to get a cover image against multiple neural networks is what we are going to solve in the next subsection.

2.2.3 Against Multiple Neural Networks. In this subsection, we focus on the problem of construction of adversarial examples against multiple neural networks. The main framework still follows the model shown in Figure 1. The differences lie in the method of gradient calculation which is shown in Figure 2 and the part of performance test.

For given h neural networks $\{\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_h\}$, a stego image need to mislead all h networks to pass the performance test. We connect these h networks to form a joint network with one input and multiple outputs. For the process of gradient calculation, we take the weighted sum of the cross-entropy loss $\{loss_1, loss_2, ..., loss_h\}$ of each network as the total loss of all networks. Because we do not know the adaptability of the current image to each network, the initial values of weights $\{\alpha_1, \alpha_2, ..., \alpha_h\}$ are set to 1 and they will be updated in the construction process. If the performance test of i-th ($1 \le i \le q$) loop fails, the weights of failed networks will be updated. For the failed network $\mathcal{N}_t (1 \le t \le h)$, the average probability that stegos are judged as cover by network $\mathcal{N}_t$ after i training loops is represented as $p_{t,i}$. Then we use $\gamma_{t,i} = 1 - p_{t,i}$ as the update step for α_t which means that the loss weight of $\mathcal{N}_t$ would be updated to $\alpha_t + \gamma_{t,i}$.

Similarly, we can control the intensity of adversarial perturbations against multiple networks by adding L_2^T to the total loss of networks. Now the loss function of multiple networks is shown as Equation (6):

$$TotalLoss = \alpha_1 loss_1 + \alpha_2 loss_2 + \cdots + \alpha_h loss_h + \varepsilon L_2^T. \tag{6}$$

With $TotalLoss$ we can calculate C's modification vector Δ to construct an adversarial example against multiple networks.

$$\Delta = \sum_{i=1}^{k} (\epsilon \nabla_{C+\Delta_i} TotalLoss), \tag{7}$$

3 EXPERIMENTS

In this section, we will validate the validity of the proposed model. The image dataset used for all experiments is the BOSSbase ver.1.01 [2]. The BOSSbase contains 10,000 images with the size of 512×512 and is a standard database for evaluating steganography and steganalysis. Taking the speed of operation and the amount of data into account, we cut each image of the dataset into four non-overlapping 256×256 images in our experiment. Therefore, we used a cropped BOSS containing 40,000 images to organize our experiments.

We will get different parameters of a neural network if different steganographic algorithms and relative payloads are utilized. Here

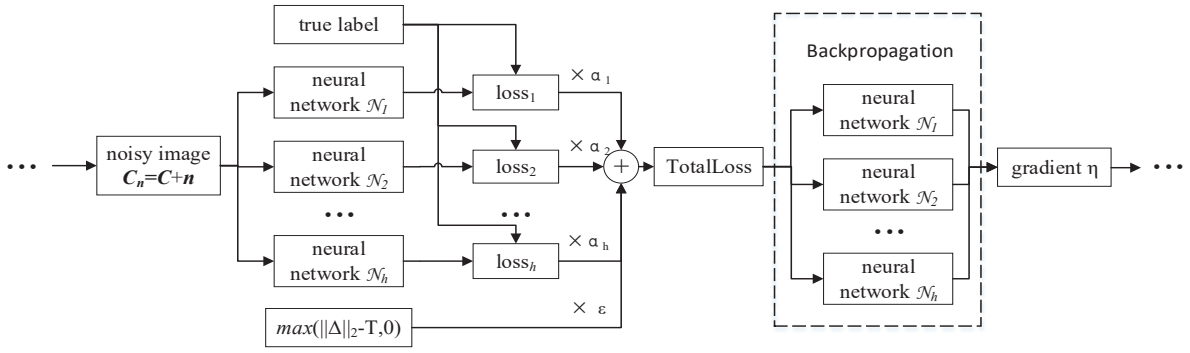

Figure 2: Diagram of gradient calculation for multiple neural network.

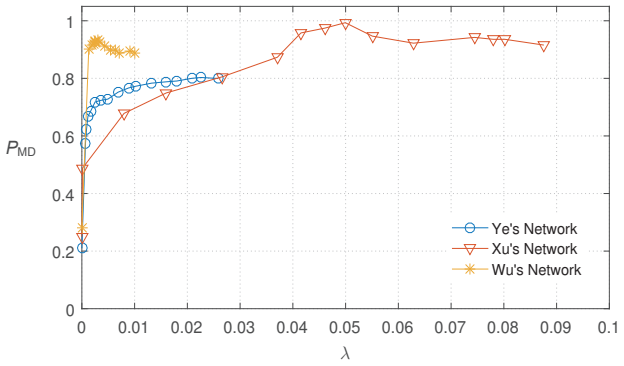

Figure 3: Testing error of adversarial examples by single network under steganography of WOW with a relative payload of 0.4 bpp.

we choose the WOW algorithm under relative payload 0.4 bpp as an example to generate 40,000 stego images. 35,000 randomly selected pairs of cover and stego images are used to train neural networks for steganalysis and SRM+EC, while the other 3000 pairs as validation dataset and the remaining 2000 pairs as testing set. Our experiments use the state-of-the-art neural networks for steganalysis of spacial images, such as Xu's [23], Ye's [24] and Wu's [22] networks. Each network uses an input image size of 256×256, while all other settings are based on the author's source code or description in their essays.

In all experiments, the learning rate is set to 1.0. The number of quantification loops per image, q, is set to 50 and the number of iterations in a loop, k, is 30. The number of testing stego images, v, is set to 300 and the value of Th is 90%. The output loss is calculated with cross-entropy. ε is set to 5×10^{-6}. In order to explore the effect of different modified intensities, we use $\lambda = \frac{T}{256 \times 256}$ as the intensity parameter of the modification and to represent the percentage of image pixel modifications. Because we are trying to generate enhanced secure covers that can accommodate unknown steganographic noise, $\boldsymbol{n}$ is set to a random matrix of -1,0 and 1.

3.1 Against Single Network

For each network, we use 2000 testing images to construct adversarial images for each λ. Then the corresponding network tests the stegos generated by these adversarial images. In most of our experiments, the training images used by networks and SRM+EC are original cover images in BOSSbase and stegos generated by WOW directly. In other words, in most experiments, we didn't use the classifiers retrained with adversarial stegos, and thus our approach does not have an impact on the false alarm rate. Therefore we evaluate the performance only with the probability of missed detection P_{MD}.

Figure 3 shows the testing error of adversarial examples. As λ increases, the performance gets better, but different networks have different effects. For Wu's network, due to the use of the residual network, we only need to change less than 1% (in fact only 0.2%) of the pixels to make an image a satisfactory adversarial example. However, for the Xu's network, we have to introduce a large number of changes to the image to achieve similar results. Modifying 2% of the pixels of an image can make the missing detection probability of Ye's network reach 80%. So when λ is large enough, the constructed images with secret messages embedded are difficult to be detected by specific neural network. This result means that we have successfully constructed adversarial cover images for a single neural network. The average number of iterations of each image is 1.31 and it takes 27.8 seconds on the GPU of NVIDIA Tesla K80.

In order to explore whether there exists generalization in our constructed images, we input images made for a specific network to other networks for steganalysis. The testing result is shown in Figure 4. G in the figure means the networks which construct the image and T means the testing networks. It can be seen is that the generalization of adversarial examples is very limited. So although the images have been modified, they still cannot effectively resist other networks.

3.2 Against Multiple Networks

Defending as many defensive systems as possible is a goal of stenographers. Similarly, constructing an adversarial example that simultaneously resists multiple neural networks for steganalysis is an important part of this paper. In order to verify the correctness of

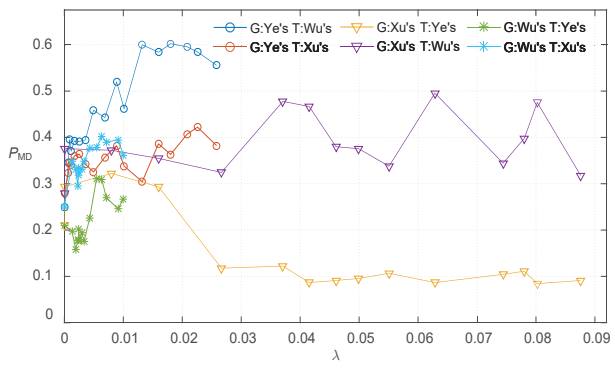

Figure 4: Testing error of adversarial examples by other networks under steganography of WOW with a relative payload of 0.4 bpp.

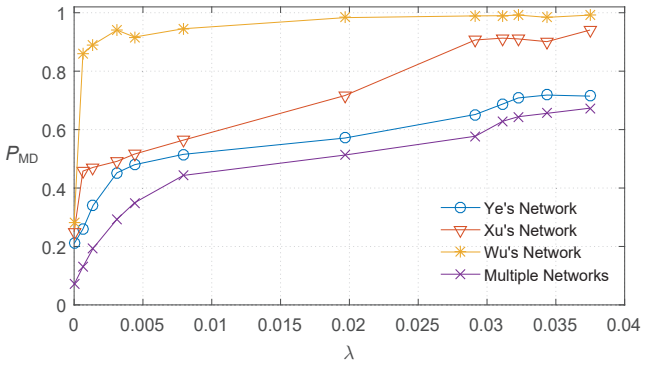

Figure 5: Testing error of multiple networks' adversarial examples under steganography of WOW with a relative payload of 0.4 bpp.

the method described in Section 2.2.3, we use multiple λ values to construct adversarial examples for multiple networks. The validation dataset is used to construct adversarial images for each λ. Each network then detects the images. It is important to emphasize that we use a strict definition of multiple networks testing error rate P_E^{Mul}, which represents the probability that all networks are misled.

The testing error of each network and multiple networks is shown in Figure 5. Multi-network adversarial examples have a very good performance for all networks and as λ increases, P_E^{Mul} can raise from 7.0% to 67.3%. And during the construction of the adversarial examples, the average number of iterations required for each image is 9.93, which will take 583s on a GPU of NVIDIA Tesla K80.

3.3 Tradeoff With SRM+EC

Since SRM+EC is an important and powerful steganalysis system, we have to consider the impact of the construction process of the adversarial images on it. We put adversarial images for multiple networks of different λ into trained SRM+EC. Unfortunately, while

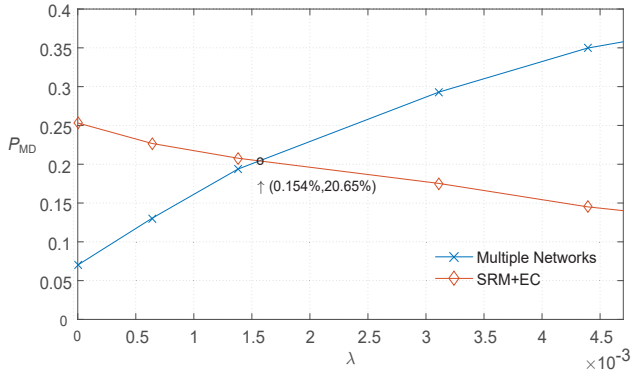

Figure 6: Testing error of multiple networks' adversarial examples by multiple networks and SRM+EC under steganography of WOW with a relative payload of 0.4 bpp.

facing SRM, the safety of the modified cover images gradually decreases as λ increases as is shown in Figure 6. It shows that the images we construct will reduce the performance against steganalysis of high-dimensional features, but we should apply the "Cask Effect Theory" to evaluate the security of steganography algorithms. In other words, if there are several steganalyzers, the security of steganographic algorithm should be defined by the minimum testing error rate. Herein, we define the security of the images as P_E^B, which is the minimum testing error rate of both systems. Our objective is to find a suitable λ that can maximize P_E^B and is stated in Equation (8):

$$\begin{aligned} &\underset{\lambda}{maximize}\ P_E^B, \\ &subject\ to\ 0 \le \lambda \le 1, \end{aligned} \tag{8}$$

where $P_E^B = min(P_E^{SRM}, P_E^{Mul})$. From Figure 6 we can see that, increasing λ from 0% to 0.154% can increase P_E^B from 7.0% to 20.6% which is the highest value that P_E^B can reach and the correctness of the result has been verified on the testing dataset.

In addition to the construction of enhanced secure covers, we can also design a scheme against both neural network and SRM+EC steganalysis by combining the framework of minimizing distortion steganography with the proposed adversarial example technique. For instance, denote the adversarial noise on the ith pixel as $\boldsymbol{n}_i$, and the ±1 distortion on the ith pixel as ρ_i^{+1} and ρ_i^{-1} respectively which is defined with a steganographic algorithm such as WOW or HILL. If $\boldsymbol{n}_i$ is positive, we will reduce the value of ρ_i^{+1} according to the magnitude of $\boldsymbol{n}_i$ and vice versa. Stegos generated from the modified distortion, which is called the "adversarial distortion", will have the ability to resist SRM+EC while misleading neural networks for steganalysis. Some experimental results on the adversarial distortion steganography (ADS) are shown in Table 1, in which the ADS-WOW is constructed with the adversarial distortion on the multi-CNN based steganalyzer. It can be seen that, with adversarial distortion, we can not only increase the ability of WOW to resist CNN based steganalyzers but also increase its ability to resist SRM. We also retrained SRM with stegos generated by ADS, and labeled it as SRM (retrained). As shown in the last line of Table 1, retraining

SRM significantly reduces the P_E on detecting ADS, but suffers from greatly increase of P_E on detecting the original WOW.

Table 1: Testing error of WOW and ADS-WOW under different steganalyzers with a relative payload of 0.4 bpp

Error Rate(%)	WOW			ADS-WOW		
Steganalyzer	P_{MD}	P_{FA}	P_E	P_{MD}	P_{FA}	P_E
Xu's CNN	25.54	26.60	26.07	86.71	26.60	56.66
Ye's CNN	21.34	18.55	19.95	74.26	18.55	46.41
Wu's CNN	28.04	35.17	31.61	75.34	35.17	55.26
Multi-CNN	7.01	58.43	32.72	56.28	58.43	57.36
SRM	25.23	25.77	25.50	47.81	25.77	36.79
SRM (retrained)	23.90	44.72	34.31	13.33	44.72	29.03

4 CONCLUSIONS

In this paper, we propose a method of iteratively constructing robust enhanced cover images that can resist the neural networks for steganalysis and the intensity of adversarial noise is controllable. The stegos, obtained by using the constructed images as cover, can effectively avoid the detection of network-based steganalyzers. Besides, we also consider how to simultaneously fight against network-based steganalyzers and SRM+EC and define the comprehensive security criterion P_E^B under the two systems. We have made a tradeoff between the two systems and evaluated the performance of our model using the BOSSbase dataset, the WOW steganography method and three state-of-the-art networks. Results show the effectiveness of our method and comprehensive security level has been improved.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and 61572452. The authors would like to thank DDE Laboratory of SUNY Binghamton for sharing the source code of steganography, steganalysis and ensemble classifier on the webpage (http://dde.binghamton.edu/download/).

REFERENCES

[1] Shumeet Baluja and Ian Fischer. 2017. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. *arXiv preprint arXiv:1703.09387* (2017).
[2] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. 2011. âĂİ Break Our Steganographic SystemâĂİ: The Ins and Outs of Organizing BOSS. In *Information Hiding*. Springer, 59–70.
[3] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. 2014. Selection-channel-aware rich model for steganalysis of digital images. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*. IEEE, 48–53.
[4] Jessica Fridrich and Jan Kodovsky. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 868–882.
[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[6] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi. 2015. Using statistical image model for JPEG steganography: uniform embedding revisited. *IEEE Transactions on Information Forensics and Security* 10, 12 (2015), 2669–2680.
[7] Vojtech Holub and Jessica Fridrich. 2012. Designing steganographic distortion using directional filters. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*. IEEE, 234–239.
[8] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1.
[9] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* (2017).
[10] Jan Kodovskỳ and Jessica Fridrich. 2012. Steganalysis of JPEG images using rich models. In *Media Watermarking, Security, and Forensics 2012*, Vol. 8303. International Society for Optics and Photonics, 83030A.
[11] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. 2012. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security* 7, 2 (2012), 432–444.
[12] Jernej Kos, Ian Fischer, and Dawn Song. 2017. Adversarial examples for generative models. *arXiv preprint arXiv:1702.06832* (2017).
[13] Sarra Kouider, Marc Chaumont, and William Puech. 2013. Adaptive steganography by oracle (ASO). In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE, 1–6.
[14] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2014. A new cost function for spatial image steganography. In *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 4206–4210.
[15] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. Safetynet: Detecting and rejecting adversarial examples robustly. *arXiv preprint arXiv:1704.00103* (2017).
[16] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*. Springer, 161–177.
[17] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2015. Deep learning for steganalysis via convolutional neural networks. *Media Watermarking, Security, and Forensics* 9409 (2015), 94090J–94090J.
[18] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[19] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang. 2014. Adaptive steganalysis against WOW embedding algorithm. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*. ACM, 91–96.
[20] Songtao Wu, Shenghua Zhong, and Yan Liu. 2017. Deep residual learning for image steganalysis. *Multimedia Tools and Applications* (2017), 1–17.
[21] Songtao Wu, Sheng-Hua Zhong, and Yan Liu. 2016. Steganalysis via Deep Residual Network. In *Parallel and Distributed Systems (ICPADS), 2016 IEEE 22nd International Conference on*. IEEE, 1233–1236.
[22] Songtao Wu, Sheng-hua Zhong, and Yan Liu. 2017. Residual convolution network based steganalysis with adaptive content suppression. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 241–246.
[23] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. 2016. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters* 23, 5 (2016), 708–712.
[24] Jian Ye, Jiangqun Ni, and Yang Yi. 2017. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2545–2557.