Signal Processing 181 (2021) 107920

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Adversarial batch image steganography against CNN-based pooled steganalysis

Li Li, Weiming Zhang*, Chuan Qin, Kejiang Chen, Wenbo Zhou, Nenghai Yu

University of Science and Technology of China, CAS Key Laboratory of Electro-Magnetic Space Information, Hefei 230026, China

ARTICLE INFO

Article history:
Received 18 August 2020
Revised 10 November 2020
Accepted 30 November 2020
Available online 5 December 2020

Keywords:
Batch steganography
Adversarial attack
Pooled steganalysis
Deep learning

ABSTRACT

The application of adversarial embedding in single image steganography exhibits its advantage in resisting convolutional neural network (CNN)-based steganalysis. As an important technique to move the steganography from the laboratory to the real world, batch steganography is developed based on the single image steganography, which uses a series of images as carriers. Furthermore, existing pooled steganalysis also applied CNN architecture for feature extraction, which aims to detect batch steganography. Therefore, it is reasonable and meaningful to introduce adversarial embedding in batch steganography to resist pooled steganalysis. However, as far as we know, there is no work about adversarial batch steganography. Adversarial batch image steganography should be able to resist pooled steganalysis which takes a group of images as a unit, therefore the loss function of the single image steganalyzer can not be directly used for adversarial embedding. In addition, adversarial embedding should be combined with batch strategy. In this paper, we propose a general framework of adversarial embedding for batch steganography, in which a new loss function is designed and the batch strategy is combined with adversarial embedding. By this framework, we can adapt most adversarial embedding algorithms for single image steganography to batch steganography. To verify the efficiency of the proposed framework, we design an algorithm called ADVersarial Image Merging Steganography (ADV-IMS) based on ADVersarial EMBedding (ADV-EMB), and carry out a series corresponding experiments. Experimental results show the proposed method significantly improves the security performance of batch steganography against pooled steganalysis and keeps a high-security level against single image steganalysis.

1. Introduction

Steganography is a technique used to create a covert communication channel, which hides secret information into multimedia such as text and images without arousing any suspects. In the past decades, digital image steganography is well developed. The most effective steganographic schemes are categorized as content-adaptive steganography, which usually consists of a heuristically defined distortion function and a method for encoding the message to minimize the total distortion [1]. Based on this framework, the near-optimal Syndrome-Trellis Codes (STC) [2] is developed for encoding, and various distortion functions [3–5] are devised. Nowadays, many researchers have attempted to introduce deep learning into the field of steganography [6–8,42]. These methods can automatically learn the steganographic strategy without any domain knowledge.

Since the steganographer in the real world has access to more than one object, batch steganography is proposed to move steganography from the laboratory to the real world, which hides secret messages into a group of images [9]. Batch steganography studies how to distribute payload across a group of images based on the distortion definition and STC embedding of single image steganography. In [10], Ker et.al proposed five strategies for non-adaptive steganography algorithms, i.e., even, max-greedy, max-random, linear, sqroot. In the **even** strategy, the message is distributed evenly into all available covers regardless to their capacity. In the **max-greedy** strategy, the steganographer wants to embed the message into the fewest possible number of covers, thus he iteratively chooses the covers with highest capacity yet to be used, and embeds a portion of the message equal to the capacity of the image. The **max-random** strategy is the same as max-greedy, except that the covers used for embedding are chosen in a random order. In the **linear** strategy, the message is distributed into all available covers proportionately to their capacity. In the **sqroot** strategy, the message is spread among all images with the length of the fragments being proportional to the square root of their ca-

* Corresponding author.
E-mail addresses: zhangwm@ustc.edu.cn (W. Zhang), chenkj@mail.ustc.edu.cn (K. Chen).

https://doi.org/10.1016/j.sigpro.2020.107920
0165-1684/

pacities. Furthermore, some works [11–13] investigate the steganographic capacity of images with the greedy strategy as the default strategy. In [14], Cogranne et.al proposed three strategies for adaptive steganography, i.e., Image Merging Sender (**IMS**), Detectability Limited Sender (**DeLS**) and Distortion Limited Sender (**DiLS**). In IMS, the steganographer merges all images into one and lets the embedding algorithm spread the payload. In DeLS and DiLS, each image from the bag contributes with the same value as the KL divergence and distortion, respectively. These strategies move the steganography closer to the real world.

Opposite to steganography, steganalysis aims at revealing the existence of the secrets. Single image steganalysis is taken as a binary classification problem, conventional methods utilize artificial features [15,16] and an ensemble classifier [17], while other state-of-the-art methods are implemented by a deep convolutional neural network (CNN) [18–20]. Besides, pooled steganalysis is usually used to detect batch steganography, most of which leverages unsupervised detection methods along with low-dimensional steganalysis features [21–25]. With the development of the deep neural network-based steganalyzer, CNN architecture is used for feature extraction in pooled steganalysis [26], which significantly improves the performance of pooled steganalysis. As a result, even if the steganographer uses batch strategies, the eavesdropper can easily find her by CNN-based pooled steganalysis.

However, many researches of computer vision show that adding well-designed small noises to the image context will dramatically mislead the image classification network with high confidence, and the well designed noise is called adversarial noise [27,28]. Since single image steganalyzer can be regarded as a binary classifier, many steganography experts combine the adversarial attack with steganography embedding to resist CNN-based steganalyzers. Zhang et al., [29] first proposed a method that generates enhanced covers by iteratively adding adversarial noises to cover image, so that the stegos generated from the enhanced covers are misclassified as covers by the steganalyzer. Li et al., [30] split the cover image into two parts thus separating the embedding perturbations and adversarial noises. Ma et al., [31] modified the pixel bits by ± 1 according to the direction of adversarial noises under the framework of single-layered STC and introduced an unbalanced distortion function for ternary embedding according to the adversarial gradients. Tang et al., [32] proposed the ADVersarial EMBedding (ADV-EMB) method which generates adversarial stego with a minimum amount of adjustable elements and achieved good security performance. These methods demonstrate that the performance of existing steganographic algorithms can be improved by combining steganography with adversarial attack.

Although existing adversarial embedding algorithms work well against single image steganalyzer, they can't be directly applied to adversarial batch steganography. Firstly, adversarial stegos in single image steganography are designed to counter single image steganalyzer which is usually modeled as an end-to-end supervised classifier. However, adversarial batch steganography should be able to resist pooled steganalysis which usually uses unsupervised methods and takes a batch of images as a detection unit. In pooled steganalysis, it should be noted that there is no differentiable end-to-end loss function that is often used in adversarial embedding. Therefore, batch adversarial steganography is a different problem from existing adversarial steganography. Secondly, batch steganography distributes the payload among a batch of images rather than a single image, in addition to the distortion design and STC embedding, payload spreading strategies should also be considered to improve the confidentiality.

To realize adversarial batch steganography countering CNN-based pooled steganalysis, we design a general loss function for pooled steganalysis, and propose a general scheme for adversarial batch steganography which combines batch strategies and adversarial embedding together. To our knowledge, this is the first work of adversarial batch steganography. Our innovations are as follows:

- Proposing a general framework of adversarial batch steganography against pooled steganalysis.
- Designing a loss function for adversarial batch steganography, which is called as MMD-loss.
- Implementing the proposed method based on ADV-EMB algorithm, and analyzing its performance on resisting different pooled steganalysis methods and single image steganalysis.

The rest of this paper is organized as follows. In Section 2, we analyse the difference between adversarial single image steganography and adversarial batch steganography, and give the background knowledge about Maximum Mean Discrepancy (MMD). In Section 3, we propose a general framework for adversarial batch steganography by designing a novel loss function, and detail its implementation based on Adversarial Embedding (ADV-EMB) method. The experiment settings and experimental results are given in Section 4. Finally, in section 5, we conclude our work and look forward to the future work.

2. Preliminary

2.1. Single adversarial steganography (SAS) vs. batch adverasrial steganogarphy (BAS)

As illustrated in Fig. 1, single image steganalysis is usually regarded as a binary classification problem, and usually a supervised machine learning method is applied. Therefore, the objective for adversarial examples is to fool the well trained classifier. Let $\mathcal{F}$ be a deep neural network to be attacked. For an input image $\mathbf{X}$, the last layer of the network $\mathcal{F}$ outputs the predicted probability, which is denoted as $\mathcal{F}(\mathbf{X})$. The output of the last feature layer is taken as the steganalysis feature used in pooled steganalysis, which

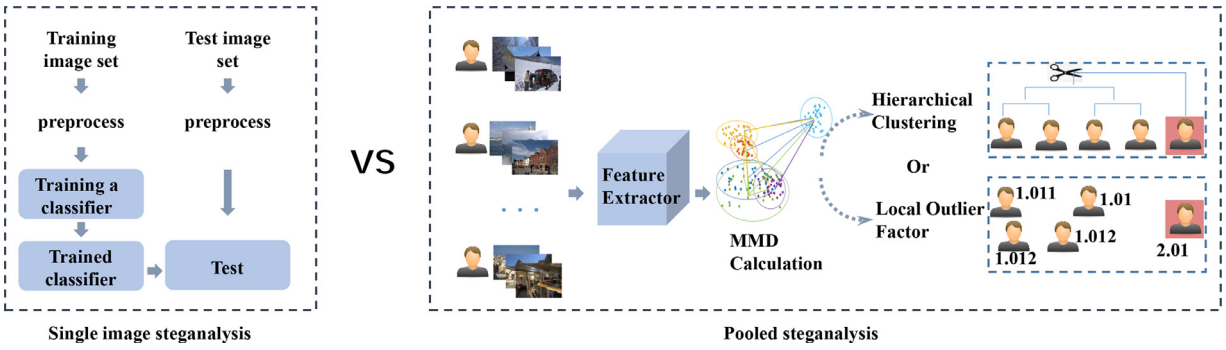


Fig. 1. Single image steganalysis vs. pooled steganalysis.

is denoted as $\mathcal{H}(\mathbf{X})$. For a single image steganalyzer, the input $\mathbf{X}$ is identified as a stego if $\mathcal{F}(\mathbf{X}) > 0.5$, else it is taken as a cover.

Traditional steganographic embedding and extraction procedures are described as Eq. (1),

$$\begin{aligned}&\mathbf{Emb}(\mathbf{X},\mathbf{m}) = \arg\min_{\mathcal{P}(\mathbf{Y})\in\mathcal{C}(\mathbf{m})} D(\mathbf{X},\mathbf{Y})\\&\mathbf{Ext}(\mathbf{Y}) = \mathcal{P}(\mathbf{Y})\mathbb{H}^{\mathbf{T}} = \mathbf{m},\end{aligned} \tag{1}$$

where $D(\mathbf{X},\mathbf{Y})$ is the modification cost when change $\mathbf{X}$ to $\mathbf{Y}$, $\mathcal{P}(\mathbf{Y})$ is a parity function shared between the sender and the receiver (e.g., $P(\mathbf{Y}) = \mathbf{Y} \mod 2$), $\mathbb{H}^{\mathbf{T}} \in \{0,1\}^{n\times m}$ is a parity-check matrix of the binary code $\mathcal{C}(n; n-m)$. $\mathcal{C}(\mathbf{m}) = \{\mathbf{z} \in \{0,1\}^n | \mathbf{z}\mathbb{H}^{\mathbf{T}} = \mathbf{m}\}$ is the coset corresponding to syndrome $\mathbf{m}$. State-of-the-art methods of adversarial embedding in single image steganography adjusts the steganography distortion of different modified direction (+1/1) according to the direction of adversarial noise. With the help of adversarial noise, the secret message is embedded into the cover $\mathbf{C}$ resulting in an adversarial stego $\mathbf{S}^*$, keeping $\mathcal{H}(\mathbf{S}^*) \leq 0.5$ at the same time, and the adversarial noise can be obtained by back propagating the loss function of the steganalyzer.

By contrast, pooled steganalysis takes a group of images as a whole, and utilizes the trained classifier as the feature extractor. Then unsupervised machine learning methods (e.g., hierarchical clustering [33] and local outlier detection [34]) are applied to detect the steganographer, so there is none differentiable loss function can be used to obtain the adversarial noise. Though in some cases, pooled steganalysis pooling the results of single images, the loss function used to train single image steganalyzer can't be directly used to attack pooled steganalysis. Therefore, we design an effective loss function using the average distance between the stegonographer and normal users in feature domain to attack pooled steganalysis from its middle link.

In addition, adversarial embedding in batch steganography embeds secret messages into a group of images $\mathcal{I} = \{\mathbf{I}_i\}$ and generates a group of adversarial stegos $\mathcal{S} = \{\mathbf{S}_i\}$, which aims at finding a solution of S^* that make the detector mistake the stego group S^* as clean. To adapt the adversarial embedding methods in single image steganography to batch steganography, a proper batch strategy to distribute payload among images is also required.

2.2. Maximum mean discrepancy (MMD)

Maximum Mean Discrepancy (MMD) is used to measure the similarity of the distribution between X and Y, which is calculated as Eq. (2),

$$\begin{aligned}&\mathrm{MMD}(\mathbf{X},\mathbf{Y})\\&= \left[\frac{1}{N_1^2}\sum_{i,j=1}^{N_1} K(\mathbf{X}_i,\mathbf{X}_j) - \frac{2}{N_1N_2}\sum_{i,j=1}^{N_1,N_2} K(\mathbf{X}_i,\mathbf{Y}_j) + \frac{1}{N_2^2}\sum_{i,j=1}^{N_2} K(\mathbf{Y}_i,\mathbf{Y}_j)\right]^{\frac{1}{2}},\end{aligned} \tag{2}$$

where N_1/N_2 is the number of samples of $\mathbf{X}/\mathbf{Y}$, $\mathbf{X}_i/\mathbf{Y}_i$ represents samples of $\mathbf{X}/\mathbf{Y}$. It calculates the norm of the difference between two different distributions, which corresponds to an ℓ_2 distance in some Hilbert space implicitly defined through a positive definite kernel function $K(\mathbf{X},\mathbf{Y})$. Radial Basis Function (RBF) kernel is a common used kernel function, which is calculated as Eq. (3), and can be proved as a linear combination of all polynomial kernel functions.

$$\begin{aligned}K(\mathbf{X}_i,\mathbf{Y}_j) &= \exp\left(\frac{-\parallel \mathbf{X}_i - \mathbf{Y}_j \parallel^2}{2\sigma^2}\right)\\&= \exp\left(-\frac{1}{2\sigma^2}\sum_k (\mathbf{X}_{i,k} - \mathbf{Y}_{j,k})\right)^2\end{aligned} \tag{3}$$

where $\mathbf{X}_{i,k}$ and $\mathbf{Y}_{j,k}$ are respectively the k_{th} dimension of sample $\mathbf{X}_i$ and $\mathbf{Y}_j$.

3. Adversarial batch steganography

3.1. Knowledge of the steganographer

We have the assumption that the well-trained feature extraction network in pooled steganalysis is available to the steganographer. Besides, both the steganographer and the eavesdropper have access to some normal social users' data. Though the steganographer has no access to the data gathered by the eavesdropper, she can collect some other normal users' data.

3.2. Motivation

It has been shown that an attacker may significantly poison a clustering process by adding a relatively small percentage of attack samples to the input data, and that some attack samples may be obfuscated to be hidden within some existing clusters [36]. The attack samples can be designed in various ways, including by minimizing the distance among corresponding elements in the target cluster. Besides, by adjusting the stegnographic distortion with the gradient of the loss function of the steganalyzer, the generated adversarial stego can confuse the steganalyzer. Therefore, we define the loss function as the average distance between the steganographer and other normal users.

In single image steganography, the steganalyzer can be misled by adjusting the conventional steganographic distortion according to the gradient map of the loss function of the steganalyzer. In batch steganography, by adjusting the conventional steganographic distortion according to the gradient map of the designed losss function, the steganographer with adversarial stegos is moved closer to other normal users, especially much closer to its neighbors. When the distance gets small enough that as between normal users, our method can attack distance-based steganalysis, such as hierarchical clustering.

In other hand, when the steganographer moves closer to normal users, the distance of the kth closest sample of the steganographer (k-distance) becomes smaller, and so is the reachability between the steganographer and its k-neighbors. the reachability between p and o is described as follows:

$$reach\_dist_k(p,o) = max\{\text{k-distance}(o), d(p,o)\} \tag{4}$$

where $d(p,o)$ represents the distance between p and o. Thus the local reachability density (lrd) gets greater, since

$$lrd(p) = \frac{1}{\frac{\sum_{o\in N_k(p)} reach\_dist_k(p,o)}{|N_k(p)|}} \tag{5}$$

Then, the Local Outlier Factor (LOF) becomes smaller.

$$LOF_k(p) = \frac{\sum_{o\in N_k(p)} \frac{lrd(o)}{lrd(p)}}{|N_k(p)|} = \frac{\sum_{o\in N_k(p)} lrd(o)}{|N_k(p)|}/lrd(p) \tag{6}$$

Fig. 2 demonstrates the difference between the steganographer with adversarial stegos and the steganographer with conventional stegos in feature domain.

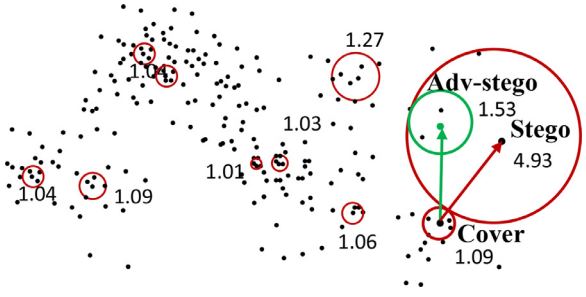


Fig. 2. Illustration of adversarial steganography.

3.3. Proposed framework

We measure the distance between different users by the MMD distance [35] between their feature presentation of images, thus the distance between two actors $\mathcal{X}$ and $\mathcal{Y}$ is represented as $\mathrm{MMD}(H(\mathcal{X}), H(\mathcal{Y}))$, which measures the similarity of the distribution of the two actors' images in feature domain. And our goal is to embed messages to a batch of images and keep the distribution of the stegos as similar as normal users as possible. To embed and extract secret messages effectively, the embedding scheme of STC in steganography is generally used in practice, which can efficiently embed secret messages into images, and extract messages from stegos exactly. The embedding and extraction procedures are described as Eq. (1), and more details can refer to reference [2]. The advantages of utilizing STC is not only it can embed and extract secret messages effectively, but it can also reduce the distance between single cover and stego to some extent by minimizing the embedding distortion, so the distance between steganographer $\mathcal{S}$ and normal user $\mathcal{U}$ can be reduced. Therefore, we apply the steganography embedding scheme of STC to batch steganography. The problem of adversarial attack against pooled steganalysis can be defined as Eq. (7),

$$\begin{aligned} &\arg\min_{\mathcal{S}} \frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}} \mathrm{MMD}(\mathcal{H}(\mathcal{S}), \mathcal{H}(\mathcal{U})) \\ &s.t.\ \mathcal{P}(\mathcal{S})\mathbb{H}^{\mathbf{T}} = \mathbf{m}, \end{aligned} \tag{7}$$

where $\mathcal{W}$ is the normal users' data gathered by the steganographer, and N is the number of users in $\mathcal{W}$.

To solve the problem defined in Eq. (7), we define the loss function as Eq. (8) when the parameters of the network ϕ is given, and call it MMD-loss. $\mathcal{U}$ is a batch of images of the normal user in $\mathcal{W}$, which are gathered by the steganographer, and $\mathcal{A}$ represents the image batch of the steganographer.

$$L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{A}; \phi) = \frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}} \mathrm{MMD}(\mathcal{H}(\mathcal{A}), \mathcal{H}(\mathcal{U})) \tag{8}$$

We apply STC for secrets embedding, and employ EVEN [10] and IMS (Image Merging Sender) [14] strategies for spreading payload among a batch of images. EVEN is a non-adaptive batch strategy, which spread payload evenly in every image, and IMS is one of the state-of-art adaptive batch strategy, which merges the cover images together and then lets existing single image steganography algorithms to distribute the payload. We adopt these two strategies for ablation experiments to valid the effectiveness of adaptive strategy, and to explore how the proposed methods perform on both conditions.

We employ the designed differentiable loss function and the two batch strategies to batch adversarial embedding based on adversarial embedding methods of single image steganography. According to batch strategies, each algorithm can be implemented as two versions, i.e., Adversarial EVEN Steganography **(ADV-EVEN)** and ADVersarial Image Merging Steganography **(ADV-IMS)**, which are detailed as follows.

1. ADV-EVEN evenly distributes payload to every image, and applies adversarial embedding to each image individually, taking Eq. (8) as the loss function to obtain the gradient used in adversarial embedding.
2. ADV-IMS first merges a batch of images into one, and then perform single image adversarial embedding on the merged large image with the merged gradient map of the merged image obtained from Eq. (8) as the loss function.

The proposed general framework of adversarial batch steganography in this section can transplant most adversarial embedding methods(e.g., cover enhancing method [29] and gradient based method [31]) in single image steganography to batch adversarial steganography. The designed framework attacks pooled steganalysis from its middle link rather than the end, which can be seemed as a type of feature attack. Therefore, it can resist most CNN-based pooled steganalysis, including unsupervised methods (e.g., hierarchical clustering [33] and local outlier factor (LOF) [34]) and supervised methods (e.g., count positive methods [9]).

In Section 3.4, we will show the detail implementation of the proposed framework based on the state-of-the-art ADV-EMB [32].

3.4. Practical implementation of adversarial embedding (ADV-EMB) for batch image steganography

In Section 3.3, we propose a general framework for adversarial batch steganography, by which we can adapt existing adversarial embedding methods of single image steganography to batch steganography. In this section, we detail the implementation of ADV-EVEN and ADV-IMS based on the state-of-the-art ADV-EMB [32].

Tang et al. proposed ADV-EMB which generates adversarial stego images with minimum amount of adjustable elements and achieved good performance. In this section, we show how to adapt ADV-EMB to the proposed adversarial batch steganographic scheme (i.e., ADV-EVEN and ADV-IMS) in spatial domain.

Typical additive distortion function for ternary embedding in single image steganography is defined as Eq. (9),

$$D(X, Y) = \sum_{i=1}^{H}\sum_{j=1}^{W}(\rho^{+}_{i,j}\delta(R_{i,j}-1) + \rho^{-}_{i,j}\delta(R_{i,j}+1)), \tag{9}$$

where H and W are respectively the height and width of each image, $R_{i,j} = X_{i,j} - Y_{i,j}$ is the difference between the pixels in the ith row and jth column of cover X and stego Y, $\delta(\cdot)$ is an indication function as Eq. (10),

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{else}, \end{cases} \tag{10}$$

and $\rho^{+}_{i,j}$ and $\rho^{-}_{i,j}$ are respectively the cost of increasing and decreasing $X_{i,j}$ by 1. In most schemes, $\rho^{+}_{i,j} = \rho^{-}_{i,j}$, leading to equal probabilities of increasing or decreasing $X_{i,j}$. However, by asymmetrically updating $\rho^{+}_{i,j}$ and $\rho^{-}_{i,j}$ during embedding, steganography security can be further improved, e.g., the CMD (Clustering Modification Direction) strategy [37,38] and ADV-EMB [32]. In [32], Tang et al. proposed to divide the pixels into two groups, i.e., common group and adjustable group. Firstly embed part of secret messages into common group. Then asymmetrically update $\rho^{+}_{i,j}$ and $\rho^{-}_{i,j}$ of the adjustable group according to the direction of adversarial noise, and embed the remaining secrets into adjustable elements according to the adjusted asymmetrical distortion. The minimum amount of adjustable elements can be found heuristically.

In adversarial batch steganography, we define the update rules as Eqs. (11) and (12), where $\rho^{+}_{k,i,j}$ and $\rho^{-}_{k,i,j}$ are respectively the cost of increasing and decreasing the element of ith row jth column in kth image by 1, and α is a parameter in the range of [0,1], $L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi)$ is calculated as Eq. (13), and $\mathcal{Z}$ represents the image batch of the steganographer whose common group have been embedded with part of secrets.

$$q^{+}_{k,i,j} = \begin{cases} \rho^{+}_{k,i,j}/\alpha, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) > 0 \\ \rho^{+}_{k,i,j}, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) = 0 \\ \rho^{+}_{k,i,j}\cdot\alpha, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) < 0 \end{cases} \tag{11}$$

$$q^{-}_{k,i,j} = \begin{cases} \rho^{-}_{k,i,j}/\alpha, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) < 0 \\ \rho^{-}_{k,i,j}, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) = 0 \\ \rho^{-}_{k,i,j}\cdot\alpha, & \text{if } -\nabla_{z_{k,i,j}} L_{\mathrm{MMD}}(\mathcal{W}, \mathcal{Z}; \phi) > 0 \end{cases} \tag{12}$$

$$L_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)=\frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}}\mathrm{MMD}(\mathcal{H}(\mathcal{Z}),\mathcal{H}(\mathcal{U})) \tag{13}$$

$L_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)$ is differentiable, and its gradient can be calculated as Eq. (14).

$$\begin{aligned}\nabla_{z_{k,i,j}}L_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)=&\frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}}\nabla_{z_{k,i,j}}\mathrm{MMD}(\mathcal{H}(\mathcal{Z}),\mathcal{H}(\mathcal{U}))\\&\cdot\nabla_{z_{k,i,j}}\mathcal{H}(\mathcal{Z})\cdot\nabla_{z_{k,i,j}}\mathcal{H}(\mathcal{U})\end{aligned} \tag{14}$$

We represent$\mathcal{H}(\mathcal{Z})$ as $\mathbf{X}$, and $\mathcal{H}(\mathcal{U})$ as $\mathbf{Y}$, then we have,

$$\begin{aligned}\nabla_{z_{k,i,j}}\mathrm{MMD}(\mathcal{H}(\mathcal{Z}),\mathcal{H}(\mathcal{U}))&=\nabla_{z_{k,i,j}}\mathrm{MMD}(\mathbf{X},\mathbf{Y})\\&=\Bigg[\frac{1}{N_1^2}\sum_{i,j=1}^{N_1}\nabla_{z_{k,i,j}}K(\mathbf{X}_i,\mathbf{X}_j)\\&\quad-\frac{2}{N_1N_2}\sum_{i,j=1}^{N_1,N_2}\nabla_{z_{k,i,j}}K(\mathbf{X}_{i,j})\\&\quad+\frac{1}{N_2^2}\sum_{i,j=1}^{N_2}\nabla_{z_{k,i,j}}K(\mathbf{Y}_i,\mathbf{Y}_j)\Bigg]^{\frac{1}{2}},\end{aligned} \tag{15}$$

where

$$\nabla_{z_{k,i,j}}K(\mathbf{X}_i,\mathbf{X}_j)=-\frac{1}{\sigma^2}\exp\left(\frac{-\parallel\mathbf{X}_i-\mathbf{X}_j\parallel^2}{2\sigma^2}\right)(\mathbf{X}_{i,j}-\mathbf{X}_{j,j}) \tag{16}$$

Algorithm 1 Adversarial even steganography (ADV-EVEN).

Input: A batch of images $\mathcal{I}=\{\mathbf{I}_1,\ldots,\mathbf{I}_B\}^{H\times W}$, secret message $\mathbf{m}$ of length M
Output: adversarial stego batch $\mathcal{S}^*=\{\mathbf{S}_1^*,\ldots,\mathbf{S}_B^*\}^{H\times W}$

Initialize the parameter $\beta=0$, $\Delta L_{\mathrm{MMD}}=e^{10}$, $L_{\mathrm{MMD}}=e^{10}$;
$\{\mathcal{P}^+=\{\rho_1^+,\ldots,\rho_B^+\},\mathcal{P}^-=\{\rho_1^-,\ldots,\rho_B^-\}\}=ComputeCost(\mathcal{I})$;
while $\Delta L_{\mathrm{MMD}}<0$ **do**
  for $\mathbf{I}_i\in\mathcal{I}$ **do**
    $\{\mathbf{I}_i^{\mathrm{com}},\mathbf{I}_i^{\mathrm{adj}}\}=RandomDivide(\mathbf{I}_i)$;
    $\mathbf{Z}_{\mathrm{c}}^i=EmbedCommon(\mathbf{I}_i,\mathbf{I}_i^{\mathrm{com}},\mathcal{P}^+,\mathcal{P}^-,\frac{M}{B}(1-\beta))$;
  end for
  $G=\{g_1,\ldots,g_B\}=\nabla_{z_{k,i,j}}L_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)$;
  for $\mathbf{I}_i\in\mathcal{I}$ **do**
    $\{q_i^+,q_i^-\}=UpdateCosts(\rho_i^+,\rho_i^-,g_i)$;
    $\mathbf{Z}_i=EmbedAdjustable(\mathbf{Z}_{\mathrm{c}}^i,\mathbf{I}_i^{\mathrm{adj}},q_i^+,q_i^-,\frac{M}{B}\beta)$;
  end for
  $L'_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)=\frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}}\mathrm{MMD}(\mathcal{H}(\mathcal{Z}),\mathcal{H}(\mathcal{U}))$
  Update $\mathcal{S}^*=\mathcal{Z}$;
  Update β by $\beta+\Delta\beta$;
  $\Delta L_{\mathrm{MMD}}=L'_{\mathrm{MMD}}-L_{\mathrm{MMD}}$;
end while
return $\mathcal{S}^*$

The details of ADV-EVEN are described in Algorithm 1. When we want to embed M bits secrets into a batch of cover images $\{\mathbf{I}_1,\ldots,\mathbf{I}_B\}^{H\times W}$, a conventional steganographic cost function (e.g., HILL [4] and SUNIWARD [5]) is used to compute conventional embedding costs, obtaining $\{\rho_1^+,\ldots,\rho_B^+\}$ and $\{\rho_1^-,\ldots,\rho_B^-\}$ (implemented by *ComputeCost*()). For each image, *RandomDivide*() is implemented to randomly divide pixels into two groups, i.e., common group of $H\times W\times(1-\beta)$ pixels and adjustable group of $H\times W\times\beta$ pixels. We first embed $\frac{M}{B}(1-\beta)$ bits secrets into the common group using conventional embedding costs by steganography coding such as STC [2] (implemented by *EmbedCommon*()). The resultant image batch is denoted as $\mathcal{Z}_c=\{\mathbf{Z}_c^k\}^{H\times W}$. Then compute the gradients of the MMD-loss with respect to the input of $\mathcal{Z}_c$, and update the embedding costs of the adjustable elements by Eqs. (11) and (12) (implemented by *UpdateCosts*()). Finally, we run *EmbedAdjustable*() for each image to embed $\frac{M}{B}\beta$ bits into the adjustable elements by using the updated embedding costs and the same coding scheme.

Theoretically, the optimal β for each images in a batch is different from each other, thus a batch of paramters $\boldsymbol{\beta}=\{\beta_1,\ldots,\beta_B\}$ should be determined to minimize the adjustable elements. It is a direct but time-consuming idea to exhaustively search all possible combinations of $\boldsymbol{\beta}$ value. After weighing pros and cons, we decide to share the same parameter in the experiments, i.e., $\beta_1=\beta_2=\ldots=\beta_B=\beta$.

Algorithm 2 Adversarial image merging steganography (ADV-IMS).

Input: A batch of images $\mathcal{I}=\{\mathbf{I}_1,\ldots,\mathbf{I}_B\}^{H\times W}$, secret message $\mathbf{m}$ of length M
Output: adversarial stego batch $\mathcal{S}^*=\{\mathbf{S}_1^*,\ldots,\mathbf{S}_B^*\}^{H\times W}$

Initialize the parameter $\beta=0$, $\Delta L_{\mathrm{MMD}}=e^{10}$, $L_{\mathrm{MMD}}=e^{10}$;
$\{\mathcal{P}^+=\{\rho_1^+,\ldots,\rho_B^+\},\mathcal{P}^-=\{\rho_1^-,\ldots,\rho_B^-\}\}=Compute\_cost(\mathcal{I})$;
while $\Delta L_{\mathrm{MMD}}<0$ **do**
  $\mathbf{I}_{\mathrm{L}}=Merge(\mathcal{I})$;
  $\rho_{\mathrm{L}}^+=Merge(\mathcal{P}^+)$, $\rho_{\mathrm{L}}^-=Merge(\mathcal{P}^-)$;
  $\{\mathbf{I}_{\mathrm{L}}^{\mathrm{com}},\mathbf{I}_{\mathrm{L}}^{\mathrm{adj}}\}=RandomDivide(\mathbf{I}_{\mathrm{L}})$;
  $\mathbf{Z}_{\mathrm{Lc}}=EmbedCommon(\mathbf{I}_{\mathrm{L}},\mathbf{I}_{\mathrm{L}}^{\mathrm{com}},\rho_{\mathrm{L}}^+,\rho_{\mathrm{L}}^-,M(1-\beta))$
  $\mathcal{Z}_c=Reshape(\mathbf{Z}_{\mathrm{Lc}})=\{\mathbf{Z}_{\mathrm{c}}^i\}^{H\times W}$, $i=1,\ldots,B$,
  $G=\{g_1,\ldots,g_B\}=\nabla_{z_{c,k,i,j}}L_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z}_c;\phi)$;
  $\mathbf{g}_{\mathrm{L}}=Merge(\mathcal{G})$
  $\{\mathbf{q}_{\mathrm{L}}^+,\mathbf{q}_{\mathrm{L}}^-\}=UpdateCosts(\rho_{\mathrm{L}}^+,\rho_{\mathrm{L}}^+,\mathbf{g}_{\mathrm{L}})$;
  $\mathbf{Z}_{\mathrm{L}}=EmbedAdjustable(\mathbf{Z}_{\mathrm{lc}},\mathbf{I}_{\mathrm{L}}^{\mathrm{adj}},\mathbf{q}_{\mathrm{L}}^+,\mathbf{q}_{\mathrm{L}}^-,M\beta)$;
  $\mathcal{Z}=Reshape(\mathbf{Z}_{\mathrm{L}})=\{\mathbf{Z}_1,\ldots,\mathbf{Z}_B\}^{H\times W}$,
  $L'_{\mathrm{MMD}}(\mathcal{W},\mathcal{Z};\phi)=\frac{1}{N}\sum_{\mathcal{U}\in\mathcal{W}}\mathrm{MMD}(\mathcal{H}(\mathcal{Z}),\mathcal{H}(\mathcal{U}))$.
  Update $\mathcal{S}^*=\mathcal{Z}$, update β by $\beta+\Delta\beta$.
  $\Delta L_{\mathrm{MMD}}=L'_{\mathrm{MMD}}-L_{\mathrm{MMD}}$.
end while
return $\mathcal{S}^*$

Algorithm 2 shows the detail implementation of ADV-IMS. Conventional steganographic cost function (e.g., HILL [4] and SUNIWARD [5]) is also first used to compute conventional embedding costs, obtaining $\{\rho_1^+,\ldots,\rho_B^+\}$ and $\{\rho_1^-,\ldots,\rho_B^-\}$ (implemented by *ComputeCost*()). Then a group of images $\mathcal{I}=\{\mathbf{I}_1,\ldots,\mathbf{I}_B\}^{H\times W}$ are reshaped into one-dimensional vectors respectively and merged together to obtain $\mathbf{I}_{\mathrm{L}}$ of size $1\times L$ by *Merge*(), where $L=B\times H\times W$. The pixels of the merged image are randomly divided into two groups, i.e., common group of $B\times H\times W\times(1-\beta)$ pixels and adjustable group of $B\times H\times W\times\beta$ pixels, which is implemented by *RandomDivide*()). We first embed $M(1-\beta)$ bit messages into the common group by *EmbedCommon*() using conventional distortion, and the resultant image is represented as $\mathbf{Z}_{\mathbf{lc}}$. Then split $\mathbf{Z}_{\mathbf{lc}}$ into $\mathcal{Z}_c=\{\mathbf{Z}_{\mathrm{c}}^1,\ldots,\mathbf{Z}_{\mathrm{c}}^B\}^{H\times W}$ by *Reshape*(), and compute the gradients of the MMD-loss with respect to the input of $\mathcal{Z}_c$. *UpdateCosts*() updates embedding cost of adjustable group according to Eqs. (11) and (12). Next, the remaining $M\beta$ bit messages are embedded into adjustable group using updated embedding cost (implemented by *EmbedAdjustable*()) obtaining $\mathbf{Z}_{\mathrm{L}}$. Finally reshape $\mathbf{Z}_{\mathrm{L}}$ into B images of original size, i.e., $\{\mathbf{Z}_1,\ldots,\mathbf{Z}_B\}^{H\times W}$.

In order to minimize the number of adjustable elements for both ADV-EVEN and ADV-IMS, we update the parameter β by $\beta'=\beta+\Delta$, where the initial value of β is 0, until the MMD-loss does not decrease any more. The experimental results show that though it is a local optimal solution, it works well.

4. Experiments

We proposed a general framework which can adapt a class of adversarial embedding for single image steganography to batch steganography, and in Section 3, we detail its implementation based on ADV-EMB. In this section, we carry out experiments, the network we used for steganalysis and feature extraction are all SRNet [20]. To evaluate the performance, following experiments are conducted:

i) We evaluate the performance of the proposed methods in the presence of an adversary-unaware detector who trained his feature extractor or single image steganalyzer with conventional stego images, the network structure and the details about training process can refer to [20]. This corresponds to a white-box attack in adversarial examples [39] and it is the most favorable case for the steganographer. Three pooled steganalysis attack are considered, i.e., Hierarchical Clustering [33], Local Outlier Factor (LOF) [34]) and Sign Test [9]. In addition, for local outlier factor, we consider different situations for the steganographer, i.e.,different numbers of actors and different images number of each actor.
ii) It is also a possible case in practice that the eavesdropper utilizes single image steganalyzer to detect stegos generated by batch steganography. So we also evaluated the proposed methods on an adversary-unaware single image steganalyzer, i.e. SRNet steganalyzer [20].
iii) To explore whether the proposed method has strong transferability against other steganalyzers,we conducted experiments by using other advanced methods, i.e., YeNet and artifacial feature based model to perform the same pooled steganalysis and single image steganalysis tasks.
iv) For ADV-IMS, we also evaluate its performance on the presence of an adversary-aware feature extractor which is re-trained with adversarial stego images. This is a challenging case for the steganographer.

4.1. Experiment settings

4.1.1. Image set

Experiments are carried out on the imagesets of BOSS [40] and BOWS [41], both containing 10,000 spatial images. We resize the images to the size of 256×256 using the MATLAB **imresize()** function, and get the original cover imageset with 20,000 images. Then we divide the dataset into four non-overlapped part: (i) 9000 images for training the feature extractor, which is represented as $\mathcal{D}_1$; (ii) 1000 images for generating the normal users' data collected by the steganographer, represented as $\mathcal{D}_2$; (iii) $\mathcal{D}_3$ contains 5000 images used for generating normal actors' images collected by the eavesdropper; (iv) $\mathcal{D}_4$ contains 5000 images for generating steganographer's image batch.

4.1.2. Simulated situation

We assume the situation that there are N_A actors, including one steganographer and $N_A - 1$ normal users, each actor has N_I images. The attacker aims to distinguish the steganographer from other normal users. We simulate normal users and steganographers with images in the dataset in the following ways:

- Randomly sample N_I images from $\mathcal{D}_2$ / $\mathcal{D}_3$ without repetition to simulate a normal user collected by the steganographer / eavesdropper. Then put them back before simulating the next normal user.
- Randomly divide $\mathcal{D}_4$ into $5000/N_I$ groups, each group contains N_I images, representing a steganographer.

4.1.3. Steganographic schemes

We employ even [10] and IMS [14] as batch strategies together with the steganographic distortion defined by HILL [4] and SUNIWARD [5], and the relative payload is set as $\{0.1, 0.2, 0.3, 0.4\}$ pit per pixel (bpp). We compare our method with conventional batch steganography and two state-of-art single image adversarial steganography [30,32]. For convenience and clarity of expression, we represent two state-of-art single image adversarial steganography as ADV-SIG1 and ADV-SIG2 respectively.

4.1.4. Steganalysis and performance evaluation

We consider both pooled steganalysis and single image steganalysis which are both based on SRNet [20].

In single image steganalysis, SRNet [20] is used as steganalyzer. Since the proposed algorithm only operates on stego image which does not affect the false alarm ratio, we mainly use missed detection ratio P_{MD} to measure the performance, which is calculated as

$$P_{\mathrm{MD}} = \frac{FN}{N_{\mathrm{stego}}}, \tag{17}$$

where FN represents the number of stegos that are taken as covers, and N_{stego} is the total number of stegos. Besides that, we also show false alarm ratios and average errors of single image steganalysis results.

In supervised pooled steganalysis, we use SRNet as single image steganalysis and then we pool the results of all the images to make a final decision, here we use Sign Test and more details can be found at reference [9].

In unsupervised pooled steganalysis, we use SRNet to extract steganalysis features. We first train it as a single image steganalyzer using covers on dataset $\mathcal{D}_1$ and corresponding conventional stegos, then remove its last layer and take the remaining network as the feature extractor $\phi_{C,S}$, which outputs a 512-dimensional feature set. Note that $\mathcal{D}_1$ is used for training the SRNet as a single image steganalyzer, and a single image rather than an actor is taken as a unit during training process. When we obtain the trained feature extractor, we can calculate the MMD distance [35] of the each pair of actors in feature domain to measure their similarity. After that, two popular anomaly detection schemes (hierarchical clustering [33] and Local Outlier Factor (LOF) [34]) are applied to discover the steganographer.

To realize hierarchical clustering, we use the MATLAB function **linkage**() to create cluster tree with **Single** as default method, and cut the hierarchical cluster tree at the second layer to divide the data into two classes by MATLAB function **cluster**(). Ideally, for the steganographer detection task, all the innocent users should be clustered as a cluster and the other cluster only consists of the steganographer. We evaluate the proposed scheme by overall identification accuracy rate (AR) as [24], which is presented as the number of correctly detected steganographic actors over the selected total number of steganographic actors, i.e.,

$$\mathrm{AR} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{total}}}, \tag{18}$$

where, N_{correct} is the number of correctly detected steganographer, and N_{total} represents the selected total number of steganographers.

LOF method calculates the value of local outlier factor (LOF) for each actor, which reflects the anomaly degree of the actor, and the details can be found at reference [34]. We rank actors according to their LOF value in descending order and use the Top-5 accuracy as the benchmark to measure the performance.

Besides, we also apply sign test for steganographer detection to measure the perforance of our method under supervised detection.

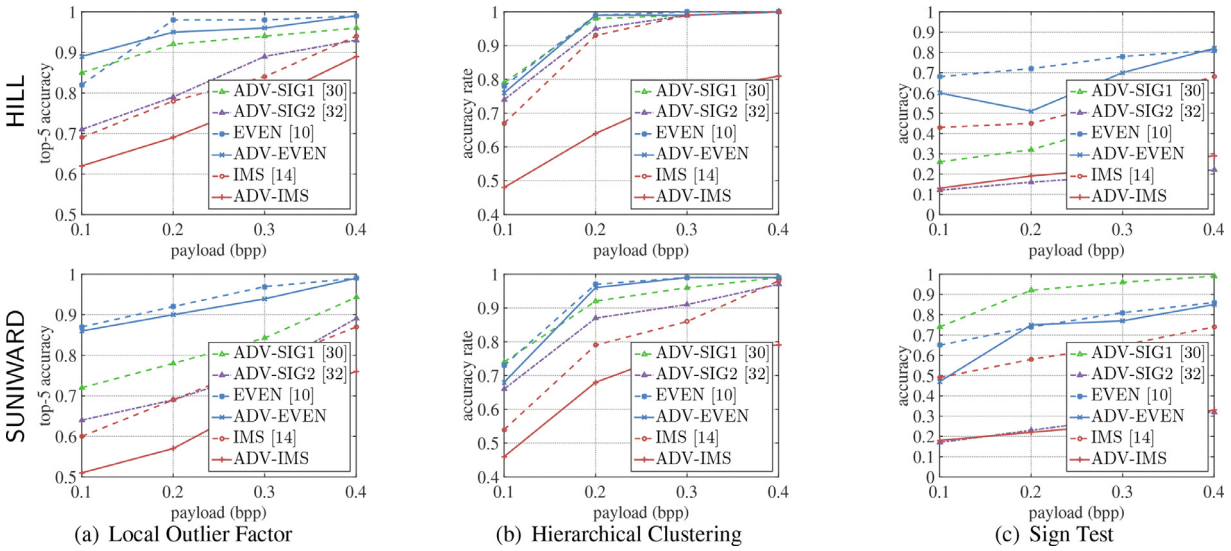


Fig. 3. Performance of the steganographer against different pooled steganalysis.

4.2. Performance against an adversary-unaware pooled steganalysis

In this part, we assume the steganographer is aware of the existence of pooled steganalysis but is unaware of the exactly unsupervised methods the eavesdropper used (LOF or hierarchical cluster). We also assume the steganographer is aware of the knowledge of the feature extractor $\phi_{C,S}$. But the steganalyst is unaware of the adversarial operation and still uses the current feature extractor $\phi_{C,S}$.

To validate the effectiveness of the proposed scheme, we set $N_A = 50$ and $N_I = 50$ in the experiments, and simulate practical situation that the eavesdropper has collected digital images of 49 normal users and a steganographer, and he tries to find out the steganographer among them. For each steganographer generated from $\mathcal{D}_4$, we conducted 20 repeated simulation experiments, and we reselect 49 normal actors from $\mathcal{D}_4$ for each repeat experiment. Fig. 3 shows the performance of the proposed methods against different pooled steganalysis, including Hierarchical Clustering, LOF and Sign Test. It can be seen that the generated adversarial stegos performs well in resisting both supervised and unsupervised pooled steganalysis, and the advantage of ADV-IMS is significant. By adjusting the stegnographic distortion with the gradient of the designed loss function, the steganographer gets closer to other normal users in feature domain, thus the steganographer are hidden within its neighbor cluster, and it can not only interfere the unsupervised pooled steganalysis but also confuse the supervised classifier.

There are two factors contribute to the improvement, i.e., adaptive batch strategy and adversarial embedding, to valid their effectiveness respectively, we carry out a series ablation experiments:

- Removed both the component to obtain the groundtruth, i.e., EVEN.
- Removed the component of adversarial embedding and only leave batch strategy in our method, i.e., IMS.
- Remove the adaptive batch strategy and leave adversarial embedding, i.e., ADV-EVEN.
- Remove none of them, i.e., ADV-IMS

As shown in Fig. 3, ADV-EVEN outperforms traditional EVEN and ADV-IMS outperforms IMS, which indicate the effectiveness of the adversarial embedding methods. By comparing IMS with EVEN, we can see the effectiveness of IMS strategy. It should be noticed that ADV-EVEN performs just a little better than EVEN, while ADV-IMS performs much better than IMS, which indicate that our method is more effective when the batch strategy adaptively distributes payload among images.

To confirm the statistical significance of the improved accuracy, we apply a t-test to evaluate the statistical significance of the proposed algorithms. The hypotheses are

$$H_0 : \mu_1 = \mu_2, H_1 : \mu_1 > \mu_2 \tag{19}$$

where μ_1 and μ_2 are the mean values of detection accuracy of original method (EVEN or IMS) and the improved method (ADV-EVEN or ADV-IMS). H_0 represents that there is no significant difference between them, while H_1 means that the improved accuracy do exists rather than random chance.

The statistic t is calculated as follows:

$$t = \frac{\mu_1 - \mu_2}{S_w\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{20}$$

where

$$S_w = \frac{1}{n_1 + n_2 + 1}[(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2], \tag{21}$$

n_1 and n_2 are the numbers of testing times, and S_1 and S_2 are the standard deviations of the original and improved algorithms, respectively. By looking up the t-score table of the standard normal distribution, the corresponding p-value can be obtained. A lower p-value indicates a lower probability that H_0 holds. If the p-value is less than a threshold, H_0 is rejected, and the improvement is deemed statistically significant and reliable.

The significance level for the test is set to $0.05(t_{0.025}(5) = 2.5706)$. Under different payloads and steganographic schemes, in most cases, the test statistic t values are larger than the corresponding quantile $0.05(t_{0.025}(5)$, which implies the detection improvements have statistical significance.

To further explore the proposed methods, we consider different situation and change the number of actors and batch size, we set $N_A = 10, 50, 100$ and $N_I = 10, 50, 100$ in the experiments, and utilize average rank of the steganographer detected by LOF as security measurement, larger rank value indicates better security performance of the algorithm. The results are shown in Figs. 4 and 5. It demonstrated that though the results are a little sensitive to batch size and actor number, the proposed ADV-IMS method performs best.

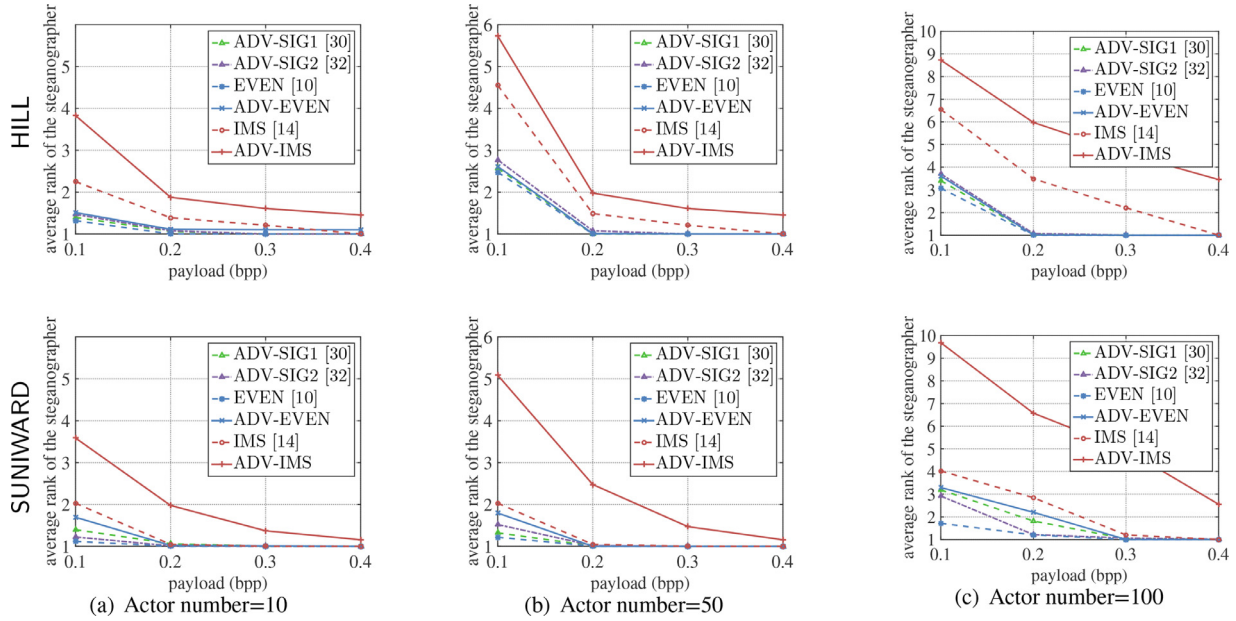


Fig. 4. Performance of the steganographer with different batch size against LOF.

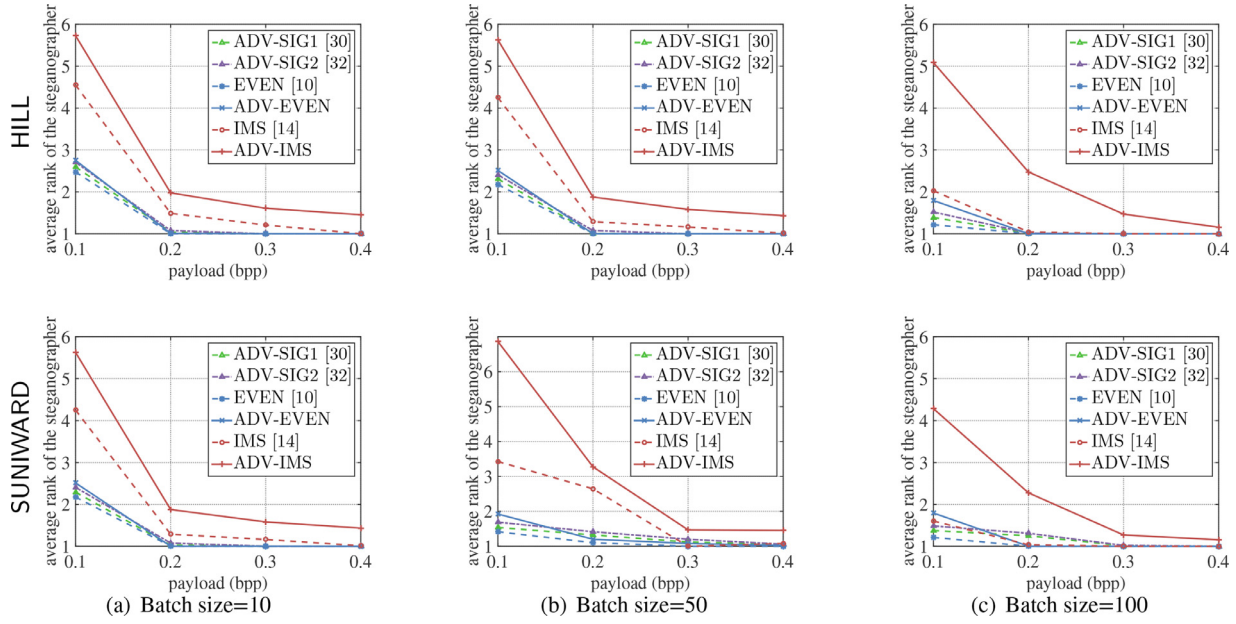


Fig. 5. Performance of the steganographer in the situation of different numbers of actors against LOF.

4.3. Performance against adversary-unaware single image steganalysis

Section 4.2 shows the generated adversarial stegos improve the security of traditional steganography algorithms on resisting pooled steganalysis. But in practice, besides pooled steganalysis, the steganographer also faced with single image steganalysis. Therefore, in this part, we explore the performance of the generated adversarial stegos on resisting single image steganalysis, here we use SRNet[1] as steganalyzer. We assume the steganalyst is unaware of the adversarial operation and still uses the SRNet trained with conventional stegos as steganalyzer even though the steganographer leverages adversarial steganography and batch strategy.

We apply HILL and SUNIWARD as steganographic algorithms to generate stegos at different payloads. Then the ensemble classifier is trained with 10,000 pairs of covers and the stegos at a fixed payload. Tables 1 and 2 show the results of single image steganalysis, the stegos are generated based on HILL distortion and SUNIWARD distortion respectively. The proposed method only operates on stgoes other than covers, it only influence the missed detection ratio. Therefore, the false alarm ratios of different algorithms are the same at the same payload, and we only focus on the missed detection error P_{MD}.

It can be seen that the adversarial stegos generated by ADV-EVEN and ADV-IMS significantly outperform EVEN and IMS respectively. However, our methods perform not as well as ADV-SIG2 when resist steganalyzer of single image, since the proposed batch

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

Table 1
P_{MD} of single image steganalysis using adversarial-unaware SRNet when the teganographer uses HILL [4] distortion.

Batch steganography	Test set	0.1 bpp $P_{FA} = 0.3146 \pm 0.0023$	0.2 bpp $P_{FA} = 0.2239 \pm 0.0018$	0.3 bpp $P_{FA} = 0.1894 \pm 0.0032$	0.4 bpp $P_{FA} = 0.1597 \pm 0.0034$
ADV-SIG1 [30]	$\mathbf{Z}_{\text{ADV-SIG1}}$ from $\mathcal{D}_4$	0.9625 ± 0.0024	0.9417 ± 0.0019	0.9122 ± 0.0026	0.7624 ± 0.0035
ADV-SIG2 [32]	$\mathbf{Z}_{\text{ADV-SIG2}}$ from $\mathcal{D}_4$	0.9925 ± 0.0035	0.9916 ± 0.0050	0.9822 ± 0.0026	0.8224 ± 0.0037
EVEN [10]	$\mathbf{S}_{\text{EVEN}}$ from $\mathcal{D}_4$	0.3721 ± 0.0019	0.2998 ± 0.0028	0.2232 ± 0.0021	0.1851 ± 0.0029
ADV-EVEN	$\mathbf{Z}_{\text{ADV-EVEN}}$ from $\mathcal{D}_4$	**0.4899 ± 0.0035**	**0.4888 ± 0.0029**	**0.2709 ± 0.0019**	**0.2025 ± 0.0031**
IMS [14]	$\mathbf{S}_{\text{IMS}}$ from $\mathcal{D}_4$	0.5956 ± 0.0037	0.5623 ± 0.0025	0.4387 ± 0.0034	0.3216 ± 0.0041
ADV-IMS	$\mathbf{Z}_{\text{ADV-IMS}}$ from $\mathcal{D}_4$	**0.8233 ± 0.0037**	**0.7985 ± 0.0029**	**0.7514 ± 0.0024**	**0.6743 ± 0.0032**

Table 2
P_{MD} of single image steganalysis using adversarial-unaware SRNet when the teganographer uses SUNIWARD [5] distortion.

Batch steganography	Test set	0.1 bpp $P_{FA} = 0.3380 \pm 0.0017$	0.2 bpp $P_{FA} = 0.2318 \pm 0.0036$	0.3 bpp $P_{FA} = 0.1629 \pm 0.0028$	0.4 bpp $P_{FA} = 0.1217 \pm 0.0034$
ADV-SIG1 [30]	$\mathbf{Z}_{\text{ADV-SIG1}}$ from $\mathcal{D}_4$	0.9131 ± 0.0041	0.8829 ± 0.0028	0.8397 ± 0.0033	0.7844 ± 0.0029
ADV-SIG2 [32]	$\mathbf{Z}_{\text{ADV-SIG2}}$ from $\mathcal{D}_4$	0.9725 ± 0.0035	0.9496 ± 0.0028	0.8999 ± 0.0027	0.8346 ± 0.0031
EVEN [10]	$\mathbf{S}_{\text{EVEN}}$ from $\mathcal{D}_4$	0.3521 ± 0.0030	0.2551 ± 0.0032	0.1898 ± 0.0032	0.1649 ± 0.0027
ADV-EVEN	$\mathbf{Z}_{\text{ADV-EVEN}}$ from $\mathcal{D}_4$	**0.5343 ± 0.0021**	**0.2917 ± 0.0032**	**0.2316 ± 0.0035**	**0.1293 ± 0.0031**
IMS [14]	$\mathbf{S}_{\text{IMS}}$ from $\mathcal{D}_4$	0.5145 ± 0.0020	0.4238 ± 0.0029	0.3427 ± 0.0026	0.2319 ± 0.0032
ADV-IMS	$\mathbf{Z}_{\text{ADV-IMS}}$ from $\mathcal{D}_4$	**0.7697 ± 0.0025**	**0.7746 ± 0.0034**	**0.7541 ± 0.0041**	**0.6518 ± 0.0031**

Table 3
Transferability results: detection errors of IMS and ADV-IMS using other advanced methods.

Steganalyzer/Feature extractor	Batch steganography	LOF	Hierarchical clustering	Sign test	Single-steganalysis
SRNet [20]	IMS	0.31	0.32	0.58	0.46
	ADV-IMS	**0.38**	**0.63**	**0.86**	**0.56**
Ye-Net	IMS	0.39	0.37	0.61	0.51
	ADV-IMS	**0.42**	**0.45**	**0.72**	**0.52**
SPAM [15]/SRM [16]	IMS	0.41	0.37	0.63	0.49
	ADV-IMS	**0.43**	**0.42**	**0.69**	**0.48**

adversarial steganography scheme adjusts the embedding cost according to the MMD-loss of features, and it attacks the steganalyzer from its middle link rather than the end. Intrinsically, it sacrifices some targeted performance for more generality. MMD-loss is more generic than the cross entropy loss of the steganalyzer, while cross entropy loss performs better in resisting single image steganalyzer. since the feature extractor is not only a part of pooled steganalysis, but also a part of the steganalyzer, thus the proposed ADV-IMS can resist both single image steganalyzer and pooled steganalysis whereas ADV-SIG can't resist pooled steganalysis (as shown in Section 4.2). Especially for a steganographer with small payload (0.1 bpp) generated by ADV-IMS based on SUNIWARD distortion, the detection accuracy of pooled steganalysis using hierarchical clustering is reduced to 0.46, and the missed detection ratio of single image steganalysis achieved 0.77.

To confirm the statistical significance of the improved accuracy, we also apply a t-test to evaluate the statistical significance of the proposed algorithms. The significance level for the test is also set to $0.05 (t_{0.025}(5) = 2.5706)$, which is usually recommended as a convenient cut off level to reject the null hypothesis, given that it were true. We underline the missed detection error in Tables 1 and 2, where the improvement of the improved method compared to the original algorithm is statistically significant.

4.4. Transferability of adversarial embedding

In order to investigate the case where the adversarial stego images are analyzed by steganalyzers other than the target one, we conducted experiments by using other advanced methods, i.e., YeNet [19] and artifact feature based model to perform the same tasks. Since the low-dimensional features are more suitable for unsupervised pooled steganalysis, we use SPAM [15] feature in LOF and clustering methods, while in sign test and single image steganalysis, we use SRM [16]. The payload of a batch of images is set as 0.1 bpp with the stegnographic distortion defined by HILL. The detection errors are reported in Table 3, showing that ADV-IMS outperforms IMS on resisting different pooled steganalysis methods.

4.5. Performance against an adversary-aware steganalyzer

In this section, we assume that the steganalyzer is aware of the steganographer's adversarial strategy, one of his possible reactions is to re-train the feature extractor with adversarial stego images. Here we only evaluate the performance on resisting LOF detection. We generate adversarial stegos from training set $\mathcal{D}_2$ as described in Algorithm 2 with SUNIWARD distortion, and add them to the training set for training the feature extractor. Then we evaluate performance of the retrained feature extractor of detecting adversarial stego batch of the steganographer which is generated from $\mathcal{D}_4$. In this way, we ensure that the steanographer did not use any prior knowledge of the eavesdropper's image set.

The results are shown in Table 4. The proposed methods performs less effecient on resisting an adversarial-aware steganalyzer. Since the adsversarial-aware steganalyzer is trained not only on

Table 4
Average rank of the steganographer detected by the LOF [34] algorithm. Compared with the adversarial-unaware steganalysis results of ADV-IMS and ADV-SIG, the adversarial-aware steganalyzer decreases the security of ADV-IMS and ADV-SIG. However, either on the adversarial-aware or adversarial-unaware condition, the proposed ADV-IMS method outperforms ADV-SIG.

Batch steganography	0.1 bpp	0.2 bpp	0.3 bpp	0.4 bpp
EVEN	1.02	1.01	1.01	1.00
ADV-EVEN-AW	**1.45**	**1.15**	**1.09**	1.00
IMS	1.79	1.71	1.21	1.17
ADV-IMS-AW	**3.34**	**1.69**	**1.32**	**1.20**

conventional stego images but also on adversarial stego images. However, the adversarial stegos still perform better than conventioanl stegos, which imply that the adversarial stego images disturb steganalyzer in detecting conventional stego images.

5. Conclusion

In this paper, we proposed an adversarial embedding scheme for batch steganography to counter pooled steganalysis, and we designed the ADV-IMS algorithm which significantly improved the steganographic security compared with single image adversarial embedding and conventional steganography. The experimental results verified the efficiency of the proposed method. However, there are still some defects in our method and we would like to improve them in future works. For example, the proposed method performs poorly when faced with adversarial-aware pooled steganalysis. Recently, there are many new works about adversarial embedding in single image steganography, it worths investigating the performance of these approaches when be applied to batch steganography. From the perspective of the eavesdropper, adversarial stegos challenge conventional steganalysis methods. Except for retraining, it should be considered how to detect the steganographer who uses adversarial batch steganography.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

CRediT authorship contribution statement

Li Li: Conceptualization, Methodology, Software, Investigation, Validation, Writing - original draft, Writing - review & editing. **Weiming Zhang:** Conceptualization, Resources, Supervision, Funding acquisition. **Chuan Qin:** Software, Writing - review & editing. **Kejiang Chen:** Writing - review & editing, Project administration. **Wenbo Zhou:** Project administration, Funding acquisition. **Nenghai Yu:** Resources, Funding acquisition.

Acknowledgments

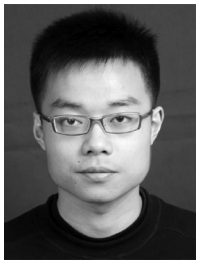
This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and 61572452, Anhui Initiative in Quantum Information Technologies under Grant AHY150400, and by the Anhui Science Foundation of China under Grant 2008085QF296.

References

[1] T. Filler, J. Fridrich, Gibbs construction in steganography, IEEE Trans. Inf. Forensics Secur. 5 (4) (2010) 705–720.
[2] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 920–935.
[3] V. Sedighi, R. Cogranne, J. Fridrich, Content-adaptive steganography by minimizing statistical detectability, IEEE Trans. Inf. Forensics Secur. 11 (2) (2015) 221–234.
[4] B. Li, M. Wang, J. Huang, et al., A new cost function for spatial image steganography, in: Proc. IEEE International Conference on Image Processing (ICIP), 2014, pp. 4206–4210.
[5] V. Holub, J. Fridrich, T. Denemark, Universal distortion function for steganography in an arbitrary domain, EURASIP J. Inf. Secur. 2014 (1) (2014) 1. Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop
[6] J. Hayes, G. Danezis, Generating steganographic images via adversarial training, Adv. Neural Inf. Process. Syst. 30 (2017) 1954–1963.
[7] J. Zhu, R. Kaplan, J. Johnson, F.F. Li, Hidden: Hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 657–672.
[8] J. Yang, D. Ruan, J. Huang, X. Kang, Y.Q. Shi, An embedding cost learning framework using GAN, IEEE Trans. Inf. Forensic Secur. 15 (2020) 839–851.
[9] A.D. Ker, Batch steganography and pooled steganalysis, in: Proc. International Workshop on Information Hiding, 2006, pp. 265–281.
[10] A.D. Ker, T. Pevný, Batch steganography in the real world, Proc. Multimed. Secur. ACM (2012) 1–10.
[11] Z. Zhao, Q. Guan, X. Zhao, et al., Universal embedding strategy for batch adaptive steganography in both spatial and JPEG domain, Multimed. Tools Appl. 77 (11) (2018) 14093–14113.
[12] F. Li, K. Wu, X. Zhang, et al., Robust batch steganography in social networks with non-uniform payload and data decomposition, IEEE Access 6 (2018) 29912–29925.
[13] X. Yu, K. Chen, Y. Wang, et al., Robust adaptive steganography based on generalized dither modulation and expanded embedding domain, Signal Process. 168 (2020) 107343.
[14] R. Cogranne, V. Sedighi, J. Fridrich, Practical strategies for content-adaptive batch steganography and pooled steganalysis, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2122–2126.
[15] T. Pevny, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 215–224.
[16] J. Fridrich, J. Kodovský, Rich models for steganalysis of digital images, IEEE Trans. Inf. Forensics Secur. 7 (2012) 868–882.
[17] J. Kodovský, J. Fridrich, V. Holub, Ensemble classifiers for steganalysis of digital media, IEEE Trans. Inf. Forensics Secur. 7 (2) (2012) 432–444.
[18] G. Xu, H. Wu, Y. Shi, Structural design of convolutional neural networks for steganalysis, IEEE Signal Process. Lett. 23 (5) (2016) 708–712.
[19] J. Ye, J. Ni, Y. Yi, Deep learning hierarchical representations for image steganalysis, IEEE Trans. Inf. Forensics Secur. 12 (11) (2017) 2545–2557.
[20] M. Boroumand, M. Chen, J. Fridrich, Deep residual network for steganalysis of digital images, IEEE Trans. Inf. Forensics Secur. 14 (5) (2018) 1181–1193.
[21] A.D. Ker, T. Pevný, The steganographer is the outlier: realistic large-scale steganalysis, IEEE Trans. Inf. Forensics Secur. 9 (9) (2014) 1424–1435.
[22] A.D. Ker, T. Pevný, A new paradigm for steganalysis via clustering, Proc. Int. Soc. Opt. Photonics (SPIE) 7880 (2011) 78800U01–78800U13.
[23] F. Li, M. Wen, J. Lei, et al., Efficient steganographer detection over social networks with sampling reconstruction, Peer-to-Peer Netw. Appl. 11 (5) (2018) 924–939.
[24] F. Li, M. Wen, J. Lei, et al., Steganalysis over large-scale social networks with high-order joint features and clustering ensembles, IEEE Trans. Inf. Forensics Secur. 11 (2) (2017) 344–357.
[25] A.D. Ker, T. Pevný, The challenges of rich features in universal steganalysis, Proc. Int. Soc. Opt. Photonics (SPIE) 8665 (2013) 86650M.
[26] M. Zheng, S. Zhong, S. Wu, et al., Steganographer detection based on multiclass dilated residual networks, in: Proc. ACM on International Conference on Multimedia Retrieval, 2018, pp. 300–308.
[27] C. Szegedy, W. Zaremba, I. Sutskever, et al., Intriguing properties of neural networks, 2013. ArXiv preprint arXiv:1312.6199.
[28] S. M., A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
[29] Y. Zhang, W. Zhang, K. Chen, et al., Adversarial examples against deep neural network based steganalysis, in: Proc. ACM Workshop on Information Hiding and Multimedia Security, 2018, pp. 67–72.
[30] S. Li, D. Ye, S. Jiang, et al., Attack on deep steganalysis neural networks, in: Proc. International Conference on Cloud Computing and Security, 2018, pp. 265–276.
[31] S. Ma, Q. Guan, X. Zhao, et al., Adaptive spatial steganography based on probability-controlled adversarial examples, 2018. ArXiv preprint arXiv:1804.02691.
[32] W. Tang, B. Li, S. Tan, et al., CNN-based adversarial embedding for image steganography, IEEE Trans. Inf. Forensics Secur. 14 (8) (2019) 2074–2087.
[33] S.C. Johnson, Hierarchical clustering schemes, Psychometrika 32 (3) (1967) 241–254.
[34] M.M. Breunig, H.P. Kriegel, R.T. Ng, et al., LOF: identifying density-based local outliers, Proc. ACM SIGMOD International Conference on Management of Data (2000) 93–104.
[35] A. Gretton, K.M. Borgwardt, M.J. Rasch, et al., A kernel method for the two-sample problem, Proc. Adv. Neural Inf. Process. Syst. (2007) 513–520.
[36] B. Biggio, I. Pillai, B.S. Rota, et al., Is data clustering in adversarial settings secure? in: Proc. ACM Workshop on Artificial Intelligence and Security, 2013, pp. 87–98.
[37] B. Li, M. Wang, X. Li, et al., A strategy of clustering modification directions in spatial image steganography, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1905–1917.
[38] T. Denemark, J. Fridrich, Improving steganographic security by synchronizing the selection channel, in: Proc. ACM Workshop on Information Hiding and Multimedia Security, 2015, pp. 5–14.
[39] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2015, ArXiv preprint arXiv:1412.6572.
[40] P. Bas, T. Filler, T. Pevný, Break our steganographic system": the ins and outs of organizing boss, in: Proc. International Workshop on Information Hiding, 2011, pp. 59–70.
[41] A. Piva, M. Barni, The first bows contest: break our watermarking system, Proc. Int. Soc. Opt. Photonics 6505 (2007) 650516.
[42] H Shi, Dong, J., Wang, W., Qian, Y., & Zhang, X., SSGAN: secure steganography based on generative adversarial networks, Pacific Rim Conference on Multimedia. Springer, Cham (2017) 534–544.

Li Li received her B.S. degree at the School of Communication and Information Engineering, Harbin Engineering University, in 2016. She is currently pursuing a Ph.D. degree in Information Security at the University of Science and Technology of China (USTC). Her research interests include steganography, steganalysis and AI security.

Weiming Zhang received his M.S. degree and Ph.D. degree in 2002 and 2005, respectively, from the Zhengzhou Information Science and Technology Institute, P.R. China. Currently, he is a professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.



Chuan Qin received his B.S. degree in 2016 from Northwest University, Xi'an, China. He is currently pursuing the Ph.D. degree with University of Science and Technology of China. His research interests include steganography, steganalysis and adversarial examples.

Kejiang Chen received the B.S. degree in School of Communication and Information Engineering, Shanghai University, in 2015. He is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). His research interests include information hiding, image processing and deep learning.

Wenbo Zhou received his B.S. degree in 2014 from Nanjing University of Aeronautics and Astronautics, China, and Ph. D degree in 2019 from University of Science and Technology of China, where he is currently postdoctoral researcher. His research interests include information hiding and AI security.

Nenghai Yu received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from the University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding.