Multimedia Tools and Applications (2020) 79:18293–18316
https://doi.org/10.1007/s11042-020-08716-w

TS-CSW: text steganalysis and hidden capacity estimation based on convolutional sliding windows

Zhongliang Yang[1] · Yongfeng Huang[1] · Yu-Jin Zhang[1]

Received: 20 March 2019 / Revised: 10 December 2019 / Accepted: 31 January 2020 /
Published online: 2 March 2020

Abstract
With the rapid development of natural language processing (NLP) technology in the past few years, the automatic steganographic texts generation methods have been greatly developed. Benefiting from the powerful feature extraction and expression capabilities of neural networks, these methods can generate steganographic texts with both relatively high concealment and high hidden capacity at the same time. For these steganographic methods, previous steganalysis models show unsatisfactory detection performance, which remains an unsolved problem and poses a great threat to the security of cyberspace. In this paper, we first collect a large text steganalysis (T-Steg) dataset, which contains a total number of 396,000 texts with various embedding rates under various formats. We analyze that there are three kinds of word correlation patterns in texts. Then we propose a new text steganalysis model based on convolutional sliding windows (TS-CSW), which use convolutional sliding windows (CSW) with multiple sizes to extract those correlation features. We observed that these word correlation features in the generated steganographic texts would be distorted after being embedded with secret information. These subtle changes of correlation feature distribution could then be used for text steganalysis. We use the samples collected in T-Steg dataset to train and test the proposed steganalysis method. Experimental results show that the proposed model can not only achieve a high steganalysis performance, but can even estimate the amount of secret information embedded in the generated steganographic texts, which shows a state-of-the-art performance.

Keywords Text steganography · Text steganalysis · Convolutional sliding window · Words correlation · Hidden capacity estimation

✉ Zhongliang Yang
yangzl15@mails.tsinghua.edu.cn

[1] The Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

1 Introduction

Claude E. Shannon has summarized that there are three basic information security systems [34]: encryption system, privacy system, and concealment system. Compared with the other two systems, the biggest characteristic of a concealment system is the extremely strong concealment of information. It embeds secret information into various normal carriers and then transmits them through public channels, hiding the existence of secret information so as to protect the security of it [51]. However, strong concealment can also be used by hackers, terrorists, and other law breakers for malicious intentions. Hence, designing an efficient steganalysis method becomes an increasingly promising and challenging task.

A concealment system can be illustrated by Simmons' "Prisoners' Problem" [37] which can be described as follows. Alice needs to transmit some secret message $m \in \mathcal{M}$ to Bob and try to avoid being suspected by Eve. She uses the steganographic function *f()* to embed the secret information m into normal carrier $c \in \mathcal{C}$ under the control of a certain key $k_A \in \mathcal{K}$, namely:

$$Emb : \mathcal{C} \times \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{S}, f(c, k_A, m) = s. \tag{1}$$

Where s represents the steganographic carrier which contains secret information m. Contrary to Alice, Bob needs to use the extraction function *g()* to accurately extract the secret information m from received carrier s, that is:

$$Ext : \mathcal{S} \times \mathcal{K} \rightarrow \mathcal{M}, g(s, k_B) = m. \tag{2}$$

Generally, in order to ensure the security of the covert communication system, Alice must try to reduce the statistical distribution differences between normal carriers and steganographic carriers, that is:

$$d_f(P_{\mathcal{C}}, P_{\mathcal{S}}) \leq \varepsilon. \tag{3}$$

In Simmons' "Prisoners' Problem", every message Alice sends to Bob will be carefully reviewed by Eve, whose task is to determine whether the transmitting carrier contains hidden information or not. In general, embedding additional information in the carrier is equivalent to superimposing noise, so it will almost certainly affect the statistical distribution characteristics of the carrier in some aspects. Therefore, Eve should try her best to find the differences in statistical distribution characteristics of the carrier before and after steganography.

Generally, according to different types of carriers, steganographic technologies can be divided into image steganography [10, 26, 44, 53], audio steganography [45, 47, 54], text steganography [20, 31, 49, 52, 55] and so on. Since text is one of the most frequently used information carrier in people's daily lives, using text for information hiding has long attracted the interest of a large number of researchers. The earliest record of using text to convey secret information could be traced back to several centuries ago [10]. However, hackers, terrorists, and other law breakers may also use such technologies to transmit dangerous information and endanger public safety. Therefore, it is very important, as well as challenging, to conduct in-depth research on text steganalysis and to achieve the accurate identification of steganographic text in cyberspace.

Text steganographic methods can usually be divided into three categories: text steganography based on retrieval mode [57, 58], text steganography based on modification mode [2, 25, 30, 35, 56], and text steganography based on generation mode [9, 21, 22, 49, 50, 55]. Text steganography based on retrieval mode usually begins by selecting a largerset of carriers and then appropriately encoding the samples in the set. During transmission, Alice selects different carriers according to the secret information needs to be transmitted and realizes the purpose of covert communication [57, 58]. They usually have very high imperceptibility because the carrier it transmitted is always “100% natural" [57]. Text steganographic methods based on modification mode hide secret information by selecting and modifying insensitive parts of texts, such as synonyms [25, 56] or syntactical structures [30]. However, both retrieval mode and modification mode can only convey very little information and have very low efficiency of information transmission, which makes them impractical.

The third kind of text steganographic methods is based on text generation technology. The most important characteristic, which is also the biggest difference from the first two kinds of methods, is that these methods do not have a carrier in advance, but need to automatically generate a stego carrier based on the secret information [9, 21, 22, 25, 50, 55]. Therefore, this kind of methods can imitate the statistical characteristics of normal carriers and reconstruct a stego carrier which is as close as possible to the statistical distribution of normal carriers, thus resisting steganalysis based on statistical analysis [55]. But this could also be the biggest difficulty of this kind of methods. Some of the early methods were difficult to generate steganographic text that looked natural enough [4, 6, 29, 40], thus limited the popularity of such methods.

In recent years, with the extensive applications of deep neural network technology in Natural Language Processing, there has been an increasing number of automatic text generation related works, like image captioning [46], neural machine translation [1], dialogue systems [18], and so on. With these technologies, some steganographic text automatic generation methods based on neural networks have appeared in recent years [9, 21, 49, 55]. This type of methods utilizes the powerful feature extraction and expression capabilities of neural networks, analyzes the statistical feature distribution of a large number of training samples and then reconstructs samples that conform to such statistical distribution. In this way, the generated steganographic texts can conform to formula (3) and achieve relatively high concealment. Figure 1 shows some steganographic texts that are automatically generated by latest steganographic methods. The special format text (Chinese poetry) is generated by the model proposed by Y. Luo *et al.*

风雨	**Wind and Rain**
江门连海色，	Rivers flow into the sea in the distance,
幽霭北灼山。	Mists among the mountains shine in the north.
送看官听澭，	Sending my guests away and walking in the mist,
风烛著暖烟。	Candles in the wind blend into the warm smoke.
1 bit/word	i'm ok now but i have to go to school tomorrow.
2 bits/word	i know i am so jealous that they didn't even get to go.
3 bits/word	i've just been in my class room and is getting tired now.

Fig. 1 Some steganographic texts that are automatically generated by latest steganography methods. The special format text (Chinese poetry) is generated by the model proposed by Y. Luo *et al.* [21] and natural texts are generated by the model proposed by T. Fang *et al.* [9]

[21] and natural texts are generated by the model proposed by T. Fang *et al.* [9]. It can be seen that these generated steganographic texts are very close to human-written texts, which poses a great challenge to text steganalysis models.

Traditional text steganalysis methods mainly first extract some text features, like word frequency [43], words transition probability [27], and some other manual designed features [32, 38]. Then they analyze the differences between these features before and after steganography to determine whether the text contains secret information [27, 32, 38, 41, 43]. However, most of the statistical features utilized by these methods are simple, which may limite the detection capabilities of these methods. At present, for steganographic texts generated by neural networks, whether special format text or natural text,previous text steganalysis methods show unsatisfactory results, which poses a great threat to the security of cyberspace, and thus becomes an urgent problem that needs to be solved.

In this paper, we propose a new text steganalysis method based on convolutional sliding windows (TS-CSW). We collect and release a large text steganalysis (T-Steg) dataset, which contains a total number of 396,000 texts with various embedding rates under various formats. Then we conduct a detailed analysis of the correlations between words in texts and define three kinds of word correlation patterns. Then we use multi-size convolution sliding windows (CSW) to extract the correlation features of these words. After mapping these features to high-dimensional feature space, we observe that when additional information is embedded in the generated texts, these features will migrate in high-dimensional space. Therefore, we can use the distribution differences ofthese features in high-dimensional space to achieve high accuracy steganalysis. Further experiments show that according to the subtle distribution differences in the high-dimensional feature space, we can even estimate the capacity of hidden information in texts, which may point out another possible direction for future text steganalysis research.

In the remainder of this paper, Section 2 introduces related work, including automatic steganographic text generation and text steganalysis. Section 3 introduces the detailed explanation of the proposed model TS-CSW. The following part, section 4, presents the experimental evaluation results and gives a comprehensive discussion. Finally, conclusions are drawn in Section 5.

2 Related work

2.1 Automatic steganographic text generation

Compared with other methods, the steganographic methods based on automatic text generation are characterized by the fact that they do not need to be given carrier texts in advance. Instead, they can automatically generate a text carrier based on secret information. This unique feature makes this kind of methods usually achieve a relatively high embedded capacity, so it has long been considered a promising research direction. However, due to technical limitations, some of the early research works can only generate steganographic text with a very simple pattern that is easy to be recognized [4, 40]. With these early attempts, a lot of researchers have been trying to combine text steganography with statistical natural language processing, and a large number of natural language processing techniques have been used to automatically generate steganographic text [6, 7, 22, 29, 36]. Most of the text automatic generation models are designed to fit the statistical language model, whose mathematical expressions are as follows:

$$\begin{aligned} p(X) &= p(x_1, x_2, x_3, \ldots, x_n) \\ &= p(x_1)p(x_2|x_1)\ldots p(x_n|x_1, x_2, \ldots, x_{n-1}). \end{aligned} \tag{4}$$

Where X denotes the whole sentence with a length of n and x_i denotes the i-th word in it. *p(X)* assigns the probability to the whole sequence. Most of the previous works use Markov chain model to calculate the number of common occurrences of each phrase in the training set and obtain the conditional probability estimate. They encode the conditional probability of each word and achieve the purpose of hiding confidential information in the text generation process [6, 7, 29, 36, 50]. Unfortunately, due to the limitations of the Markov model, the textual steganographic carriers they generated were still not perfect [55].

In recent years, along with the development of neural network technology [17, 33], the field of statistical natural language processing has also been greatly developed, more and more automatic text generation technologies based on neural networks have emerged [1, 7, 29, 46]. This type of technology has gradually migrated to the field of information hiding, and more and more models for automatic generation of steganographic texts based on neural networks have emerged [9, 21, 49, 55]. In these methods, numerous texts are fed to neural networks to learn the statistical language model of plain texts, and then texts with hidden information satisfying the learned statistical language model could be generated. Furthermore, based on the generated texts format, these methods can be further divided into steganography based on special format texts generation [21] and natural texts generation [9, 49, 55]. Compared with natural texts, special format texts combine some syntactic rule information in addition to the statistical distribution characteristics of training samples in the generation process [21]. This makes the generated special format texts consistent with the training samples in terms of statistical distribution and syntactic structure.

2.2 Text steganalysis

Steganalysis is a process of identifying whether a given carrier is a normal carrier c or a steganographic carrier s. According to Eq. (3), the task of steganalysis is mainly to find the differences in the distribution of statistical features between normal carriers and steganographic carriers. Therefore, current text steganalysis methods basically adopt this security framework, that is, by constructing specific statistical features or analytical methods, to find the differences in statistical distribution between the covertext and the stegotext to conduct steganalysis.

For example, Yang *et al.* [43] proposed a novel linguistics steganalysis approach based on meta features and immune clone mechanism. They defined 57 meta features to represent texts, including the average length of words, the space rate, the percentage of letters and so on. Then the immune clone mechanism was exploited to select appropriate features so as to constitute effective detectors. Meng *et al.* [27] proposed a linguistic steganography detecting algorithm using Statistical Language Model (SLM). They calculated the perplexity of normal text and stego-text with the language model, and then determined whether the text inputted contained covert information by setting a threshold. Taskiran tet al. [38] first calculated the statistical correlation, which was measured by N-window mutual information, of the words in the generated steganographic text, and then used SVM to classify the given text into stego-text or normaltext. Samanta *et al.* [32] proposed statistical text steganalysis tools based on Bayesian

Estimation and Correlation Coefficient methodologies. Din *et al.* [8] proposed a formalization of genetic algorithm method in order to detect hidden message on an inputted text. Chen *et al.* [5] used the statistical characteristics of correlations between the general service words gathered in a dictionary to classify the given textsegments into stego-text segments and normal text segments. These text steganalysis techniques only analyze the unilateral statistical properties of the text. Once the neural network learns these feature expressions and generates steganographic texts that match the statistical distribution with the training samples, these steganalysis methods will face challenges.

3 TS-CSW methodology

The information-theoretic definition of steganographic security starts with the basic assumption that the cover source can be described by a probability distribution, $P_{\mathcal{C}}$, on the space of all possible cover, $\mathcal{C}$. The value $P_{hcalC}(\mathcal{B}) = \int_{\mathcal{B}} P_{\mathcal{C}}(X)dX$ is the probability of selecting cover $X \in \mathcal{B} \subset \mathcal{C}$ for hiding a message. For a given stegosystem assuming on its input covers $X \in \mathcal{C}, X \sim P_{\mathcal{C}}$ and messages $m \in \mathcal{M}$, the distribution of stego cover is $P_{\mathcal{S}}$. Steganalysis can be viewed as a detection problem which can be modeled as a simple hypothesis testing [3]:

$$H_0 : X \sim P_{\mathcal{C}}$$

$$H_1 : X \sim P_{\mathcal{S}}.$$

Any steganalysis method can be described by a map $F : \mathbb{R}^d \rightarrow \{0, 1\}$, where $F = 0$ means that x is detected as cover, while $F = 1$ means that x is detected as stego. Therefore, the methods of steganalysis are usually to construct a variety of corresponding statistical features, and based on these features to find the differences in the statistical distribution between the covertext and the stegotext. Figure 2 shows the overall framework of the proposed text steganalysis method (TS-CSW). Our model consists of two parts, one is a words correlation extraction module and

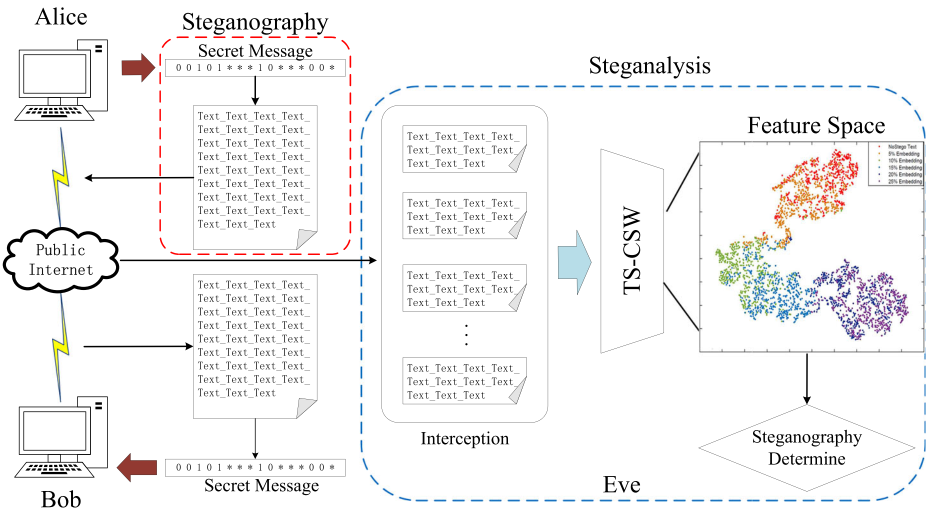


Fig. 2 The overall framework of the proposed text steganalysis method (TS-CSW)

the other is a feature classification module. The first part mainly uses multi-size convolutional sliding windows to extract the words correlation features of texts. The second part analyzes the extracted features and finally determines whether the text contains covert information or not.

3.1 Words correlation analysis

Text is a highly-encoded information carrier, so there are strong semantic coherences and correlations between words in texts. Once we embed additional information into the text, it is possible to influence the semantic correlations of these words. Here we first give a detailed analysis of the correlations between words in the text. For a text with multiple sentences, we can represent it as: $T = \{X_1, X_2, \ldots, X_m\}$, where X_i is the i-th sentence and m is the number of sentences in the text. Each sentence X_i consists of multiple words arranged in order, which can be expressed as $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\}$, where $x_{i,j}$ is the j-th word in X_i and n_i is the number of words in X_i. When all words are uncorrelated, their appearances are independent. Therefore, we have

$$\begin{aligned} &P(x_{i,j}, x_{k,l}) = P(x_{i,j}) \cdot P(x_{k,l}), \\ &s.t. \quad \forall i, k \in [1, m]; j \in [1, n_i]; l \in [1, n_k]. \end{aligned} \tag{5}$$

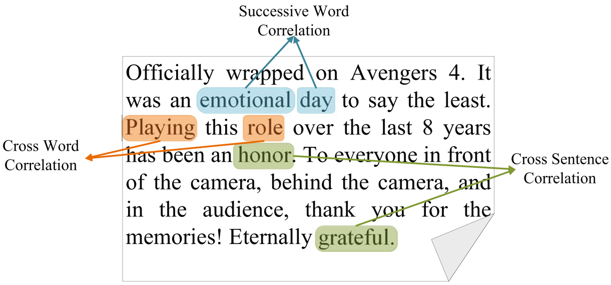
When the two sides of Eq. (5) are not equal, for example, the left side is of higher value than right side, we think there is a correlation between the two words $x_{i,j}$ and $x_{k,l}$. Since there are multiple collocations between wordsin the texts, we summarised three kinds of words correlations, which have been explained in Fig. 3:

- **Successive Word Correlation**

Usually for a sentence X_i, we can model it as a sequence signal form $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\}$. For a sequence signal, the successive signal, which are words in a sentence, often have strong semantic coherence and correlation. We name this kind of correlation as successive word correlation, which can be expressed as: $P(x_{i,j}, x_{k,l} | i = k, |l - j| = 1)$.

- **Cross word correlation**

Usually for a sentence containing multiple words, due to the syntactic rules, it is possible that two words that are not adjacent will form a collocation relationship with strong semantic relevance. Each word in a sentence not only has semantic connections with adjacent words, but also may have potential semantic connections with distant words. Therefore, we define a long-



Officially wrapped on Avengers 4. It was an emotional day to say the least. Playing this role over the last 8 years has been an honor. To everyone in front of the camera, behind the camera, and in the audience, thank you for the memories! Eternally grateful.

Fig. 3 Three kinds of words correlations in a text

distance dependency called cross word correlation, which can be expressed as $P(x_{i,j}, x_{k,l} | i = k, |l - j| < n_i)$.

- **Cross sentence correlation**

More generally, for a text containing multiple sentences, there is also a certain semantic relevance between different sentences. Therefore, there may be some potential semantic correlations between words in different sentences, which we call cross sentence correlation and can be expressed as: $P(x_{i,j}, x_{k,l} | i \neq k, j \in [1, n_i], l \in [1, n_k])$.

Previous text steganalysis methods also tried to analyze these kinds of words correlations, and then perform steganalysis by analyzing whether there are changes in the statistical distribution after steganography. Meng *et al.* [27] mainly used 3-gram to model texts, which analysed the correlations between each word and the two words before it. Taskiran *et al.* [38] defined an N-window mutual information, analyzed the correlations between each wordand the surrounding N words, and then used SVM to classify them. Samanta *et al.* [32] proposed statistical text steganalysis tools that used the Bayesian model to calculate the correlation coefficient between words and then performed text steganalysis. In paper [5], they used the statistical characteristics of correlations between the general service words gathered in a dictionary to classify the given text segments into stego-text segments and normal text segments. These methods have limitations. They only considered the correlations between each word and its surrounding words, and ignored the potential semantic connection between distant words. Our feature extraction module contains convolutional sliding windows of various sizes, which can extract the correlation features of words from different distances in text. Therefore, the proposed model can theoretically obtain better statistical distribution characteristics and achieve higher performance steganalysis.

3.2 Semantic feature extraction by CSW

The steganalysis algorithm based on sliding windows was firstly applied in the field of speech steganalysis [13, 42, 54]. Y. Huang *et al.* [13] used a fixed-length sliding window to slide over the speech stream. Whitin each sliding window, they used the Regular Singular (RS) algorithm [11] to extract features and then determined if LSB steganography has occurred in the speech segment. However, the sliding window they designed had a fixed length, so it could only extract the correlations between each frame and a fixed range of frames. In this paper, in order to extract the correlation features of words with different distances as shown in Fig. 3, we propose a multi-size convolutional sliding window steganalysis method, which has been shown in Fig. 4.

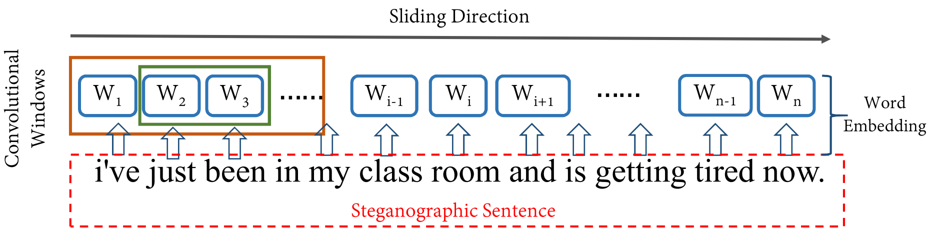


Fig. 4 The words correlation feature extraction model based on multi-size convolutional sliding windows

In recent years, convolutional neural network has made notable progress in fields such as computer vision [15] and natural language processing [14, 48]. A large number of researches and applications have elaborated that convolutional neural network has a powerful ability in feature extractions and expressions [15, 48]. They can represent texts into a dense vector through learning and mapping them into a continuous vector space [16, 28]. In this vector space, semantically similar texts are distributed in the same region, and we can mine the potential semantic relevance between words and sentences [28, 48].

For each input sentence X, we illustrate it with a matrix $X \in \mathbb{R}^{n \times d}$, as shown in Eq. (6), where the i-th row indicates the i-th word in sentence X and each word is represented as a d-dimension vector which is randomly initialized:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,d} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,d} \end{bmatrix} \tag{6}$$

The feature extraction part contains sliding windows of different sizes and each size contains multiple windows. Different sizes of convolutional sliding windows are used to learn the correlations between words in different surrounding neighborhoods. For each size, multiple sliding windows are used to learn multiple dimensional features. The width of each window is the same with the width of the input matrix. Suppose that the height of the k-th window is h, the convolutional sliding window can be expressed as $W^k \in \mathbb{R}^{h \times d}$, that is

$$W^k = \begin{bmatrix} w^k_{1,1} & w^k_{1,2} & \cdots & w^k_{1,d} \\ w^k_{2,1} & w^k_{2,2} & \cdots & w^k_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ w^k_{H,1} & w^k_{H,2} & \cdots & w^k_{h,d} \end{bmatrix}. \tag{7}$$

Convolution operation is a feature extraction process for the elements in the local region of the input matrix. It mainly extracts the correlation features between each word and successive words. For example, when $w^k_{1,1}$ and $a_{1,\,1}$ coincide, the feature c^k_1 extracted from $x_{1\,:\,h}$ by the convolutional window can be:

$$c^k_1 = f\left(\sum_{i=1}^{h}\sum_{j=1}^{d} w^k_{i,j} \cdot a_{i,j} + b^k_{i,j}\right), \tag{8}$$

where the weight $w^k_{i,j}$ denotes the importance of the j-th value in the i-th word vector, $b^k_{i,j}$ is the bias term and f is a nonlinear function. Here we follow previous works [15] and use ReLu function as our nonlinear function, which is defined as

$$y = ReLu(x) = max(0, x). \tag{9}$$

Each window W^k slids from the first word of the inputted sentence to the end of it with a certain step T_c, and calculates the features of each local region. Here it fuses the correlation features between each word and its adjacent words, and further extracts the correlation features between the distant words in the same sentence, namely cross word correlation. Finally, the correlation feature extracted by convolutional windows W^k is:

$$C^k = \left[c_1^k, c_2^k, \ldots, c_{\frac{n-h+1}{T_c}}^k\right]^\top. \tag{10}$$

The pooling layer can reduce the number of neural network parameters while maintaining the overall distribution of the data, which can effectively prevent the model from over-fitting and improve the robustness of the model [15]. The pooling operation is very similar to the convolution operation while the only difference is that it only calculates the average or maximum value of the local area. We conduct a max pooling operation after each convolution operation on the feature C^k. Supposing the height of a pooling kernel is H_p and the step size is T_p, then the output is:

$$M^k = \left[m_1^k, m_2^k, \ldots, m_{N_p}^k\right]^\top, \tag{11}$$

$$m_i^k = max\left(c_i^k, c_{i+1}^k, \ldots, c_{i+H_p-1}^k\right). \tag{12}$$

$$N_p = \frac{\frac{n-h+1}{T_c} - H_p + 1}{T_p}, \tag{13}$$

In the process described above each convolution kernel W^k produces one feature M^k. After all the convolution and pooling operations have been completed, all the extracted features are concatenated end to end to obtain a feature vector of the inputsentence, which can be indicated as

$$E^\top = \left[E_1^\top; E_2^\top; \ldots; E_l^\top\right], \tag{14}$$

where $E_i = E^i$, l indicates the number of the features.

3.3 Feature fusion and steganography determine

The method described above is the processing of feature extraction for each single sentence. But a stegotext may contain more than one sentences: $T = \{X_1, X_2, \ldots, X_m\}$, where X_i is the i-th sentence and m is the number of sentences in the text. For this situation, we compare two corresponding feature fusion methods, which can be called TS-CSW (Single) and TS-CSW (Multi). TS-CSW (Single) concatenates the sentences together first, and then extracts the features in a unified way. TS-CSW (Multi) first extracts the features of each sentence, and then concatenates them together as the whole feature expression. To be more specific, we use $E()$ to represent the word correlation features extracted by the method described inthe preceding section, and F to represent the feature expression of the whole text T. Then, feature fusion of TS-CSW (Single) and TS-CSW (Multi) can be expressed as follows:

$$\begin{array}{ll} TS\text{−}CSW(Single): & F = E([X_1; X_2; \ldots; X_m]), \\ TS\text{−}CSW(Multi): & F = [E(X_1); E(X_2); \ldots; E(X_m)]. \end{array} \tag{15}$$

This design makes TS-CSW (Single) pays more attention to the semantic coherence between sentences, while TS-CSW (Multi) pays more attention to the expression of the internal semantics of each sentence. Taking the poetry carrier as an example, the feature extraction and fusion by TS-CSW (Single) and TS-CSW (Multi) have been shown in Fig. 5.

To avoid ambiguity, here we use F to represent the feature vector extracted by both of these two models. The extracted features collected in F contain all the semantic information of the input text, and we can use them to determine whether confidential information exists in the input text. Following previous works [19, 48] we define a weight matrix W_F and compute the weighted sum of each feature element:

$$y = W_F \cdot F + b_f, \tag{16}$$

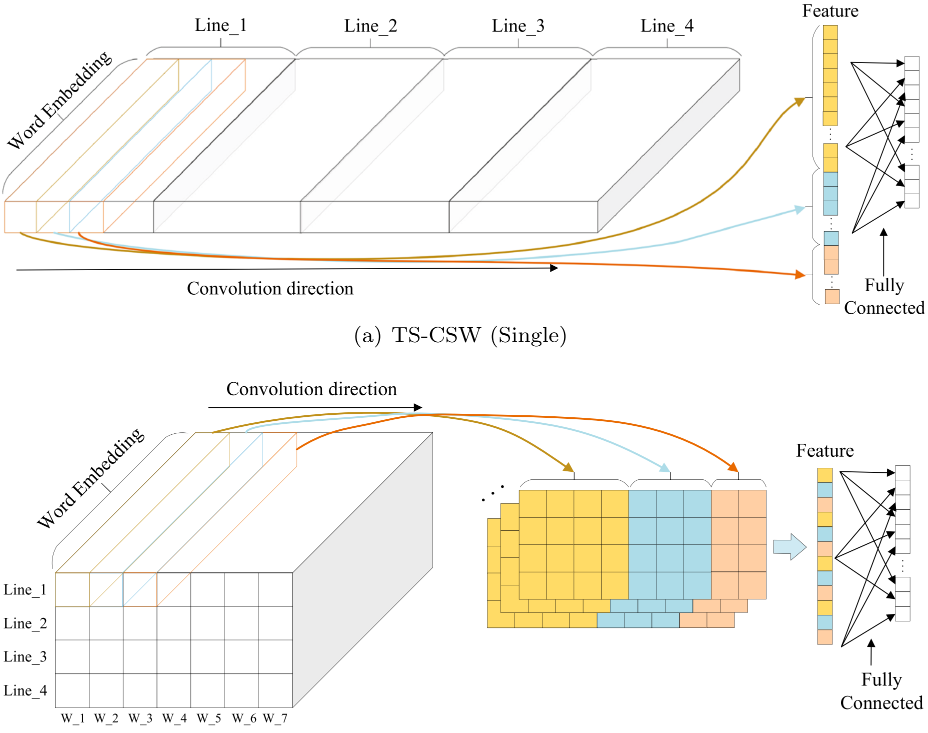
where W_F and b_f are learned weight matrix and bias. The values in weight matrix W_F reflect the importance of each feature. To get normalized output between [0, 1], we send the value through a sigmoid function S:

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{17}$$

and the final output is

$$O = S(y) = S\left(W_F \cdot F + b_f\right). \tag{18}$$

The output value reflects the probability that our model believes that the input text contains confidential messages. We can set a detection threshold and then the final detection result can



(a) TS-CSW (Single)

(b) TS-CSW (Multi)

Fig. 5 The proposed two corresponding models for feature fusion and classification, called TS-CSW (Single) and TS-CSW (Multi)

be expressed as

$$Result = \begin{cases} Stegotext & (O \geq threshold) \\ Covertext & (O < threshold) \end{cases} \tag{19}$$

In the process of training, we update network parameters by applying backpropagation algorithm, and the loss function of the whole network consists of two parts, one is the error term and the other is the regularization term, which can be described as:

$$LOSS = -\frac{1}{N}\sum_{N} T_i \cdot log(O_i) + \|W_F\|_2, \tag{20}$$

where N is the batch size of texts. O_i represents the probability that the i-th sample is judged to contain covert information, T_i is the actual label of the i-th sample. The error term in the loss functioncalculates the average cross entropy between the predicted probability value and the real label. We hope that through self-learning of the model, prediction error can get smaller and smaller, that is, the prediction results are getting closer to the reallabel. In order to strengthen the regularization and prevent overfitting, we adopt the dropout mechanism and a constraint on l2-norms of the weight vectors during the training process. Dropout mechanism means that in training process of deep learning network, the neural network unit is temporarily discarded from the network, i.e. set to zero, according to a certain probability. This mechanism has been proved to effectively prevent neural network from overfitting, and significantly improve the model's performance [15].

4 Experiments and analysis

In this section, we designed several experiments to test the proposed model. We will start with introducing the dataset collected and used in this work, then we will introduce the model structure and experimental parameter settings, finally we will present the experiments results and discuss them.

4.1 Dataset collection

Since currently there is no public text steganalysis dataset available, in order to train the proposed model and promote the development of related fields, we construct a text steganalysis dataset named as T-Steg, which is available in the Github repository[1]. The steganographic texts in T-Steg are mainly divided into two types and are generated by the latest steganography methods, including special format steganographic texts, generated by the text steganography algorithm proposed in [21], and natural steganographic texts, which are generated by the method proposed in [9]. In order to facilitate the relevant researchers to use T-Steg dataset, we will briefly introduce these two text steganography methods, and further introduce the construction process of T-Steg dataset.

These two models essentially use the Recurrent Neural Networks (RNNs) to learn the statistical language models of a large number of normal texts, and then generate steganographic text based on

[1] https://github.com/YangzlTHU/TS-CNN

the learned statistical language model. Method proposed in teLuo2017Text further fuses some syntactic rule information to generate traditional Chinese ancient poetry. The advantage of this method is that the generated steganographic texts are consistent with the normal samples both statistically and syntactically, thus further enhancing the concealment. Chinese ancient poems can be divided into two forms. One is with five words per line (FW), and the other is with seven (SW). Each format can be further divided into two categories, that is, each poem contains four lines (FL) or eight lines (EL). Since classical Chinese quatrains have strong semantic relevance between adjacent lines, they use an attention-based bidirectional encoder-decoder model to build the line-to-line module. The encoder first maps the input sentence to a feature vector which contains the semantic meaning of each character of the input, then the decoder calculates and generates the next sentence based on the feature vector of this input sentence. For sentence generation, they use another RNN to calculate the probability distribution of each word, and then select the words that match the tone pattern to form a candidate word list. Then they encode the words in the candidate word list, and select the corresponding words according to the code stream that needs to be embedded, so as to generate semantically coherent and structurally correct ancient poems while embedding secret information.

The model proposed in [9] also used the Recurrent Neural Networks to build a text steganography system. They first trained the RNNs with a large amount of normal text to obtain a good statistical language model, and then used the trained statistical language model to generate steganographic texts. For information hiding, they divided the dictionary in advance and fixed the code for each word. Then in the generation process, the most appropriate word in the corresponding subset was selected as the output according to the code stream. We reproduced their model and chose three large-scale text datasets containing the most common text media on the Internet to train it, which were Twitter [12], movie reviews [23] and News [39].

Both two steganography models can adjust the embedding rate by adjusting the size of candidate list for each word, that is, the number of bits embedded in each word. We have generated 10,000 texts for various types of text at different embedding rates, except for poetry text without covert information, which were only 4,000. Because when the generated poetry does not contain covert information, the candidate list for each word in the generation process is 1, so the number of poems that can be generated islimited. Since the original paper [21] only provides 4000 initial states, so only 4,000 poems can be generated. Therefore, our T-Steg dataset has a total number of 396,000 texts including various embedding rates under various formats, whichhave been shown in Table 1.

Table 1 The collected text steganalysis (T-Steg) dataset, which contains a total number of 396,000 texts with various embedding rates under various formats, including special format texts generated by [21] and natural texts generated by [9]

Bit (s) / word (*bpw*)			0	1	2	3	4	5	Total
Special Format (Chinese) [21]	FW	FL	4,000	10,000	10,000	10,000	10,000	10,000	54,000
		EL	4,000	10,000	10,000	10,000	10,000	10,000	54,000
	SW	FL	4,000	10,000	10,000	10,000	10,000	10,000	54,000
		EL	4,000	10,000	10,000	10,000	10,000	10,000	54,000
Natural Text (English) [9]	News [39]		10,000	10,000	10,000	10,000	10,000	10,000	60,000
	IMDB [23]		10,000	10,000	10,000	10,000	10,000	10,000	60,000
	Twitter [12]		10,000	10,000	10,000	10,000	10,000	10,000	60,000
Total			46,000	70,000	70,000	70,000	70,000	70,000	396,000

4.2 Experimental setting and training details

Almost all the parameters in our model can be obtained through training, but there are still some hyper-parameters that need to be determined. To determine these hyper-parameters, we designed multiple sets of comparative experiments. Finally, to synthesize the performance of all aspects of the model, the hyper-parameters of our model were setted as follows. For embedding layer, we mapped each word to a 400-dimensions vector ($d = 400$ in Formula (6)), which was randomly initialized. The width of the convolutional windows was 400 which was the same as the input matrix. However, the height of the sliding window (h in Formula (7)) was not fixed. We compared the results of different window sizes and finally setted window sizes as follows: for longer text, such as generated steganographic news and movie reviews, we setted the window height to be 3, 4, 5; for shorter text, such as generated steganographic twitter and potries, we setted the window height to be 1, 2, 3, and each of the different heights had 128 convolution kernels. The dimension of the feature extracted for each text inputted is $3 \times 128 = 384$ (length of F in Formula (15)). We setted the size of fully connected layer to be 100 (length of y in Formula (16)) and detection threshold to be 0.5.

We trained our model by minimizing the *LOSS* function over a batch size number of samples. We used stochastic gradient descent with momentum 0.9 to train the parameters of our network. The update rule for weight w is:

$$w_{i+1} = w_i + \alpha \cdot V_i - \lambda \cdot < \frac{\partial L}{\partial w_i} > \Big|_{D_i}, \tag{21}$$

where i is the iteration idex, $\alpha \in (0, 1]$ is the momentum factor, V is the momentum variable, λ is the learning rate, and $< \frac{\partial L}{\partial w_i} > \Big|_{D_i}$ is the average over the i-th batch D_i of the derivative of the *LOSS* function with respect to w, evaluated at w_i.

To train and test the proposed model, for natural texts, we randomly selected about 80% of the samples for training, 20% of the samples for testing. For poetries, in order to keep the number ratio of positive samples to negative samples 1 to 1, we randomly picked up 3,500 positive and 3,500 negative samples from Nostego-texts (bpw=0) and Stego-texts with different embedding rate for training, and then randomly picked up another 500 samples from each dataset for model testing.

Our model can quickly converge during the training processing, with less than 30 epochs (one epoch means that all the training samples finish one training session), and can reach a steady state with high accuracy and a very smooth loss curve. Figure 6 shows the training process of TS-CSW (Single) on SW-FL set.

4.3 Evaluation results and discussion

4.3.1 Steganalysis efficiency

We first tested the efficiency of the proposed model for text steganalysis, which is the time it took to analyze a piece of text that may contain covert information. Results in Table 2 validate the efficiency of our model for steganalysis. From the results we can see, as the sample length increases, the required steganalysis time gradually increases. Even so, for the longest text, namely samples in the SW-EL set, which contains 56 words per sample, it only takes an

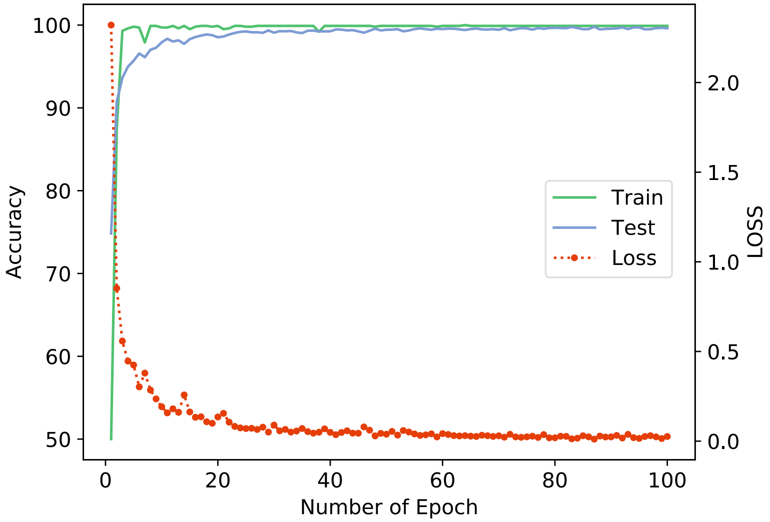


Fig. 6 The processing of training, including the accuracy of train/test set and the loss of the training set varies with the number of epochs

average time of 9.78/9.83 ms for TS-CSW (Single) and TS-CSW (Multi). These results show that the proposed model has a very high steganalysis efficiency and can implement almost real-time steganographic detection analysis of these texts

4.3.2 Steganalysis accuracy

In order to objectively reflect the performance of the proposed model, in this section, we choose three representative text steganalysis algorithms, which are proposed in [8, 27, 32], respectively. Authors in [27] used the trained language model to calculate the perplexity of each input sample text, and used them as the statistical features to determine whether they contain covert information. Samanta *et al.* tesamanta2016real proposed statistical text steganalysis tools based on Bayesian Estimation and Correlation Coefficient methodologies. Din *et al.* [8] proposed a formalization of genetic algorithm method in order to detect steganographic text.

We used several evaluation indicators commonly used in classification tasks to evaluate the performance of our model, which are precision, recall and accuracy. The conceptions and formulas are described as follows:

Table 2 The average prediction time (ms) for each sample in test set

Dataset	FW		SW		News	IMDB	Twitter
	FL	EL	FL	EL			
TS-CSW (Single)	4.81±1.16	7.48±1.41	6.20±1.30	9.78±1.46	3.35±0.72	3.34±0.57	1.26±0.70
TS-CSW (Multi)	4.76±0.56	7.50±0.65	6.52±0.62	9.83±0.98	-	-	-

- Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}. \tag{22}$$

- Precision measures the proportion of positive samples in the classified samples.

$$Precision = \frac{TP}{TP + FP}. \tag{23}$$

- Recall measures the proportion of positives that are correctly identified as such.

$$Recall = \frac{TP}{TP + FN}. \tag{24}$$

TP (True Positive) represents the number of positive samples that are predicted to be positive by the model, FP (False Positive) indicates the number of negative samples predicted to be positive, FN (False Negative) illustrates the number of positive samples predicted to be negative and TN (True Negative) represents the number of negative samples predicted to be negative. Experiment results have been shown in Tables 3 and 4.

According to the results, we can draw the following conclusions. Firstly, compared to other text steganalysis methods, the proposed models, both TS-CSW (Single) and TS-CSW (Multi), have achieved the best detection results on various metrics, including different text format and different embedding rates. Especially in the case of low embedding rate, for example, when the number of bit embedded in per word (bpw) is 1, our model has more obvious detection performance advantages over other models. We have also plotted the ROC curves of each model on SW-FL set when $bpw = 1$, which is shown in Fig. 7. Combined with these test results, we can find that our model has significantly practical value.

Secondly, in Tables 3 and 4, we notice that under various datasets, the detection performance of each model has improved with the increase of the embedding rate. These results do meet our previous conjecture that with the increaseof embedding rate, it will damage the coherence of text semantics, that is, the semantic relevance of the words. Therefore it is easier to be distinguished from the normal texts. Figure 8 shows the change of steganalysis performance of the TS-CSW (Single) on the poetry dataset with the increase of embedding rate.

Thirdly, we notice that for the proposed models, the text length seems to be a significant performance impact factor. We can compare the results on EL set with those on FL set under each embedding rate, since the length of each sample in EL is twice as long as that of FL. For the TS-CSW (Single), most of the test results on EL are better than those on FL. But for TS-CSW (Multi), the results are the opposite, that is, most of the test results on FL are better than

Table 3 Results of different steganalysis methods

Format		bpw	[27]			[32]			[8]			TS-CSW (Single)			TS-CSW (Multi)		
Metric			Acc	P	R	Acc	P	R	Acc	P	R	Acc	P	R	Acc	P	R
FW	FL	1	0.521	0.597	0.518	0.768	0.775	0.768	0.724	0.706	0.766	0.907	0.931	0.872	0.929	1.000	0.877
		2	0.600	0.671	0.588	0.868	0.877	0.868	0.843	0.840	0.848	0.955	0.921	0.959	0.969	1.000	0.877
		3	0.681	0.747	0.659	0.884	0.896	0.884	0.895	0.896	0.894	0.970	0.942	0.960	0.973	0.978	1.000
		4	0.705	0.769	0.681	0.872	0.885	0.861	0.899	0.892	0.908	0.990	1.000	1.000	0.981	0.979	0.979
		5	0.798	0.858	0.765	0.868	0.893	0.864	0.933	0.930	0.936	0.990	0.962	1.000	0.986	1.000	0.958
	EL	1	0.515	0.598	0.513	0.775	0.775	0.775	0.750	0.778	0.700	0.815	0.869	0.769	0.858	0.969	0.718
		2	0.592	0.671	0.579	0.917	0.956	0.897	0.904	0.902	0.906	0.986	1.000	0.980	0.960	0.956	0.897
		3	0.675	0.742	0.654	0.918	0.924	0.918	0.944	0.932	0.958	0.985	0.952	0.975	0.961	1.000	0.901
		4	0.712	0.772	0.689	0.915	0.948	0.915	0.961	0.964	0.958	0.999	1.000	1.000	0.975	0.963	1.000
		5	0.819	0.869	0.789	0.911	0.923	0.901	0.970	0.974	0.966	0.996	1.000	1.000	0.987	0.981	0.981
SW	FL	1	0.539	0.616	0.533	0.666	0.667	0.666	0.710	0.703	0.726	0.922	0.978	0.920	0.917	1.000	0.800
		2	0.619	0.691	0.604	0.898	0.901	0.898	0.918	0.927	0.908	0.980	0.962	0.927	0.981	0.979	0.942
		3	0.691	0.762	0.667	0.928	0.931	0.941	0.954	0.960	0.948	0.984	0.979	0.979	0.991	1.000	0.941
		4	0.730	0.796	0.703	0.925	0.930	0.898	0.973	0.972	0.974	0.997	1.000	1.000	0.991	1.000	0.979
		5	0.810	0.865	0.779	0.916	0.927	0.913	0.985	0.986	0.984	0.995	1.000	0.976	0.995	1.000	0.981
	EL	1	0.523	0.624	0.519	0.656	0.656	0.656	0.675	0.659	0.724	0.861	0.956	0.745	0.893	0.974	0.796
		2	0.589	0.682	0.575	0.926	0.927	0.926	0.940	0.956	0.922	0.993	1.000	0.981	0.972	0.960	0.941
		3	0.651	0.721	0.633	0.942	0.942	0.942	0.950	0.957	0.942	0.989	1.000	1.000	0.971	1.000	0.962
		4	0.696	0.754	0.675	0.958	0.959	0.958	0.978	0.982	0.974	0.995	1.000	1.000	0.988	0.977	1.000
		5	0.793	0.829	0.774	0.954	0.957	0.952	0.984	0.984	0.984	0.999	1.000	1.000	0.995	0.960	1.000

Table 4 Results of different steganalysis methods

Method			[27]			[32]			[8]			TS-CSW (Single)		
Format		bpw	Acc	P	R	Acc	P	R	Acc	P	R	Acc	P	R
Natural Text [9]	News [39]	1	0.532	0.517	0.382	0.763	0.739	0.812	0.840	0.869	0.801	0.904	0.935	0.868
		2	0.513	0.535	0.204	0.786	0.762	0.832	0.835	0.867	0.791	0.896	0.922	0.866
		3	0.597	0.679	0.367	0.824	0.767	0.931	0.897	0.909	0.882	0.965	0.975	0.955
		4	0.755	0.831	0.640	0.859	0.797	0.962	0.938	0.962	0.911	0.971	0.990	0.953
		5	0.847	0.918	0.761	0.881	0.829	0.959	0.961	0.976	0.945	0.979	0.986	0.971
	IMDB [23]	1	0.577	0.642	0.345	0.767	0.779	0.744	0.787	0.829	0.722	0.878	0.930	0.817
		2	0.713	0.807	0.560	0.849	0.934	0.871	0.869	0.911	0.818	0.950	0.975	0.924
		3	0.840	0.925	0.741	0.900	0.877	0.931	0.916	0.944	0.885	0.961	0.977	0.945
		4	0.909	0.969	0.845	0.937	0.905	0.975	0.962	0.975	0.947	0.983	0.989	0.977
		5	0.909	0.989	0.828	0.929	0.921	0.940	0.977	0.987	0.966	0.995	0.996	0.993
	Twitter [12]	1	0.538	0.520	0.387	0.654	0.652	0.658	0.665	0.664	0.670	0.780	0.871	0.657
		2	0.544	0.523	0.399	0.745	0.762	0.712	0.750	0.827	0.631	0.826	0.892	0.743
		3	0.577	0.669	0.303	0.809	0.798	0.826	0.834	0.889	0.764	0.920	0.966	0.871
		4	0.729	0.836	0.570	0.842	0.824	0.871	0.885	0.950	0.813	0.942	0.984	0.898
		5	0.850	0.916	0.770	0.851	0.839	0.870	0.899	0.961	0.832	0.946	0.987	0.902

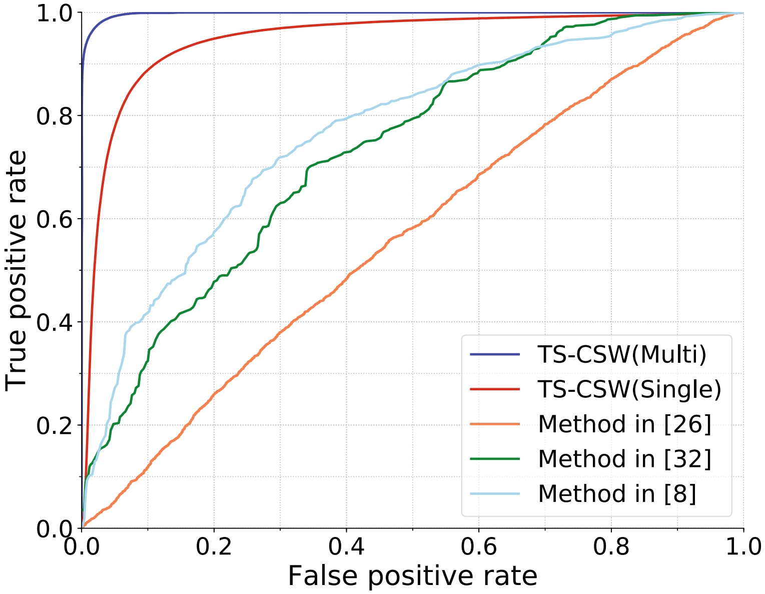


Fig. 7 The ROC curves of each model on SW-FL set when $bpw = 1$

those on EL, especially at low embedding rates. Similarly, for the same pattern (FL or EL), the text length of the SW sample is longer than that of the FW. If we compare the results of FL and EL in FW and SW respectively, we can also draw the same conclusion, that is, TS-CSW (Single) is more suitable for the detection of longer texts, while TS-CSW (Multi) is more suitable for the detection of shorter texts. We analyze that the reason for this phenomenon is

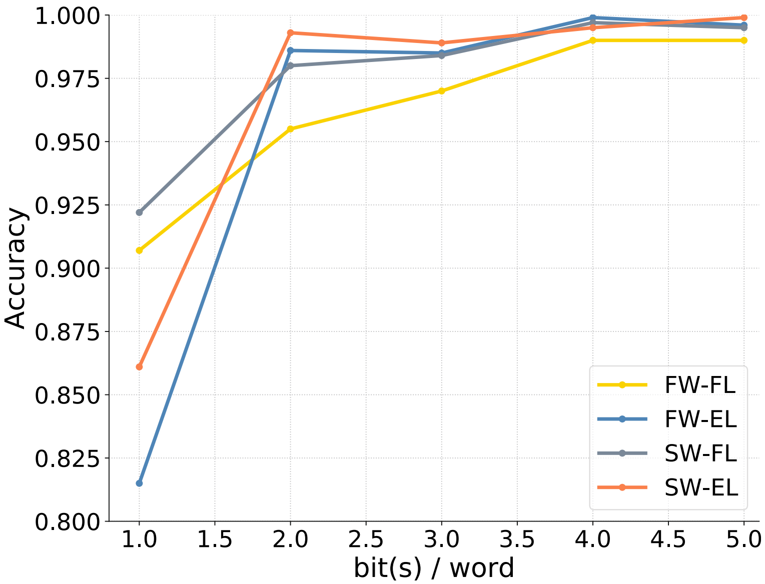


Fig. 8 The change of the steganalysis performance of the TS-CSW (Single) on the poetry dataset with the increase of embedding rate

that these two models focus on different correlations between words. While TS-CSW (Multi) extracts features from each sentence in the poems, it focuses on the correlation between words in a single sentence, which includes successive word correlations and cross word correlations. TS-CSW (Single) treats the entire poem as a whole and then performs semantic extraction. It pays more attention to the the relevance between sentences, which are cross sentence correlations. When the embedding rate of covert information in the generated poems is low, it first affects the correlation between words in a single sentence, but it still maintains a strong correlation between different sentences. Therefore, the changes in the feature space will first be detected by an TS-CSW (Multi). As the embedding rate increases, it gradually begins to affect the correlation between sentences. This extra semantic change will be detected by TS-CSW (Single), and therefore exhibits a higher performance detection.

Finally, when we compare the results of natural steganographic text, we find that at the same embedding rate, the detection accuracy of generated news and movie comment is better than the generated twitter text. The reason might be that Twitter text is not a serious text, and it tends to be colloquial. Therefore, the Twitter texts have a large degree of irregularity and a large variance in quality. Therefore, even if we embed information inside, its damage to the quality of the text will be more difficultto detect. However, it is worth noting that even in this case, the detection accuracy of our model is still higher than other models, and in most cases it can reach higher than 90%.

4.4 Embedded rate estimation

As we have mentioned before, our model can automatically extract the correlations between words in the text and map them to a high-dimensional feature space. We can use t-Distributed Stochastic Neighbor Embedding (t-SNE) [24] technique for the dimensionality reduction and visualization of this feature space, which can be found in Fig. 9. In this feature space, each point represents a poem containing hidden information with different embedding rates, and different colors indicate different embedding rate poems.

From the Fig. 9, firstly, we find that points with the same embedding rate are clustered in the same area, indicating that our model accurately detects the subtle differences in the semantics of these texts under different information embedding rates. Secondly, we can clearly see that as the embedded rate of hidden information increases in the generated texts, their distribution in the semantic space will gradually change and migrate, from the red area (which contains no hidden information) to the purple area ($bpw = 5$). When the embedding rate is low, such as $bpw = 1$, the area formed by the corresponding yellow dots and the red dots (containing no hidden information) still have some overlaps, but the change of the center of gravity can beclearly seen. When the embedding rate is high, for example, the area formed by the points of dark blue and purple dots have very clear boundaries with the red dots area.

The results in Fig. 9 fully demonstrate our model's ability to extract the correlations between words in the text, as well as the ability to implement text steganalysis using subtle differences of these correlations' distributions under different information embedding rates. This is also the core point of this paper, that is, after embedding hidden information in texts, it will damage the correlations between words in the text, and if we can find a suitable method (such as the TS-CSW method proposed in this paper) to extract these correlation features, then we can use these subtle differences in the feature space to achieve effective steganalysis.

Further, besides steganographically determined, we find our model can even make use of the distribution of texts in feature space to estimate the capacity of hidden information inside.

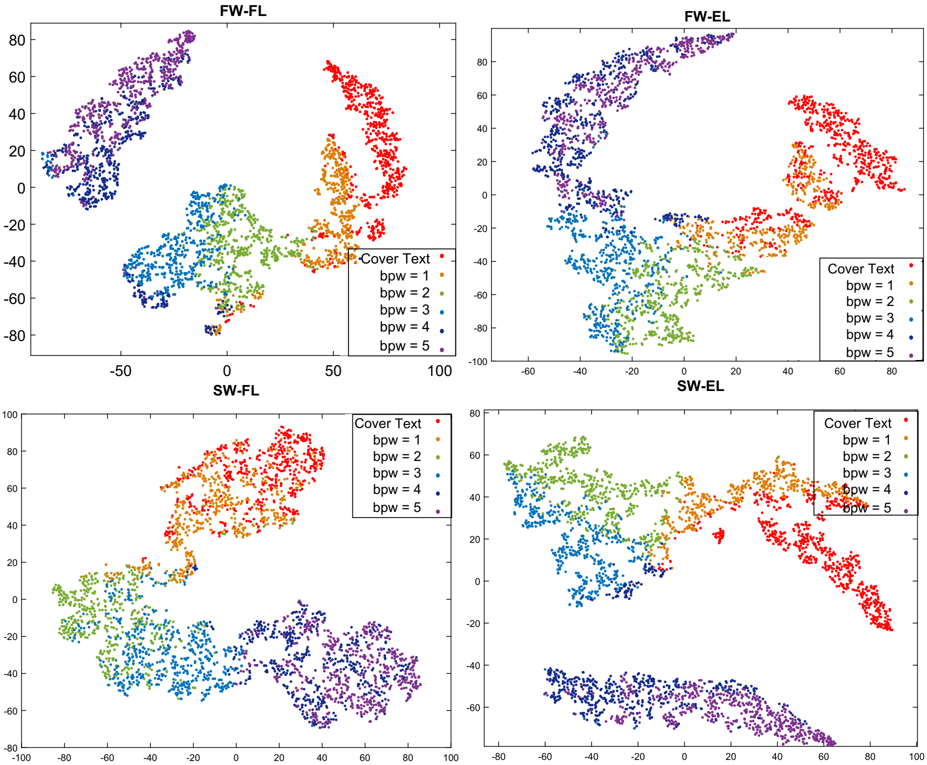


Fig. 9 The distribution of poetries under different information embedding rates in the feature space. Each point represents a poem containing hidden information with different *bpw* from 0 to 5, and different colors indicate different embedding rate poems. We can clearly see that as the embedded rate of hidden information increases in the generated texts, their distribution in the semantic space will gradually change and migrate

We mixed the steganographic texts of different embedding rates and then used the proposed model and other previous models for multi-classification. We want to know if the proposed model can accurately estimate the capacity of concealed information in steganographic text. The experimental results are shown in Table 5. From Table 5, we can see that our model can

Table 5 The results of the proposed models' estimate of the capacity of the covert information in texts

Model		FW		SW		News	IMDB	Twitter
		FL	EL	FL	EL			
Method in [27]	P	0.258	0.266	0.261	0.246	0.445	0.490	0.417
	R	0.297	0.311	0.292	0.286	0.396	0.512	0.363
Method in [32]	P	0.465	0.510	0.493	0.540	0.701	0.742	0.620
	R	0.473	0.511	0.492	0.521	0.396	0.512	0.363
Method in [8]	P	0.465	0.513	0.496	0.568	0.745	0.767	0.638
	R	0.472	0.515	0.501	0.568	0.741	0.760	0.615
TS-CSW (Single)	P	0.711	0.744	0.724	0.772	0.803	0.849	0.741
	R	0.708	0.743	0.735	0.769	0.799	0.844	0.705
TS-CSW (Multi)	P	0.751	0.724	0.737	0.718	-	-	-
	R	0.749	0.718	0.734	0.712	-	-	-

achieve an estimated accuracy higher than 70% for the hidden information in the text, which outperforms all the other models. Although there is still much room for improving these results, but our results point out another possible direction for future text steganalysis research, that is, besides simply judging whether the text contains covert information or not, we can further try to estimate how many secret information contained in steganographic texts. This will play a more active role in maintaining the security of cyberspace.

5 Conclusion

In this paper, we propose a new text steganalysis method. We analyzed the correlation between words in these generated steganographic texts. To extract those correlation features, we propose the words correlation extraction model, which is based on convolutional sliding windows (CSW). We find that after embedding the secret information, there exists a subtle distribution difference of the features space. Then we propose the feature classification model to classify those correlation features into cover text and stego text categories. To train and test the proposed model, we collected and released a large text steganalysis (T-Steg) dataset, which contains a total number of 396,000 texts with various embedding rates under various formats including special format and natural texts. Experimental results show that the proposed model achieves nearly 100% precision and recall, outperforms all previous methods. Besides, our model can even make use of the subtle distribution difference of the features to estimate the capacity of the hidden information inside, and the estimated accuracy rate is above 70%. We hope that this paper will serve as a reference guide for researchers to facilitate the design and implementation of better text steganalysis.

Acknowledgements The authors thank Dr. Zhiwen Hu for serving as scientific advisors to this research and thank Qi Li for participating in writing of the manuscript. This research is supported by the National Key R&D Program (2018YFB0804103) and the National Natural Science Foundation of China (No.U1536207, No.U1636113 and No.61862002).

References

1. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. Comput Sci
2. Bitar AW, Darazi R, Couchot JF, Couturier R (2017) Blind digital watermarking in pdf documents using spread transform dither modulation. Multimedia Tools and Applications 76(1):143–161
3. Cachin C (2004) An information-theoretic model for steganography. Inf Comput 192(1):41–56
4. Chapman M, Davida G (1997) Hiding the hidden: A software system for concealing ciphertext as innocuous text. In: International Conference on Information and Communications Security, pp. 335–345. Springer
5. Chen Z, Huang L, Yu Z, Yang W, Li L, Zheng X, Zhao X (2008) Linguistic steganography detection using statistical characteristics of correlations between words. In: International Workshop on Information Hiding, pp. 224–235. Springer
6. Dai W, Yu Y, Deng B (2009) Bintext steganography based on markov state transferring probability. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1306–1311. ACM
7. Dai W, Yu Y, Dai Y, Deng B (2010) Text steganography system using markov chain source model and des algorithm. JSW 5(7):785–792
8. Din R, Yusof SAM, Amphawan A, Hussain HS, Yaacob H, Jamaludin N, Samsudin A (2015) Performance analysis on text steganalysis method using a computational intelligence approach. In: Proceeding of

International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015), Palembang, Indonesia, pp. 19–20
9. Fang T, Jaggi M, Argyraki K (2017) Generating steganographic text with lstms. arXiv preprint arXiv:1705.10742
10. Fridrich J (2009) Steganography in digital media: principles, algorithms, and applications. Cambridge University Press
11. Fridrich J, Goljan M, Du R (2001) Detecting lsb steganography in color, and gray-scale images. IEEE multimedia 8(4):22–28
12. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1(12)
13. Huang Y, Tang S, Zhang Y (2011) Detection of covert voice-over internet protocol communications using sliding window-based steganalysis. IET communications 5(7):929–936
14. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882
15. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105
16. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196
17. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
18. Lifeng S, Zhengdong L, Hang L (2015) Neural responding machine for short-text conversation pp. 52–58
19. Lin Z, Huang Y, Wang J (2018) Rnn-sm: Fast steganalysis of voip streams using recurrent neural network. IEEE Transactions on Information Forensics & Security PP(99), 1–1
20. Liu Y, Sun X, Gan C, Hong W (2007) An efficient linguistic steganography for chinese text. In: IEEE International Conference on Multimedia & Expo
21. Luo Y, Huang Y (2017) Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry. In: ACM Workshop on Information Hiding and Multimedia Security, pp. 99–104
22. Luo Y, Huang Y, Li F, Chang C (2016) Text steganography based on ci-poetry generation using markov chain model. Ksii Transactions on Internet & Information Systems 10(9):4568–4584
23. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1, pp. 142–150. Association for Computational Linguistics
24. Maaten LVD (2014) Accelerating t-SNE using tree-based algorithms. JMLR.org
25. Mahato S, Khan DA, Yadav DK (2017) A modified approach to data hiding in microsoft word documents by change-tracking technique. Journal of King Saud University - Computer and Information Sciences
26. Meng YY, Gao BJ, Yuan Q, Yu FG, Wang CF (2008) A novel steganalysis of data hiding in binary text images. In, IEEE Singapore International Conference on Communication Systems
27. Meng P, Hang L, Yang W, Chen Z, Zheng H (2009) Linguistic Steganography Detection Algorithm Using Statistical Language Model. IEEE Computer Society
28. Mikolov T, Yih WT, Zweig G (2013) Linguistic regularities in continuous space word representations. In HLT-NAACL
29. Moraldo HH (2014) An approach for text steganography based on markov chains. arXiv preprint arXiv:1409.0915
30. Murphy B, Vogel C (2007) The syntax of concealment: reliable methods for plain text information hiding. Proc Spie
31. Odeh A, Elleithy K, Faezipour M (2014) Steganography in text by using ms word symbols. In, American Society for Engineering Education
32. Samanta S, Dutta S, Sanyal G (2016) A real time text steganalysis by using statistical method. In: Engineering and Technology (ICETECH), 2016 IEEE International Conference on, pp. 264–268. IEEE
33. Schmidhuber J (2015) Deep learning in neural networks: An overview. Neural Netw 61:85–117
34. Shannon CE (1949) Communication theory of secrecy systems. Bell Labs Technical Journal 28(4):656–715
35. Shirali-Shahreza MH, Shirali-Shahreza M (2008) A new synonym text steganography. In: Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on, pp. 1524–1526. IEEE
36. Shniperov A, Nikitina K (2016) A text steganography method based on markov chains. Autom Control Comput Sci 50(8):802–808
37. Simmons GJ (1984) The prisoners' problem and the subliminal channel. Advances in Cryptology Proc Crypto pp. 51–67
38. Taskiran CM, Topkara M, Delp EJ (2006) Attacks on lexical natural language steganography systems. Proceedings of SPIE - The International Society for Optical Engineering 6072:607209–607209–9
39. Thompson A (2017) Kaggle. https://www.kaggle.com/snapcrack/all-the-news/data

40. Wayner P (1992) Mimic functions. Cryptologia 16(3):193–214
41. Xiang L, Sun X, Gang L, Gan C (2007) Research on steganalysis for text steganography based on font format. In: International Symposium on Information Assurance & Security
42. Xie C, Cheng Y, Chen Y (2011) An active steganalysis approach for echo hiding based on sliding windowed cepstrum. Signal Processing 91(4):877–889
43. Yang H, Cao X (2010) Linguistic steganalysis based on meta features and immune mechanism. Chinese Journal of Electronics 19(4):661–666
44. Yang C, Liu F, Luo X, Liu B (2008) Steganalysis frameworks of embedding in multiple least-significant bits. IEEE Transactions on Information Forensics and Security 3(4):662–672
45. Yang Z, Peng X, Huang Y (2017) A sudoku matrix-based method of pitch period steganography in low-rate speech coding. In: International Conference on Security and Privacy in Communication Systems, pp. 752–762. Springer
46. Yang Z, Zhang YJ, ur Rehman S, Huang Y (2017) Image captioning with object detection and localization. In: International Conference on Image and Graphics, pp. 109–118. Springer
47. Yang Z, Du X, Tan Y, Huang Y, Zhang YJ (2018) Aag-stega: Automatic audio generation-based steganography. arXiv preprint arXiv:1809.03463
48. Yang Z, Huang Y, Jiang Y, Sun Y, Zhang YJ, Luo P (2018) Clinical assistant diagnosis for electronic medical record based on convolutional neural network. Scientific reports 8(1):6329
49. Yang Z, Zhang P, Jiang M, Huang Y, Zhang YJ (2018) Rits: Real-time interactive text steganography based on automatic dialogue model. In: International Conference on Cloud Computing and Security, pp. 253–264. Springer
50. Yang ZL, Jin S, Huang YF, Zhang YJ, Li H (2018) Automatically generate steganographic text based on markov model and huffman coding. arXiv preprint arXiv:1811.04720
51. Yang Z, Hu Y, Huang Y, Zhang Y (2019) Behavioral security in covert communication systems. arXiv preprint arXiv:1910.09759
52. Yang Z, Huang Y, Zhang YJ (2019) A fast and efficient text steganalysis method. IEEE Signal Processing Letters pp. 1–1
53. Yang Z, Wang K, Ma S, Huang Y, Kang X, Zhao X (2019) Istego100k: Large-scale image steganalysis dataset. arXiv preprint arXiv:1911.05542
54. Yang Z, Yang H, Hu Y, Huang Y, Zhang YJ (2019) Real-time steganalysis for stream media based on multichannel convolutional sliding windows. arXiv preprint arXiv:1902.01286
55. Yang ZL, Guo XQ, Chen ZM, Huang YF, Zhang YJ (2019) Rnn-stega: Linguistic steganography based on recurrent neural networks. IEEE Transactions on Information Forensics and Security 14(5):1280–1295
56. Yuling L, Xingming S, Can G, Hong W (2007) An efficient linguistic steganography for chinese text. In: 2007 IEEE International Conference on Multimedia and Expo, pp. 2094–2097. IEEE
57. Zhang J, Shen J, Wang L, Lin H (2016) Coverless text information hiding method based on the word rank map. In: International Conference on Cloud Computing and Security, pp. 145–155
58. Zhou Z, Mu Y, Wu QJ (2018) Coverless image steganography using partial-duplicate image retrieval. Soft Computing pp. 1–12

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