Chinese Journal of Electronics
Vol.32, No.1, Jan. 2023

Linguistic Steganalysis via Fusing Multi-Granularity Attentional Text Features

WEN Juan, DENG Yaqian, PENG Wanli, and XUE Yiming

(*College of Information and Electrical Engineering, China Agricultural University, Beijing 100094, China*)

Abstract — Deep learning based language models have improved generation-based linguistic steganography, posing a huge challenge for linguistic steganalysis. The existing neural-network-based linguistic steganalysis methods are incompetent to deal with complicated text because they only extract single-granularity features such as global or local text features. To fuse multi-granularity text features, we present a novel linguistic steganalysis method based on attentional bidirectional long-short-term-memory (BiLSTM) and short-cut dense convolutional neural network (CNN). The BiLSTM equipped with the scaled dot-product attention mechanism is used to capture the long dependency representations of the input sentence. The CNN with the short-cut and dense connection is exploited to extract sufficient local semantic features from the word embedding matrix. We connect two structures in parallel, concatenate the long dependency representations and the local semantic features, and classify the stego and cover texts. The results of comparative experiments demonstrate that the proposed method is superior to the state-of-the-art linguistic steganalysis.

Key words — Information hiding, Natural language processing, Linguistic steganalysis, Attentional BiLSTM, Dense connection.

I. Introduction

Steganography is a vital branch of information hiding and an integral part of the concealment system [1]. It mainly conceals secret information within the digital carriers such as images [2], [3], videos [4], [5], and texts [6].

In recent years, linguistic steganography has achieved remarkable improvement [7]–[9]. On the contrary, linguistic steganalysis aims to identify the existence of the embedding trace and has also achieved significant progress. In the early stage, these methods were based on the combination of hand-crafted features and classifiers [10], [11], which limited the detection capability and universality of steganalysis algorithms [12]. While in the current stage, studies mainly rely on end-to-end neural networks (NNs) to extract text features [12]–[18]. However, the semantic features extracted by these NNs-based methods are insufficient. That is because these kinds of methods do not fully integrate the semantic information of sentences with multiple granularities. They either pay attention to the high-level information of the output layer or give too much consideration to the shallow text semantics, which restricts their detection performance.

To address the above-mentioned limitation and further improve the detection performance, we propose an effective NNs-based linguistic steganalysis scheme to extract text features at multiple granularities. On the one hand, inspired by the attention mechanism [19], we construct a long dependency representations extractor by the bidirectional long-short-term-memory (BiLSTM) equipped with the scaled dot-product attention module. On the other hand, motivated by the short-cut [20] and dense connection [21], we propose a short-cut dense convolutional neural network (CNN) structure named SDC (short-cut dense CNN) to directly capture extra local semantic features from the word embedding matrix. Two structures are connected in parallel to form an entire structure, namely BiLSTM-SDC. This structure can impose helpful information on the learned representation and force the model to learn better text representations from both shallow and deep layers, hence improving detection accuracy. The experimental results demonstrate that the proposed linguistic steganalysis scheme outperforms the state-of-the-art NNs-based methods.

The main contributions of this paper are highlighted as follows:

Manuscript Received Jan. 17, 2022; Accepted May 13, 2022. This work was supported by the National Natural Science Foundation of China (61872368, 61802410).
 DOI:10.23919/cje.2022.00.009

1) We propose an effective NNs-based linguistic steganalysis model, namely BiLSTM-SDC, for fully extracting comprehensive feature representations by merging multi-granularity attentional text features via two parallel modules.

2) We construct a structure that enables parallel extraction of text representations. On the one hand, we employ the scaled dot-product attention to learn the different weights and design an attentional BiLSTM to optimize the long dependency representations. On the other hand, we propose a short-cut dense CNN structure parallel to the attentional BiLSTM to capture sufficient local features of the inputs.

3) We conduct extensive experiments on four datasets, and demonstrate the superiority of the proposed model.

The remaining of this paper is structured as follows: Section II summarizes relevant work in the fields of linguistic steganography and linguistic steganalysis. Section III describes in detail the proposed BiLSTM-SDC scheme. We present the experimental settings and report the results in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

In this section, we briefly review the most widely used approaches in the linguistic steganography and steganalysis fields.

1. Linguistic steganography

Linguistic steganography has been an active research area for decades, which is typically grouped into modification-based methods and generation-based methods. The modification-based methods embed the secret information mostly by substituting synonyms [22]–[24] or rewriting phrases [25]. They are not secure enough since the statistical distribution of generated stego texts is changed seriously. To tackle the obvious shortage, the generation-based methods are proposed to embed confidential information during the text generation process. Specifically, the candidate pool [26] at each moment is constructed by obeying a specific encoding algorithm [8], [9], and the stego word is formed using the bitstream of the confidential information present at the time. Due to the elaborate design, the generation-based methods achieve dramatic advances in terms of steganographic text quality and embedding capacity. However, these methods cannot guarantee statistical imperceptibility, a critical criterion for evaluating linguistic steganography.

To investigate the problem, Dai *et al.* [27] first used the total variation distance (TVD) to quantify the statistical imperceptibility, and then proposed a novel encoding method, patient-Huffman. Later on, Ziegler *et al.* [28] proposed a new steganographic algorithm using arithmetic coding to improve the statistical imperceptibility. Currently, Zhou *et al.* [29] generated stego texts based on the adaptive probability distribution and focused on eliminating the exposure bias produced due to the discrepancy between training and inference stages. Yang *et al.* [30] proposed an improved generative text steganography method to enhance the statistical imperceptibility of steganographic text. The candidate pool was constructed based on probability difference instead of greedy sampling, and entropy coding was applied to embed secret information. Furthermore, Yi *et al.* [31] proposed a novel linguistic steganographic method which enables the receiver to collect the tokens of the specific positions to directly constitute the secret message in a seemingly-natural steganographic text generated by the off-the-shelf BERT model equipped with Gibbs sampling. Deepthi *et al.* [32] used support vector machine (SVM), recurrent neural network (RNN), and CNN to provide secure data with confidentiality and integrity. The dramatic progress of linguistic steganographic methods makes an enormous challenge for linguistic steganalysis.

2. Linguistic steganalysis

Corresponding to the development of linguistic steganography, substantial progress has also been made in steganalysis. Generally, linguistic steganalysis can be categorized into two categories: feature-based and NNs-based approaches. The feature-based approaches extract the statistical properties of the inputs and then apply binary classification to the extracted features [10], [11], [33], [34]. Due to the fact that these methods rely on a variety of heuristic qualities that are manually constructed by domain specialists, they cannot be flexibly adaptable to other steganography algorithms and text domains.

In order to mitigate the limitations of the feature-based approaches, researchers have proposed a series of NNs-based approaches that apply an end-to-end structure to automatically learn the feature representations of the inputs. Wen *et al.* [12] first attempted to use the multisize CNN for linguistic steganalysis. Yang *et al.* [13] employed BiLSTM to capture the distortion of statistic distribution before and after embedding the confidential information. These NNs-based methods only obtain either the local or the global feature representations of the input text, limiting the detection ability of these methods. For the purpose of integrating the local and the global semantic features, Niu *et al.* [14] proposed the R-BiLSTM-C model composed of the BiLSTM and the asymmetric CNN. Yang *et al.* [16] used the convolutional sliding windows to learn the semantic features with multiple sizes. Hao *et al.* [15] designed a dense

BiLSTM with a feature pyramid to learn the comprehensive long dependencies of input texts. Some scholars made performance improvements in terms of inference time and model size [17], [18]. These methods rely on the CNN to extract high-level local features from the global semantic space generated by the BiLSTM and neglect the coarse granularity of semantic text units such as words, sentences, and so on. Therefore, it is meaningful to develop a new steganalysis model for optimizing the input text's joint feature representations.

III. The Proposed Method

In this section, the overall framework of the proposed linguistic steganalysis model BiLSTM-SDC was introduced firstly. Then, we elaborate on the word embedding module and the feature representation extracting module. Finally, the binary classifying module is presented in detail.

1. Overall architecture

As shown in Fig.1, the framework of the proposed BiLSTM-SDC is composed of the word embedding module, the features extracting module, and the binary classifying module. Firstly, the input sentence is converted into a matrix using the word embedding module which consists of two dynamic embedding layers followed by an integrating convolutional layer. Then, the local semantic features and the long dependency representations are extracted by the SDC and attentional BiLSTM, respectively. Sequentially, the two types of features are concatenated to form the ultimately joint features. Finally, the joint features are fed into a fully-connected layer with a softmax activation function to realize the linguistic steganalysis.

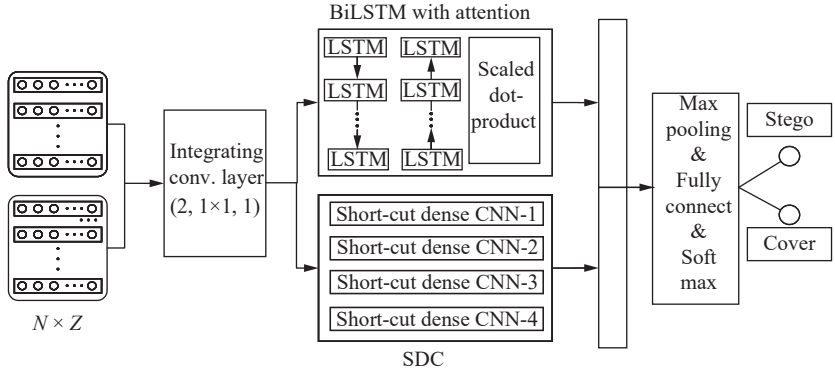

Fig. 1. The overall framework of the BiLSTM-SDC.

2. Word embedding module

Most competitive NNs-based linguistic steganalysis models [12], [14] have the dynamic word embedding layers, which are initialized by different approaches, including the random initialization with uniform distribution in the interval [−1; 1] [12] and the well-trained Google word2vec [21]. The former ensures that the data distribution remains unchanged so that the information in the network can be better transmitted, while the latter utilizes extra information to make the model converge faster. Similar to the previous models, the proposed BiLSTM-SDC choose the Google word2vec initialization to initialize the dynamic embedding layers and map an input sequence of sentence representations $\{w_1, w_2, ..., w_N\}$ into two different matrices $D_{\rm r}, D_{\rm g} \in \mathbb{R}^{N\times Z}$, where each Z-dimension vector denotes a word w_i of the input sentence. Subsequently, a convolutional layer is utilized to yield a word embedding matrix $D_{\rm m} \in \mathbb{R}^{N\times Z}$ integrating the two different word representations. The output is denoted as follows:

$$D_{\rm m} = [D_{\rm r}, D_{\rm g}] \tag{1}$$

3. Feature representation extracting module

1) Attentional BiLSTM

We use a BiLSTM model to extract features from both the forward and backward directions to obtain the context information of the word at the same time. The computation process is simplified as follows:

$$\overrightarrow{h_i}, \overrightarrow{s_i} = f_{LSTM_f}(d_i, \overrightarrow{h_{i-1}}, \overrightarrow{s_{i-1}}), \quad i \in \{1, ..., N\} \tag{2}$$

$$\overleftarrow{h_i}, \overleftarrow{s_i} = f_{LSTM_b}(d_i, \overleftarrow{h_{i-1}}, \overleftarrow{s_{i-1}}), \quad i \in \{N, ..., 1\} \tag{3}$$

where $d_i \in \mathbb{R}^{1\times Z}$ is the i-th row of the D_m representing the word embedding vector of the i-th word of input sentence. f_{LSTM_f} and f_{LSTM_b} denote the forward and backward LSTM functions respectively, and the $\overrightarrow{h_i}, \overrightarrow{s_i}$, $\overleftarrow{h_i}, \overleftarrow{s_i} \in \mathbb{R}^{1\times M}$ (M is the number of hidden units of the LSTM layer) represent corresponding output vectors. Even if the BiLSTM is specialized for sequential modelling and can learn the long dependency representations of the input texts, the semantic representations extracted by BiLSTM can not reflect the different contributions of the different words. For the steganography,

since several output words of the steganographic sentences generated by a well-trained LM might be suboptimal, the different words have more dramatic influence on the semantic representations of the steganographic sentence than that of cover sentence. As a result, assigning the different weights to the all semantic features is an effective approach to optimize the long dependency representations of the input text. Generally, attention mechanism [19], [20] can focus on the suboptimal words and reduce the impact of normal generated words on the text semantics. In the BiLSTM-SDC, in order to prevent the extremely small gradients of the softmax layer, we employ the scaled dot-product attention to learn the different weights. The attention mechanism is defined as follows:

$$\alpha_i = f_{\text{softmax}}\left(\frac{w_i h_i}{\sqrt{M}} + b_i\right), \quad i \in \{1, ..., N\} \tag{4}$$

$$Out_i = \alpha_i h_i, \quad i \in \{1, ..., N\} \tag{5}$$

where f_{softmax} denotes the softmax activation function. w_i and b_i are the parameters of the attention layer. $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \in \mathbb{R}^{2\times M}$ represents the i-th hidden state of the BiLSTM, and its assigned weight is α_i. Out_i denotes the i-th weighted output.

2) Short-cut dense CNNs

The short-cut dense CNNs is designed to directly capture extra local semantic features from the word embedding matrix. The idea behind SDC is to integrate the features from all convolutional layers to improve the representation ability. However, directly fusing the features of the traditional convolutional layers is not practical because it would dramatically increase computation and memory consumption. Hence, we use the short-cut and dense connection, which is diagrammed in Fig.2. The SDC is composed of four parallel short-cut and dense connecting CNN structures where each CNN consists of four convolutional layers. For each block, the $(C_i, k_1 \times k_2, C_o)$ denotes the number of inputs, the convolutional kernel size, and the number of outputs. In the first two layers, the complex filters $K_c \in \mathbb{R}^{K\times Z}$ are factorized into two simple filters $K_{s1} \in \mathbb{R}^{1\times Z}$ and $K_{s2} \in \mathbb{R}^{K\times 1}$. After that, two conventional convolutional layers are used to learn high-level representations.

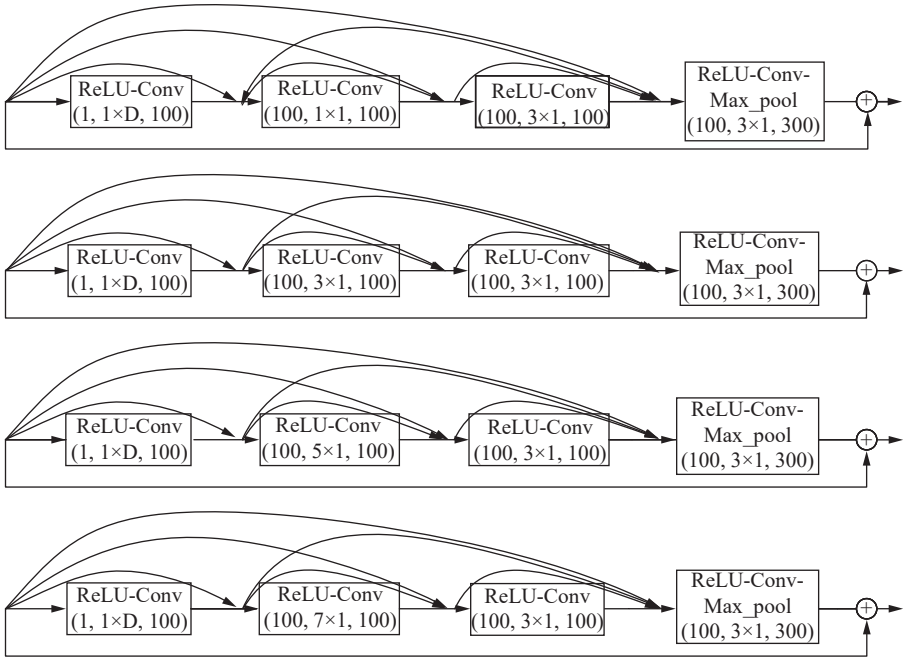

Fig. 2. The detail of short-cut dense CNNs.

We employ a dense structure to connect each convolutional layer to every other layer in a feed-forward fashion. Then, the feature maps extracted by each parallel CNN incorporate the word embedding matrix with the short-cut connection. Finally, the above feature maps are concatenated to form the ultimate local semantic features. The procedure can be formulated as:

$$R_i^j = f_{\text{relu}}(w_i^j[R_0; \cdots; R_{i-1}^j]), \quad i, j \in \{1, ..., 4\} \tag{6}$$

$$\gamma^j = f_{\text{maxpool}}(R_4^j) \oplus R_0, \quad j \in \{1, ..., 4\} \tag{7}$$

$$\Gamma = [\gamma^1; \gamma^2; \gamma^3; \gamma^4] \tag{8}$$

where f_{relu} denotes the ReLU [35] activation function, and f_{maxpool} is the max pooling operation. w_i^j is the weights of the i-th convolutional layer of the j-th parallel CNN where the bias term is omitted for simplicity. $[;]$ and $\oplus$ refer to the concatenation and addition operation, respectively. R_i^j is the i-th convolutional layer out-

put in the j-th parallel CNN, except that R_0 is equals to the $D_{\rm m}$ that denotes the output matrix from the word embedding module. γ^j represents the the j-th parallel CNN output, and the Γ denotes the final output of the SDC.

4. Binary classifying module

A fully-connected layer followed by a softmax activation function is used to map the feature representations into the probability values of two categories. Firstly, the long dependency representations and the local semantic features are concatenated to form the comprehensive joint semantic features, and then the features are fed into the fully-connected layer. Finally, the softmax activation function is exploited to produce the probabilities of two class labels. The module can be formulated as:

$$y' = f_{\rm softmax}(w \times [\Gamma; Out_N] + b) \tag{9}$$

where $f_{\rm softmax}$ denotes the softmax activation function. w and b are the learnable parameters of the fully connection layer. y' represents the probabilities of two class labels. Moreover, in order to prevent the overfitting, the commonly used *dropout* operation [36] is used behind the fully connection layer.

IV. Experiments and Analysis

1. Experimental setup

The experiments are conducted on four widely used corpora, including Mscoco [37], Twitter [38], IMDB [39], and News [40]. The detailed information of the experimental dataset is shown in Table 1. We use two generation-based linguistic steganography to verify the performance of the proposed BiLSTM-SDC, i.e., the block-based and the variable length coding (VLC)-based algorithms proposed by Fang *et al.* [8] and Yang *et al.* [9], respectively. The embedding capacity of the stego text is from 1 to 4 bits per word (bpw). Since the number of payloads is not fixed in VLC, we use the average value of the embedding capacity in the total generated stego text as the payload. We randomly select the 10,000 and 1,000 sentences from the well-processed corpora as cover text in training and testing phases, respectively. The same number of sentences are generated by the above two steganography models as stego text.

Table 1. The detailed information of experimental dataset

Dataset	Number of sentences				Average length
	Training set		Testing set		
	Cover	Stego	Cover	Stego	
Twitter	10,000	10,000	1,000	1,000	6.67
Mscoco	10,000	10,000	1,000	1,000	10.39
IMDB	10,000	10,000	1,000	1,000	14.41
News	10,000	10,000	1,000	1,000	17.61

The experiments are implemented on the PyTorch1.1.0 and NVIDIA 1080Ti graphics cards. The hyperparameters are set as follows: The word embedding dimension is 300, and the number of the BiLSTM hidden state is 200. The Kaiming initialization is used to initialize the weights of SDC, and the weights of other layers are initialized using "Xavier" initialization. And the Adam optimization algorithm [41] is utilized to update the parameters, and the learning rate is set to 0.001. The keep probability of the dropout [36] layers is 0.5, and the batch size is 64. In addition, the final experimental results are the average values of the three best validation models in the training phase. The accuracy and precision are used as the evaluation metrics which are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative, and we assume the stego text are positive samples.

2. Comparison with prior arts

In this subsection, we mainly compare the proposed BiLSM-SDC with five state-of-the-art NNs-based steganalysis methods, i.e. LS-CNN [12], TS-BiRNN [13], R-BiLSTM-C [14], BiLSTM-Dense [15], and the MS-TL [17]. The steganographic models we choose are the two most commonly used generation-based schemes: block-based and VLC-based steganography schemes.

The detection results are shown in Tables 2 and 3, respectively. As noted in Table 2 , it is obvious that these steganalysis models have better detection performance for high-capacity embedding scenarios. The straightforward reason is that the semantic distribution inconsistency between the cover and stego texts will gradually magnify with the increase of the embedding capacity. Besides, the results show that in all but a few cases, the BiLSTM-SDC outperforms the previously reported models, establishing a new state-of-the-art performance for linguistic steganalysis.

From Table 3 , the results indicate that the proposed BiLSTM-SDC is superior to the other five NNs-based methods against the VLC-based steganography, irrespective of the embedding capacity and dataset. The

reason is that the proposed BiLSTM-SDC can not only effectively capture the long dependency representations using the attentional BiLSTM, but also directly extract the different granularities of local semantic features from the word embedding matrix. In addition, it is worth noting that an increase of the embedded capacity leads to detection performance degradation rather than enhancement. The main reason for these anomalous results is that the VLC-based steganography encodes the bitstream using the Huffman tree constructed by the reverse order of conditional probability during stego-text generation, which leads to the larger discrepancy of the probability distribution between the cover and stego texts for the lower embedding capacity. This phenomenon, also called Psic Effect which is explicitly addressed and analyzed in the reported research [42].

3. Impact of the short-cut dense CNNs

To further investigate the effect of the short-cut and dense connection structure, we implement a series of experiments where the proposed BiLSTM-SDC is compared with three different variants consisting of the proposed model without the short-cut (w/o sc), without the dense connection (w/o dense), and both of them (w/o both). The experimental results summarized in Table 4 indicate that the model equipped with the short-cut and dense connection can remarkably improve the detection performance for the tested payloads and corpora. Furthermore, it is demonstrated that the effectiveness of the dense connection is better than that of the short-cut structure for linguistic steganalysis.

It is worth mentioning that in the model structure some model variants have certain similarities with the previous models [13]–[15]. But there are also subtle differences between them. For example, variants (w/o both) and TS-BiRNN [13] are all based on BiLSTM structure, but the former applies an attention mechan-

Table 2. Results of different steganalysis methods against BLOCK-based steganography

Method		LS-CNN		TS-BiRNN		R-BiLSTM-C		BiLSTM-Dense		MS-TL		BiLSTM-SDC	
Dataset	bpw	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.
Mscoco	1	0.8180	0.8195	0.8030	0.8156	0.821	0.8153	0.8200	**0.8560**	0.8281	0.8246	**0.8345**	0.8328
	2	0.8645	0.8654	0.8680	0.8629	0.8670	0.8700	0.8605	0.8767	**0.8820**	**0.8841**	0.8740	0.8659
	3	0.9100	0.9080	0.9045	0.9140	0.9110	0.9225	0.9085	0.9049	0.9160	0.9160	**0.9235**	**0.9239**
	4	0.9470	0.9573	0.9525	0.9566	0.9525	0.9476	0.947	0.9391	0.9543	0.9556	**0.9620**	**0.9611**
Twitter	1	0.8365	0.8439	0.824	0.8333	0.84	0.8427	0.8285	0.8211	**0.853**	0.858	0.852	**0.8761**
	2	0.8870	0.8901	0.8850	0.8710	0.8890	0.8950	0.8905	0.8925	**0.9012**	**0.8981**	0.8970	0.8954
	3	0.9260	0.9185	0.9230	0.9123	0.9230	**0.9360**	0.9270	0.9261	**0.9380**	0.9314	0.9335	0.9331
	4	0.9515	0.9505	0.9470	0.9627	0.9530	0.9650	0.9410	0.9672	0.9545	0.9496	**0.9665**	**0.9700**
IMDB	1	0.8395	0.8477	0.847	0.8395	0.8445	0.8290	0.8435	0.8560	0.8543	**0.8599**	**0.869**	0.8462
	2	0.8965	0.893	0.8855	0.8898	0.9060	0.9118	0.8980	0.8887	0.9077	0.9074	**0.9180**	**0.9286**
	3	0.9395	0.9296	0.9335	0.9383	0.9395	0.9313	0.9415	0.9473	0.9478	0.9439	**0.9572**	**0.9599**
	4	0.9630	0.9639	0.9660	0.9578	0.9645	0.9558	0.9675	0.9558	0.9650	0.9622	**0.9695**	**0.9681**
News	1	0.8638	0.8387	0.8710	0.8725	0.8780	0.8828	0.8653	0.8662	0.8762	0.8778	**0.8930**	**0.8962**
	2	0.9192	0.9092	0.9135	0.9131	0.9245	0.9196	0.9257	0.9172	0.9313	**0.9345**	**0.9340**	0.9255
	3	0.9415	0.9692	0.9563	0.9591	0.9523	0.9623	0.9440	0.9496	0.9617	0.9645	**0.9625**	**0.9705**
	4	0.9635	0.9674	0.9807	**0.9830**	0.9762	0.9751	0.9753	0.9750	0.9755	0.9744	**0.9808**	0.9816

Table 3. Results of different steganalysis methods against VLC-based steganography

Method		LS-CNN		TS-BiRNN		R-BiLSTM-C		BiLSTM-Dense		MS-TL		BiLSTM-SDC	
Dataset	bpw	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.	Acc.	Precis.
Mscoco	1	0.9505	0.9709	0.9515	0.9529	0.9620	0.9676	0.9515	0.9475	0.9605	0.9560	**0.9700**	**0.9825**
	2.35	0.8815	0.8700	0.8775	0.8880	0.8980	0.8941	0.8887	0.8841	0.8968	**0.9147**	**0.9065**	0.9102
	3.08	0.8425	0.8210	0.8475	0.8338	0.8620	0.8410	0.8523	0.8565	0.8693	**0.8819**	**0.8785**	0.8729
	3.51	0.8305	0.8263	0.8480	0.8799	0.8565	0.8862	0.8430	0.8430	0.8493	0.8716	**0.8630**	**0.8998**
Twitter	1	0.9305	0.9388	0.9370	0.9396	0.9290	0.9333	0.9315	0.9328	0.9329	0.9330	**0.9425**	**0.9410**
	1.81	0.9170	0.9064	0.9045	0.8923	0.9230	0.9172	0.9020	0.8963	0.9332	0.9268	**0.9350**	**0.9466**
	3.24	0.8405	0.8271	0.8435	0.8391	0.8870	0.8802	0.8660	0.8541	0.8778	0.8607	**0.9025**	**0.9133**
	4.48	0.8040	0.8280	0.8115	0.8039	0.8335	**0.8428**	0.8095	0.8037	0.8380	0.8269	**0.8550**	0.8221
IMDB	1	0.9720	0.9101	0.9595	0.9646	**0.9765**	0.9779	0.9633	0.9573	0.9722	0.9749	0.9750	**0.9808**
	1.82	0.9525	**0.9613**	0.9575	0.9620	**0.9600**	0.9556	0.9458	0.9502	0.9537	0.9521	**0.9600**	0.9582
	3.22	0.9270	0.9287	0.9100	0.9236	0.9175	0.9273	0.9233	0.9225	**0.9822**	0.9309	0.9325	**0.9519**
	4.41	0.8585	0.8574	0.8565	0.8583	0.8545	**0.8658**	0.8580	0.9371	0.8622	0.8544	**0.8685**	0.8567
News	1	0.9795	0.9771	0.9643	0.9542	**0.9798**	0.9734	0.9682	0.9591	0.9775	0.9745	0.9790	**0.9848**
	1.82	0.9725	0.9586	0.9630	0.9549	0.9663	0.9676	0.9643	0.9610	0.9672	0.9597	**0.9740**	**0.9712**
	3.22	0.9398	0.9318	0.9223	0.9192	0.9318	0.9333	0.9270	0.9206	0.9388	0.9308	**0.9415**	**0.9367**
	4.41	0.9260	0.9088	0.9088	0.8957	0.9150	0.8903	0.9030	0.9032	0.9267	0.9124	**0.9295**	**0.9257**

ism. Variants (w/o both) and R-BiLSTM-C [14] are composed of BiLSTM and CNN, but the former adopts parallel connection mode, while the latter adopts serial connection mode. Variants (w/o sc) and BiLSTM-Dense [15] all apply dense connection. However, the former uses four densely connected CNNs for local feature extraction, while the latter uses densely connected LSTMs for global feature extraction.

Although most of the existing steganalysis models are based on LSTM and CNN, the structural design and connection mode have a significant impact on the model effect. It can be seen from Table 4 that although the BiLSTM-Dense uses densely connected LSTMs, the model performance is still slightly worse than our model variant (w/o sc) with densely connected CNNs. Furthermore, the parallel connection will fuse text features at multiple granularities. However, without some necessary means, such as dense connection and short-cut, the proposed model cannot compete in detection accuracy with R-BiLSTM-C.

Table 4. The comparison of our model variants and the common models

Model	Mscoco		Twitter		IMDB		News	
	1 bpw	3 bpw	1 bpw	3 bpw	1 bpw	3 bpw	1 bpw	3 bpw
W/o both	0.814	0.9058	0.8375	0.9154	0.8395	0.9415	0.8595	0.9585
W/o dense	0.8195	0.9132	0.8365	0.9185	0.842	0.9477	0.869	0.9592
W/o sc	0.8275	0.9225	0.8445	0.925	0.854	0.953	0.8786	0.9593
TS-BiRNN	0.803	0.9045	0.824	0.923	0.847	0.9335	0.871	0.9563
R-BiLSTM-C	0.821	0.911	0.84	0.923	0.8445	0.9395	0.878	0.9523
BiLSTM-Dense	0.82	0.9085	0.8285	0.927	0.8435	0.9415	0.8653	0.944
BiLSTM-SDC	**0.8345**	**0.9235**	**0.852**	**0.9335**	**0.869**	**0.9572**	**0.893**	**0.9625**

4. Further investigating the concatenation schemes

In addition, as can be seen from Table 5, we compare two concatenation schemes for the attentional BiLSTM and SDC, i.e. serial and the parallel (BiLSTM-SDC) concatenation. Specifically, in the serial model, the SDC does not directly extract the local semantic features from the embedding matrix but captures the high-level representations from the output of the attentional BiLSTM. The experimental results demonstrate that directly extracting the different granularities of local semantic features from the word embedding matrix can significantly boost the detection performance for linguistic steganalysis.

Table 5. The effectiveness of two concatenation schemes

Model	Mscoco		Twitter		IMDB		News	
	1 bpw	3 bpw	1 bpw	3 bpw	1 bpw	3 bpw	1 bpw	3 bpw
Serial model	0.8305	0.914	0.833	0.93	0.8605	0.9385	0.8565	0.9410
BiLSTM-SDC	**0.8345**	**0.9235**	**0.852**	**0.9335**	**0.869**	**0.9572**	**0.8930**	**0.9625**

5. Effect of the attentional BiLSTM

The superiority of the attentional BiLSTM is demonstrated in Fig.3. We experiment with four corpora containing 1–4 bpw of secret information. The hyperparameters are set as follows. The number of the BiLSTM hidden state is 200, and the learning rate is set to 0.001. The Adam optimization algorithm is utilized to update the parameters, and tanh is the activation function of the attention mechanism. Furthermore, the BiLSTM with and without attention mechanism are compared to verify the feature extraction performance. The result shows that the attention mechanism can refine the long dependency representations extracted by the original BiLSTM to improve the detection performance for linguistic steganalysis.

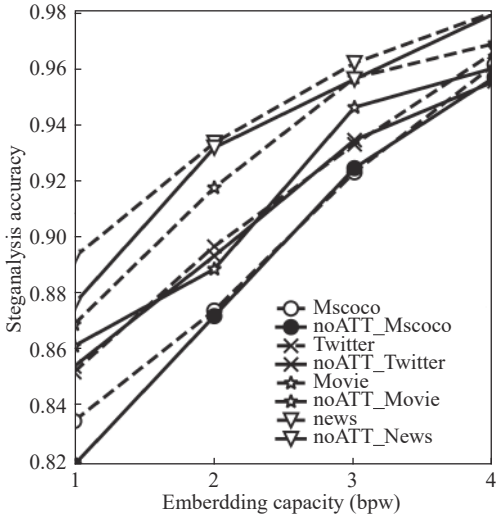

Fig. 3. The effectiveness of the attention mechanism.

V. Conclusions

Recently, with the advanced neural language models, generation-based steganography has achieved remarkable improvement, which makes a huge challenge for the corresponding steganalysis. In this paper, we propose a novel NNs-based linguistic steganalysis meth-

od. First, the BiLSTM with scaled dot-product attention mechanism is used to optimize the long dependency representations of the inputs, and then a parallel CNN structure with the short-cut and dense connection is applied in parallel to directly extract sufficient local semantic features from the word embedding matrix. Finally, the long dependency representations and the local features are concatenated to form the final joint features, which are fed into the global maxpooling and softmax layer to classify the stego and cover texts. The experimental results demonstrate that the proposed BiLSTM-SDC is superior to the previous state-of-the-art NNs-based methods against the generation-based linguistic steganography.

In further work, we will finetune a pretrained language model such as bidirectional encoder representations from transformers in a specific dataset of the linguistic steganalysis to further improve the detection performance for linguistic steganalysis, and take advantage of the transfer learning to investigate the generalization ability for the mismatch problem of the linguistic steganalysis.

References

[1] F. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding-A survey," *Proceedings of the IEEE*, vol.87, no.7, pp.1062–1078, 1999.

[2] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol.11, no.2, pp.221–234, 2016.

[3] Y. Luo, J. Qin, Y. Tan, *et al.*, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol.17, no.1, pp.125–135, 2020.

[4] Y. Tew and K. Wong, "An overview of information hiding in H. 264/AVC compressed video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.24, no.2, pp.305–319, 2014.

[5] Y. Xue, J. Zhou, H. Zeng, *et al.*, "An adaptive steganographic scheme for H. 264/AVC video with distortion optimization," *Signal Processing: Image Communication*, vol.76, no.8, pp.22–30, 2019.

[6] Y. Luo and Y. Huang, "Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, Philadelphia, PA, USA, pp.99–104, 2017.

[7] J. Wen, X. Zhou, M. Li, *et al.*, "A novel natural language steganographic framework based on image description neural network," *Journal of Visual Communication and Image Representation*, vol.61, no.5, pp.157–169, 2019.

[8] T. Fang, M. Jaggi, and K. J. Argyraki, "Generating steganographic text with lstms," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, pp.100–106, 2017.

[9] Z. Yang, X. Guo, Z. Chen, Y. Huang, *et al.*, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol.14, no.5, pp.1280–1295, 2018.

[10] L. Xiang, X. Sun, G. Luo, and B. Xia, "Linguistic steganalysis using the features derived from synonym frequency," *Multimedia Tools and Applications*, vol.71, no.3, pp.1893–1911, 2014.

[11] H. Yang and X. Cao, "Linguistic steganalysis based on meta features and immune mechanism," *Chinese Journal of Electronics*, vol.19, no.4, pp.661–666, 2010.

[12] J. Wen, X. Zhou, P. Zhong, *et al.*, "Convolutional neural network based text steganalysis," *IEEE Signal Processing Letters*, vol.26, no.3, pp.460–464, 2019.

[13] Z. Yang, K. Wang, J. Li, *et al.*, "TS-RNN: Text steganalysis based on recurrent neural networks," *IEEE Signal Processing Letters*, vol.26, no.12, pp.1743–1747, 2019.

[14] Y. Niu, J. Wen, P. Zhong, *et al.*, "A hybrid R-BILSTM-C neural network based text steganalysis," *IEEE Signal Processing Letters*, vol.26, no.12, pp.1907–1911, 2019.

[15] H. Yang, Y. Bao, Z. Yang, *et al.*, "Linguistic steganalysis via densely connected LSTM with feature pyramid," in *Proceedings of IH&MMSec'20: ACM Workshop on Information Hiding and Multimedia Security*, Denver, CO, USA, pp.5–10, 2020.

[16] Z. Yang, Y. Huang, and Y. Zhang, "TS-CSW: Text steganalysis and hidden capacity estimation based on convolutional sliding windows," *Multimedia Tools and Applications*, vol.79, no.25, pp.18293–18316, 2020.

[17] W. Peng, J. Zhang, Y. Xue, *et al.*, "Real-time text steganalysis based on multi-stage transfer learning," *IEEE Signal Processing Letters*, vol.28, no.12, pp.1510–1514, 2021.

[18] Y. Xue, L. Kong, W. Peng, *et al.*, "An effective linguistic steganalysis framework based on hierarchical mutual learning," *Information Sciences*, vol.586, no.2, pp.140–154, 2022.

[19] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Proceedings of 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, pp.5998–6008, 2017.

[20] K. He, X. Zhang, S. Ren, *et al.*, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp.770–778, 2016,

[21] G. Huang, Z. Liu, L. van der Maaten, *et al.*, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp.2261–2269, 2017.

[22] L. Xiang, X. Yang, J. Zhang, *et al.*, "A word-frequency-preserving steganographic method based on synonym substitution," *International Journal of Computational Science and Engineering*, vol.19, no.1, pp.132–139, 2019.

[23] M. Li, K. Mu, P. Zhong, *et al.*, "Generating steganographic image description by dynamic synonym substitution," *Signal Processing*, vol.164, no.9, pp.193–201, 2019.

[24] L. Xiang, J. Yu, C. Yang, *et al.*, "A word-embedding-based steganalysis method for linguistic steganography via synonym substitution," *IEEE Access*, vol.6, no.10, pp.64131–64141, 2018.

[25] H. M. Meral, B. Sankur, A. S. Özsoy, *et al.*, "Natural language watermarking via morphosyntactic alterations," *Computer Speech & Language*, vol.23, no.1, pp.107–125, 2009.

[26] L. Xiang, W. Wu, X. Li, *et al.*, "A linguistic steganography based on word indexing compression and candidate selection," *Multimedia Tools and Applications*, vol.77, no.26, pp.28969–28989, 2018.

[27] F. Z. Dai and Z. Cai, "Towards near-imperceptible steganographic text," in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, Florence, Italy, pp.4303–4308, 2019..

[28] Z. M. Ziegler, Y. Deng, and A. M. Rush, "Neural linguistic steganography," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, pp.1210–1215, 2019.

[29] X. Zhou, W. Peng, B. Yang, *et al.*, " Linguistic steganography based on adaptive probability distribution," *IEEE Transactions on Dependable and Secure Computing*, vol.19, no.5, pp.2982–2997, 2022.

[30] B. Yang, W. Peng, Y. Xue, *et al.*, " A Generation-based Text Steganography by Maintaining Consistency of Probability Distribution," *KSII Transactions on Internet and Information Systems*, vol.15, no.11, pp.4184–4202, 2021.

[31] B. Yi, H. Wu, G. Feng, *et al.*, " ALiSa: Acrostic linguistic steganography based on BERT and Gibbs sampling," *IEEE Signal Processing Letters*, vol.29, no.4, pp.687–691, 2022.

[32] G. Deepthi, N. V.SriLakshmi, P. Mounika, *et al.*, "Linguistic steganography based on automatically generated paraphrases using recurrent neural networks," *Mobile Computing and Sustainable Informatics*, vol.68, no.1, pp.723–732, 2022.

[33] C. M. Taskiran, U. Topkara, M. Topkara, *et al.*, "Attacks on lexical natural language steganography systems," in *Proceedings of SPIE 6072, Security, Steganography, and Watermarking of Multimedia Contents VIII*, SPIE, article no.607209, 2006.

[34] Z. Chen, L. Huang, H. Miao, *et al.*, " Steganalysis against substitution-based linguistic steganography based on context clusters," *Computers Electrical Engineering* , vol.37, no.6, pp.1071–1081, 2011.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, USA, pp.315–323, 2011.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol.15, no.1, pp.1929–1958, 2014.

[37] T. Lin, M. Maire, S. J. Belongie, *et al.*, "Microsoft COCO: Common objects in context," in *Proceedings of 13th European Conference on Computer Vision (ECCV 2014)*, Zurich, Switzerland, pp.740–755, 2014.

[38] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," available at: *https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf*, 2009.

[39] A. L. Maas, R. E. Daly, P. T. Pham, *et al.*, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, pp.142–150, 2011.

[40] Kaggle, "All the news," available at: *https://www.kaggle.com/snapcrack/all-the-news/data*, 2017-8-20.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, pp.1–15, 2015.

[42] Z. Yang, S. Zhang, Y. Hu, *et al.*, " VAE-Stega: Linguistic steganography based on variational auto-encoder," *IEEE Trans. Inf. Forensics Secur.*, vol.16, no.1, pp.880–895, 2021.

WEN Juan received the B.E. degree in information engineering and Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications. She is now an Associate Professor at China Agricultural University. Her research interests include artificial intelligence, information hiding, and natural language processing. (Email: wenjuan@cau.edu.cn)

DENG Yaqian is currently pursuing the M.E. degree in computer technology with the College of Information and Electrical Engineering, China Agricultural University. Her research interest includes information hiding.
(Email: dengyaqian@cau.edu.cn)

PENG Wanli received the B.E. degree from College of Physics and Electronic Engineering, Harbin Normal University, China, in 2018. He is currently pursuing the Ph.D. degree with the College of Information and Electrical Engineering, China Agricultural University, Beijing, China. His research interests include information hiding and natural language processing. (Email: hunanpwl@cau.edu.cn)

XUE Yiming (corresponding author) is currently a Professor in the College of Information and Electrical Engineering, China Agricultural University. His research interests include multimedia processing, multimedia security, and VLSI design. (Email: xueym@cau.edu.cn)