Multimedia Tools and Applications
https://doi.org/10.1007/s11042-018-6582-4

Side Channel Steganalysis: When Behavior is Considered in Steganographer Detection

Li Li[1] · Weiming Zhang[1] · Kejiang Chen[1] · Hongyue Zha[1] · Nenghai Yu[1]

Received: 3 June 2018 / Revised: 25 July 2018 / Accepted: 20 August 2018
Published online: 27 August 2018

Abstract
This paper intents to solve the challenging problem of steganographer detection in the real world from a new perspective: side channel attack. We propose utilizing the behavior of actors in the social network to identify the steganographer. While there are many behavior information may expose the steganographer, we just consider the correlation between images sequence as an example in this paper. Base on the assumption that the steganographer choosing images for communication randomly, we design the feature of subtractive images adjacent model (SIAM) to represent the correlations between images sequence of each actor. And then a binary classifier is used to identify the steganographer. To simulate the real world, the images in the experiment are all crawled from twitter. The experimental result shows good performance of our method.

Keywords Steganographer Detection · Side Channel · Behavior · Steganalysis · Social Network

1 Introduction

Steganography aims to communicate by hiding messages in innocent-looking media. The most popular media used recently is digital image because of its convenience to obtain and the developed steganography algorithm. In practice, a complete secret communication through images steganography usually contains four processes: cover selection, steganography algorithm selection, message embedding and sending it to the recipient. The user who sends the stego images is called steganographer in this paper. The communication channel is observed by an adversary or warden who tries to establish whether the communicating parties use steganography, which called steganalysis.

Traditional steganalysis focuses on identifying single object as cover or stego, the problem is usually solved as the problem of binary classification, including feature extraction and

✉ Weiming Zhang
zhangwm@ustc.edu.cn

[1] Key Laboratory of Electro-magnetic Space Information, University of Science and Technology of China, Hefei 230026, China

classification. Designing features that can capture the impact of embedding operations is a charming research direction and many works have been done, the design of rich model [4] and selection channel aware rich model [3] propel the steganalysis to a state of art. Recently, with the development of the deep learning, many researchers introduce the deep learning methods for steganalysis and obtain satisfying performance [1, 13, 15, 16].

Though big progress was made in traditional steganalysis, it is difficult to move them from laboratory to the real world. In the real world, the amount of payload is unknown and there are large amounts of actors and a vast number of digital media, bringing some challenges to steganalysis. One is that it is time-consuming to classify the objects one by one since the large amount of media. Another problem is caused by the mismatching, since there are multiple actors, the source of media varies. It is difficult to find a common classifier match all source of media. And with the increase of media, the number of false alarm objects increase. Last but not least, when the payload is unknown, the performance of traditional steganalysis worse down significantly. These challenges bring "steganographer detection" into come.

The aim of steganographer detection is to find out the suspect user of sending stego in social network. By doing this we can narrow down the monitor scope in social network or break the communication of the suspect. The steganographer detection is evolved from pooled steganalysis. Pooled steganalysis was first proposed by Ker [6] aiming to detect batch steganography, which take a batch of images as a whole and establish whether it includes stego. Ker et al. [7] proposed identifying the guilty among some normal actors using a cluster method, assuming the normal users taking a batch of images of cover and the guilty taking a batch of images of stego generated by some embedding strategy and steganography algorithm, which propel the pooled steganalysis into steganographer detection. The steganographer can be identified by cluster methods, and MMD distance was used to measure the distance between users in feature space. Ker et al. [8] improved their prior work by introduce outlier detection method to steganographer detection. First the local outlier factor (LOF) in the field of anomaly detection was used to represent the density of the users in feature space, and then ranked users according to the value of the LOF, finally the top K users were taken as the steganographer while K was determined by the author. Li et al. [12] proposed an ensemble clustering method to improve the performance of single clustering. And they proposed feature calibration in steganographer detection in 2017 [11]. Recently, Li et al. [11] proposed steganographer detection with sampling reconstruction. The features were extracted from the sample images reconstructed by select DCT blocks with higher embedding probability, and then the cluster method was used to identify the steganographer. Zheng et al. [18] first proposed using deep residual network to extract the features in steganographer detection.

Though a batch of objects were considered as a whole in these methods, most research concentrated on the design of the feature, they just considered the correlation between neighbored pixels from the perspective of traditional steganalysis and simply combine feature of each single image together to represent the user. What's more, according to our experiment results, when taking the these methods from laboratory to the real world, the performance worse down. Fig. 1 shows the performance of the results using local outlier factor rank in the Bossbase and the twitter images crawled by our partner. The *y* label represents the average rank of actors in 100 users, and the *x* label represents the payload of each image. The nsf5 steganography algorithm was used in this experiment.

In the real world, steganographer may hide messages in digital images by steganography tools, and then post it on social network to achieve his secret communication when it is needed. The behavior of the steganographer in the process of the communication differs from

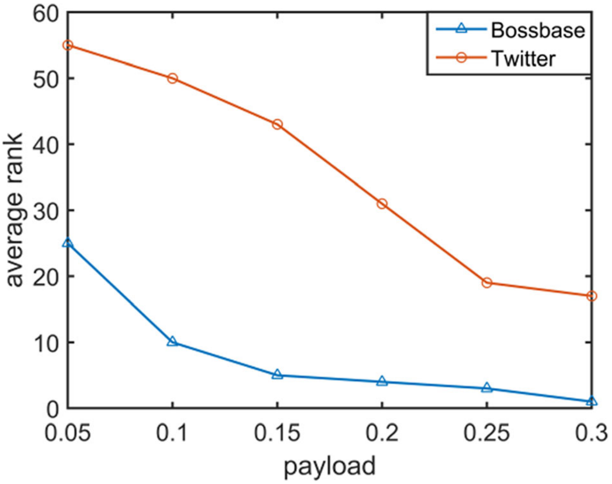


Fig. 1 Compared results of Bossbase and Twitter

normal users in social network, and this difference can be used to identify the steganographer effectively. While the main channel refer to the object itself, we call the behavior information as side channel, such as the semantic relationship between different media the user posts, whether the content the user posted consistent with his or her characteristic and identification. We compare the steganographer detection through behavior analysis to the side-channel attack in cryptography. In cryptography, the side-channel attack is based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. From this point of view, a steganographer detection paradigm is proposed in this paper.

At present, most steganography tools focus on the steganography algorithm, but need users to choose carrier by self, which just take the security in terms of steganography into consideration and only make it difficult to identify an object as cover or stego but ignoring the step of cover selection and the semantics of image content. We have carried out the research about popular JPEG image steganography tools such as JPHide, JSteg and OutGuess and find out that majority of them can output a JPEG format images with secret messages unnoticeably, but do not select suitable carriers for users. And as far as we know, there are some studies about payload spreading among batch of images, which tend to hide messages in images with high security capacity ([2]; Andrew David [9, 17]), but there is no research about selecting carriers for steganography according to users' interests, habits or images contents. So for non-experts steganographer or lazy steganographer, he may choose images as carrier randomly. Under this assumption, we take the correlation between images sequence as an example feature of the side channel to identify the steganographer. A subtractive images adjacent model (SIAM) is designed to represent the correlation between adjacent images, and then a binary classifier is used to classify the users in social network into normal users or steganographers.

The rest of this paper is organized as follows. In Sect.2, we introduce some background knowledge of ensemble classifier and some symbol definition. In Sect.3, the process of steganographer detection and the SIAM feature we designed are described in detail. In Sect.4, the experiment environment and the experiment results are discussed. The conclusion and future work are listed in Sect.5.

2 Preliminary

For better understanding the method we proposed to detect steganographer in the social network, we introduce some background knowledge in this section, including some notation of symbols and the classifier we used.

2.1 Notation

In this article, we define boldface symbols are used for matrices and the calligraphic font is used for the set. The superscript represents the count of a sequence, and the subscript represents the dimensional of the vector or the location of elements in matrix.

We use the symbols R and N to represent the set of all real numbers and integers. For any$x \in R$, the operation of rounding to an integer is denoted as $round(x)$. The truncation function with threshold $T > 0$ is defined for any $x \in R$ as $trunct(x) = x$ for $x \in [-T, T]$ and $trunct(x) = T \cdot sign\ (x)$ otherwise. For a vector $\mathbf{x}$, the absolute operation of $\mathbf{x}$, $abs(\mathbf{x})$ return a vector with all elements equal to the absolute value of the corresponding elements in $\mathbf{x}$. For a finite set $\mathcal{S}$, $|\mathcal{S}|$ denotes the number of its elements, and for a vector $\mathbf{x}$, it refer to the dimension of the vector.

The symbols $\mathbf{X} = \left(X_{ij}\right) \in \{0, \dots, 255\}_{n_1 \times n_2}$ and $\mathbf{Y} = Y_{ij} \in \{0, \dots, 255\}_{n_1 \times n_2}$ always represent pixel values of an 8-bit grayscale cover image with $n = n_1 \times n_2$ pixels and its corresponding stego image. Let $\mathcal{A}$ represent the user in social network, we use "user" and "actor" alternatively in this paper referring to all users in social network. The "anomaly" refer to the actors with behavior different from majority of actors, and the "steganographer" refer to the actors who communicate secretly by hiding messages in images. $\mathbf{I}$ represents the image posted by $\mathcal{A}$, $\mathbf{I} = \mathbf{Y}$ for steganographer and $\mathbf{I} = \mathbf{X}$ for others. Then we have $\mathcal{A} = \left\{\mathbf{I}^{\mathbf{1}}, \mathbf{I}^{\mathbf{2}}, \dots, \mathbf{I}^{\mathbf{N}}\right\}$, N is the number of the images of each actor. To distinguish the $\mathbf{I}$ with the mutual information representation, we represent images using $\mathbf{X}$ instead of $\mathbf{I}$ in the section of behavior analysis, regardless of the steganography.

2.2 Ensemble Classifier

The ensemble classifier we used is proposed to solve the problem of traditional steganalysis [10]. It applies the bagging method called random forest algorithm to improve the performance of binary classifier.

The ensemble classifier consists of L base learners, B^l, $l = 1, \dots, L$, each base learner is implement as the Fisher Linear Discriminant (FLD), and trained on a different d sub-dimensional subspace of the feature space selected uniformly at random. The optimal number of base learners, L, and the dimensionality of each feature subspace, $\mathcal{D}_{sub}$, are determined automatically during ensemble training. The stopping criterion utilize the out-of-bag error to estimate. The final decision is made according to majority voting of individual base learners. The details are as Fig. 2.

3 Steganography Detection Algorithm

The process of the algorithm we designed for steganographer detection is illustrated in Fig. 3. It is accomplished by two steps. Firstly, we extract the SIAM feature we designed representing the correlation between adjacent images of each user, and then a binary classifier is used for identifying

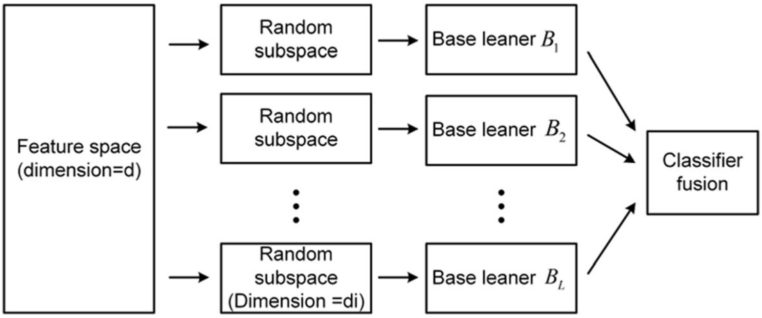


Fig. 2 Ensemble classifier

the steganographer, in this paper, we use the ensemble classifier. The details about feature design are described in the following subsection.

3.1 Subtractive Images Adjecent Model(SIAM) Feature Design

As we discussed in the introduction, since the majority of steganography tools only focus on the steganography algorithm but ignore the attack from other perspective, there are many side channel information we can use to detect the steganographer. In this section, we take the relationship between images sequence as an example of side channel. As we all know, the images posted by the user can be modeled sequentially, and there should be some relationship between adjacent images in sequence; but for the steganographer, the posted images are used for secret communication, and the website account of the steganographer may be only used when the communication is needed, so the relationship between his images sequence may not so clear and differs from the normal users. From this point of view, we establish the behavior features according to the relationship between adjacent images posted by the users.

3.1.1 Rationale of SIAM Feature

Similar to the design of feature reflect the relationship between the pixels [11], we design the feature that reflects the relationship between adjacent images in a sequence. The feature extraction process is demonstrated in Fig. 4.

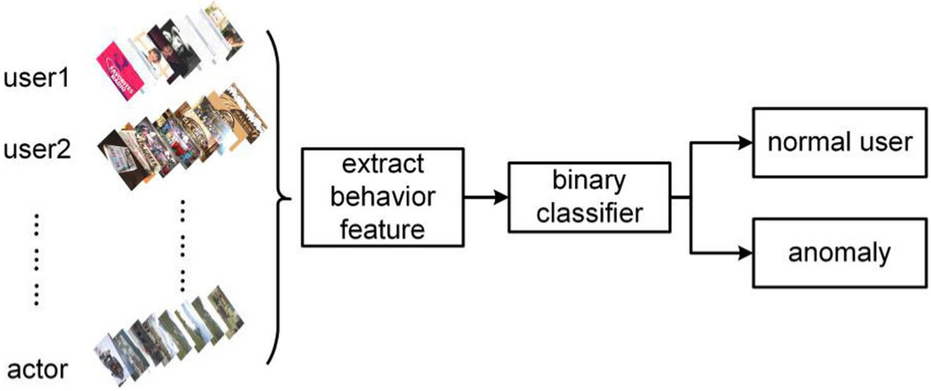


Fig. 3 Diagram of the steganographer detection

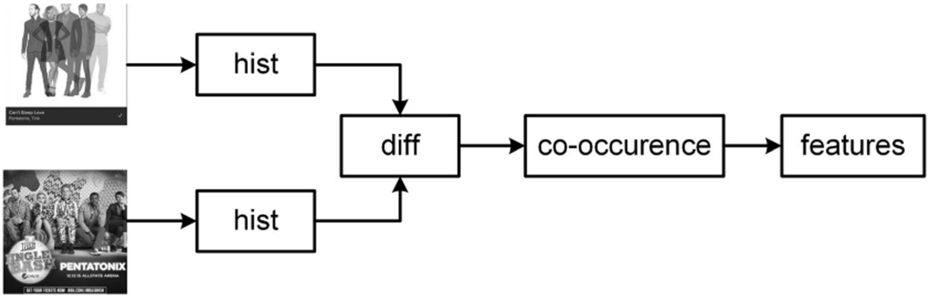


Fig. 4 Process of behavior feature extraction

First, we transform the images to gray scale and utilize the gray scale histogram to represent each image, the second-order residual filter is used to calculate the residual of adjacent images in sequence. Then we calculate the histogram and the second co-occurrence of the residual. In principle, higher order co-occurrence between pixels images can be modeled by histograms of pairs, triples, or larger groups of neighboring images, the reason for which we discard higher order co-occurrence are as follows.

1) The number of bins of the histograms increases exponentially as the increase of the order.
2) For most users in social websites such as twitter, a piece of twitter may include at most 4 images. The correlation between images with large interval may be submerged by the noise.

Due to high time correlation in images sequential, the grayscale histograms of neighboring images are similar. To verify this, we use 100 users' data we crawled, calculate the difference of the adjacent images histograms and the statistical results are shown in Fig. 5. Firstly, all the images are preprocessed to 8-bit grayscale images with size of 512 × 512, and we use the histogram of 256 bins to represent the images, such as $\mathrm{H}^i = \left[h_1^i, h_2^i, \ldots, h_{256}^i\right]$represent i_{th} image. h_k^i stands for the value of the k_{th} bin of the histogram of i_{th} image. Then we calculate the difference of each bin of the histogram of each adjacent images $R = \mid H^i - H^{i+1} \mid$. Thus we obtain 256-dimension residual vector for each pair of adjacent images. We calculate the

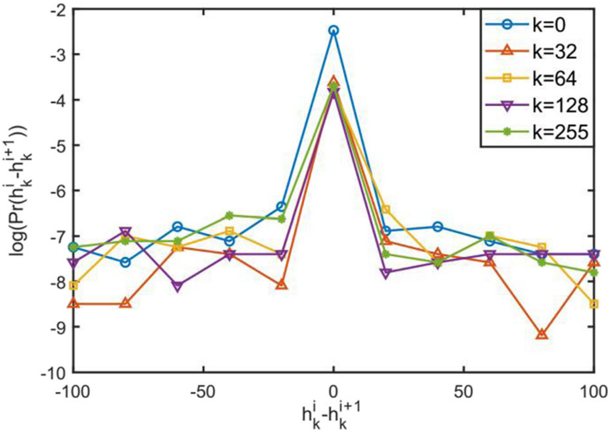


Fig. 5 Probability for $h_k^i - h_k^{i+1}$, estimated from 10,000 images of 100 twitter users

frequency of each dimension of R, $P\left(h_k^i - h_k^{i+1}\right)$. For easy to observe, we take the logarithm of P and take $k = 0, 32, 64, 128, 255$ for example in Fig. 5. As we can see, for all k, the maximum of $\log\left(\Pr\left(h_k^i - h_k^{i+1}\right)\right)$ locate around $h_k^i - h_k^{i+1} = 0$, which means that the majority of the difference of neighboring images are 0, and it is independent of the grayscale k. Since the residual of adjacent images are small, the correlation of adjacent images in sequence of normal user is verified and base on this, we design the images adjacent model (SIAM) in Sect.3.1.2.

A good feature should capture those characteristics of images that can be robustly estimated. To prove that the feature is independent of the images content. We calculate the mutual information between $h_k^i - h_k^{i+1}$ and h_k^i, represented as $I\left(h_k^i - h_k^{i+1}, h_k^i\right)$, $H\left(h_k^i - h_k^{i+1}\right)$ stands for the entropy of $h_k^i - h_k^{i+1}$ and $H\left(h_k^{i+1} | h_k^i\right)$ stands for the conditional entropy of h_k^{i+1} on condition that h_k^i is known.

$$I\left(h_k^i - h_k^{i+1}, h_k^i\right) = H\left(h_k^i - h_k^{i+1}\right) - H\left(h_k^i - h_k^{i+1} | h_k^i\right) = H\left(h_k^i - h_k^{i+1}\right) - H\left(h_k^{i+1} | h_k^i\right) \tag{1}$$

$$H\left(h_k^i - h_k^{i+1}\right) = \sum_{i=1}^{N-1} p\left(h_k^i - h_k^{i+1}\right) \log\left(p\left(h_k^i - h_k^{i+1}\right)\right) \quad , \quad k = 1, \dots, 256 \tag{2}$$

$$H\left(h_k^{i+1} | h_k^i\right) = \sum_{i=1}^{N-1} p\left(h_k^i, h_k^{i+1}\right) \log\left(p\left(h_k^i | h_k^{i+1}\right)\right) \quad , \quad k = 1, \dots, 256 \tag{3}$$

We replace the probability with the frenquency, and the frequency is estimated from 10,000 images of twitter users while each user own 100 sequential images. We calculate the mutual information for $k = 1, \dots, 256$ respectively, and the experiments result is shown in Fig. 6. We can see the dependency between $h_k^i - h_k^{i+1}$ and h_k^i is farely small when k is larger than 50.

This allows the researchers to use the difference between $h_k^i - h_k^{i+1}$ and h_k^i to model the relationship between neighbored images in sequence flow of users in twitter.

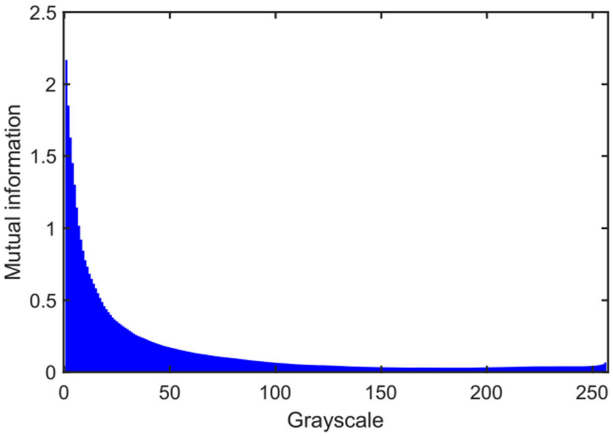


Fig. 6 The mutual information between $h_k^i - h_k^{i+1}$ and h_k^i for different gray scale

3.1.2 The SIAM Feature

We call the feature we modled as subtractive images adjacent model (SIAM). The detailed descreption about the feature extraction are as follows.

We represent the image by the grayscale histogram, for example, $\mathbf{h}^i = \left[h_1^i, h_2^i, \dots, h_{256}^i\right]$ represent the i_{th}images $\mathbf{X}^i$ of a user, the feature extraction for the user is as follows.

1) Preprocess

We use 100 users' images sequence with 100 images of each user. All images are preprocessed to 8-bit grayscale images with size of 512 × 512.

2) Residual calculation

We define the residual between images $\mathbf{X}^i$ and $\mathbf{X}^j$ as:

$$\mathbf{d}^{i,j} = abs\left(\mathbf{h}^i - \mathbf{h}^j\right) \tag{4}$$

And calculate $\mathbf{d}^{i,\,i+1}$ of every actor, for $i = 1, \dots, N-1$.

Since for image $\mathbf{X}^i = (X_{ij}) \in \{0, \dots, 255\}_{n1 \times n2}$, $\mathrm{h}_k^i \in \{0, \dots, n1 \times n2\}$, $d_k^{i,j} \in \{0, \dots, n1 \times n2\}$, $k = 1, \dots, 256$. The difference can also be seen as the result of a high pass filter imposed on the images sequence, it can capture the local changes of the sequence as well as the correlation between images.

3) Logarithm and rounding

We take the logarithm of $\mathbf{d}^{i,\,i+1}$ for $i = 1, \dots, N-1$, and round it to $[0, T]$ with a threshold T, obtain $round(\log \mathbf{d}^{i,\,i+1})$ for $i = 1, \dots, N-1$, combine them together to obtain $\mathbf{d}$,

$$\mathbf{d} = \begin{bmatrix} round\left(\log \mathbf{d}^{1,2}\right) \\ round\left(\log \mathbf{d}^{2,3}\right) \\ \vdots \\ round\left(\log \mathbf{d}^{N-1,N}\right) \end{bmatrix} \tag{5}$$

4) Co-occurrence matrix

We calculate the histogram and second order co-occurrence of $\mathbf{d}$, which is represented as

$$\mathbf{h}' = [h'_1, \dots, h'_{T+1}] \tag{6}$$

$$\mathbf{c} = \begin{bmatrix} c_{1,1}, \dots, c_{1,T+1} \\ \cdot \quad , \dots, \quad \cdot \\ \cdot \quad , \dots, \quad \cdot \\ c_{T+1,1}, \dots, c_{T+1,T+1} \end{bmatrix} \tag{7}$$

respectively. Since $c_{i,\,j}$ and $c_{j,\,i}$ represent the similar relationship, we combine $c_{i,\,j}$ and $c_{j,\,i}$ together by addition with the $c_{i,\,i}$ unchanged. We reshape the left element to a vector $\mathbf{c}'$, obtaining $\mathbf{c}' = [c_1, \dots c_{91}]$.

5) Combination

We combine the histogram and the co-occurrence of $\mathbf{d}$together and obtaining $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2]$, $\mathbf{f}_1 = \mathbf{h}$, $\mathbf{f}_2 = \mathbf{c}'$.

$$|\mathbf{F}| = |\mathbf{f_1}| + |\mathbf{f_2}| = T + 1 + \frac{(T+2) \times (T+1)}{2} \tag{8}$$

In this paper, we set $T = 12$, thus obtain a 104-d feature for every actor.

4 Experiment result

4.1 Simulating the Real World

To simulate the environment that identifying the steganographer by monitoring the whole network in the real world. We take twitter as an example since it is one of the most popular social networks with a large number of users and multiple digital media. We crawl images of 3000 users from twitter using the tweepy API, the tweepy is a public API for twitter developers [14]. For the data we crawled, we only preserve the JPEG format images and crop the images to size of 512 × 512 with the center unchanged, and then we discard users whose images number is less than 100. After the preprocessing, we have 700 users left. For each of left users, we take images sequence of 100 for experiment.

We divide 700 users into two parts of 350 users randomly. For one part, we keep the original order of the images unchanged to mimic normal actors, and for another, we combine all users' images together to generate a database, when we simulate a steganographer, we choose images randomly from this database. Though the steganography operation does not affect the behavior feature, to better simulate the steganographer in the real world, the steganography algorithm is used for steganographer. The steganography algorithm we used is nsf5 [5] and embed evenly among batch of images [9]. In this paper, all experiments are implemented by simulating embedding.

In this paper, we define the unconscious anomaly users pick images randomly for steganography, and the reasons are as follows: firstly, the normal users' behavior usually not so easy to model and imitate, such as how many images he posted at a time and how often he posts a twit, and usually the account of the anomaly user are only used when the communication is needed, it is consuming and difficult to model the normal users' behavior and keep the account act as a normal user. Secondly, the images used for steganography is usually selected from a database, whether it is the database used especially for steganography, the images of which is more suitable for steganography and difficult to detect using traditional steganalysis, or a database composed of the photos taken by the user self and pictures download from the network, it is difficult to find images satisfying the rule of a normal user. It seems that one simple way to keep the rule of a normal account is to crawl normal user's images from social network and save them sequentially, and then select carriers from them sequentially for communication when needed. But the problem of it is that when using techniques of images retrieval to find the same images in the network, by comparing the two images, It is easy to judge whether there is information hiding in the image or not. Considering the feasibility and for convenience, we assume the unconscious steganographer choose images randomly.

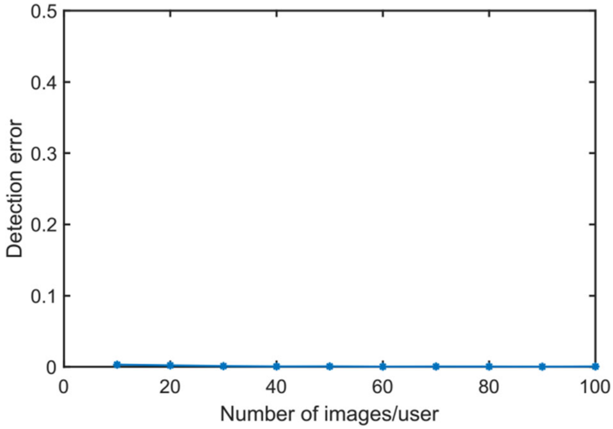


Fig. 7 Experiment result for different number of images

We do not compare our results with the existing steganographer detection methods. The reason is that existing methods focus on detecting the existence of the secret message while our method concentrate on the behavior anomaly. The accuracy of the other methods depend on the batch strategies and the steganography algorithm while our method accuracy is influenced by the carrier selection strategy.

4.2 Steganographer without Behavior Security Consciousness

In this section, we assume the steganographer unconscious of behavior security, he only logs in his account and posts images when the secret communication is needed, which means all images posted by the steganographer is random. We extract the SIAM feature of each actor and then the ensemble classifier is used for classification, both the "normal users" and "anomaly users" are divided into two part, half for training called training set and the rest for testing called testing set. The experiment result shows the false alarm ratio is 0.0001 and the missed-detection ratio is 0.0004. The experiment result shows that we can almost find the steganographer of behavior anomaly accuracy.

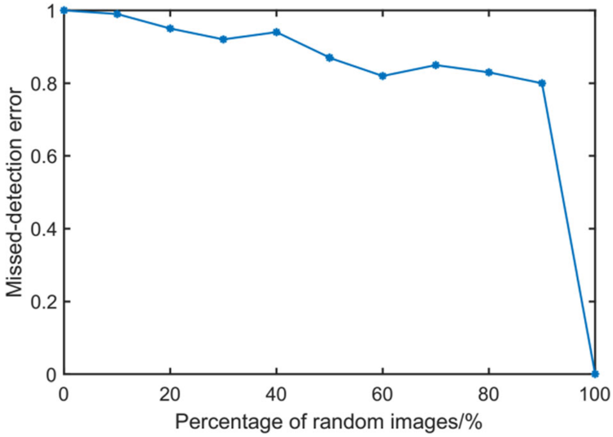


Fig. 8 The performance of the original classifier

Table 1 The missed detection error of retrained classifier

Percentage of random images	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Missed detection error	0.02	0.01	0.02	0.1	0.1	0.1	0.1	0.1	0.4	0.00

4.3 Robustness of the Algorithm

In the real world, there exists some user whose images is less than 100, and it is probably that the number of steganographer's images is less than 100. To test the performance of our algorithm under this condition, we carry out the experiment with different number of images. And the result is shown in Fig. 7, the detection error is the average of the false alarm ratio and the missed-detection ratio. It demonstrates that when the number of images decrease, the algorithm still works well.

4.4 Steganographer with Different Level of Security Consciousness

In Sect.5.1, we assume all the images posted by the steganographer are chosen randomly. In the real world, for some steganographer with behavior security consciousness, he or she may try to mimic the normal user in peace time by posting images of usual life as normal users. When the communication is needed, he may choose carrier from his or her database for hiding information. Though it is time-consuming, there may be someone who are willing to pay much for the confidentiality of the communication. Considering this situation and for better simulating the real world, we define steganographer of different level of behavior security consciousness according to the percentage of the random images in images sequence of 100. We implement this by inserting some random images in normal users' data we crawled. The percentage of the random images determine the security level of the steganographer, and the total number of images of each user is 100. We use the classifier trained in Sect. 4.1 for steganographer without behavior security consciouness to test different level steganographers, the experiment results is shown in Fig. 8.

It demenstrated that when the testing steganographer is mismatch with the trained classifier, the perfroamnce worse down. To reduce the impact of the mismatch, we regenerate the training set by mixing steganographers of different level of security, while each level of steganographer contributes the same number of training samples. Apply the retrained classifer to test steganographer with different security consciousness, the false alarm ratio is increased to 0.0057, and the missed-detection error is as Table 1. It demenstrated that by adding steganographers with different level of security consciousness into training set, we obtain a general classifier effective for steganographer with differen level of security consciousness.

5 Conclusions

In this paper, we try to solve the problem of steganographer detection from a new respect of view and design a new paradigm of steganographer detection in social network. We propose utilizing features of side channel information which is neglected by the steganographer to classify the normal user and steganographer. As an example, this paper focus on the relationship between adjacent images, the Images Adjecent Model(SIAM) feature is designed to represent the correlation between images. Considering the unknown randomness of the steganographer's images in the real world, we improved our method by using mixed training set, thus decrease the effect of mismatch.

Since in social network, there are many behavior that can leak the communication of the steganographer, such as the consistency between different media, the relationship between user's portrait and the objects he or she posts. We will keep study more about designing behavior features that can distinguish the steganographer and the normal user by taking more factors into consideration.

Acknowledgments This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 6157245.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

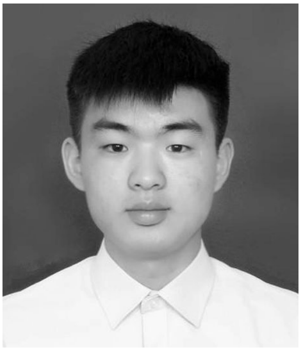
References

1. Chen M, Sedighi V, Boroumand M (2017) JPEG-Phase-Aware Convolutional Neural Network for Steganalysis of JPEG Images
2. Cogranne R, Sedighi V, Fridrich J (2017) Practical strategies for content-adaptive batch steganography and pooled steganalysis. Paper presented at the Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on
3. Denemark T, Sedighi V, Holub V (2014) Selection-channel-aware rich model for steganalysis of digital images. Paper presented at the Information Forensics and Security (WIFS), 2014 IEEE International Workshop on
4. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensic Secur 7:868–882
5. Fridrich J, Pevný T, Kodovský J (2007) Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. Paper presented at the Proceedings of the 9th workshop on Multimedia & security
6. Ker AD (2006) Batch steganography and pooled steganalysis. Paper presented at the Information Hiding
7. Ker AD, Pevný T (2011) A new paradigm for steganalysis via clustering. Paper presented at the Media Forensics and Security
8. Ker AD, Pevný T (2014) The steganographer is the outlier: realistic large-scale steganalysis. IEEE Trans Inf Forensics Secur 9(9):1424−1435
9. Ker AD, Pevny T (2012) Batch steganography in the real world. Paper presented at the Proceedings of the on Multimedia and security
10. Kodovsky J, Fridrich J, Holub V (2012) Ensemble classifiers for steganalysis of digital media. IEEE Trans Inf Forensic Secur 7(2):432–444
11. Li F, Wen M, Lei J (2017) Efficient steganographer detection over social networks with sampling reconstruction. Peer-to-Peer Networking and Applications, 1-16
12. Li F, Wu K, Lei J, Wen M, Bi Z, Gu C (2016) Steganalysis over large-scale social networks with high-order joint features and clustering ensembles. IEEE Trans Inf Forensic Secur 11(2):344–357
13. Qian Y, Dong J, Wang W, Tan T (2015) Deep learning for steganalysis via convolutional neural networks. SPIE Media Watermarking, Security, and Forensics, vol 9409. https://doi.org/10.1117/12.2083479
14. Russell MA (2013) Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More: " O'Reilly Media, Inc."
15. Tan S, Li B (2014) Stacked convolutional auto-encoders for steganalysis of digital images. Paper presented at the Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)
16. Xu G, Wu H, Shi YQ (2016a) Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett 23(5):708–712

17. Zhao Z, Guan Q, Zhao X (2016) Embedding Strategy for Batch Adaptive Steganography. Paper presented at the International Workshop on Digital Watermarking
18. Zheng, M., Zhang, S.-h., & Wu, S. (2017). Steganographer detection via deep residual network. Paper presented at the Multimedia and Expo (ICME), 2017 IEEE International Conference on.

Li Li received her B.S. degree in 2016 from Harbin Engineering University (HEU), Harbin, China. She is currently pursuing the Ph.D. degree in Information Security in University of Science and Technology of China (USTC). Her research interests include steganographer detection and steganalysis.

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.



Kejiang Chen received his B.S. degree in 2015 from Shanghai University, Shanghai, China. He is now pursuing the Ph.D. degree in University of Science and Technology of China. His research interests include data hiding, and deep learning.

Hongyue Zha has received a bachelor's degree in 2014 at University of Science and Technology of China(USTC). Since then, he has been persuing his doctor's degree at USTC. He was intrested in the research of steganography, steganalysis, adversial networks and appliments related.

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 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.