Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.037067
Article

A General Linguistic Steganalysis Framework Using Multi-Task Learning

Lingyun Xiang[1,*], Rong Wang[1], Yuhang Liu[1], Yangfan Liu[1] and Lina Tan[2,3]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[2]School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK
[3]School of Computer Science, Hunan University of Technology and Business, Changsha, 410205, China
*Corresponding Author: Lingyun Xiang. Email: xiangly210@163.com
Received: 21 October 2022; Accepted: 21 December 2022

Abstract: Prevailing linguistic steganalysis approaches focus on learning sensitive features to distinguish a particular category of steganographic texts from non-steganographic texts, by performing binary classification. While it remains an unsolved problem and poses a significant threat to the security of cyberspace when various categories of non-steganographic or steganographic texts coexist. In this paper, we propose a general linguistic steganalysis framework named LS-MTL, which introduces the idea of multi-task learning to deal with the classification of various categories of steganographic and non-steganographic texts. LS-MTL captures sensitive linguistic features from multiple related linguistic steganalysis tasks and can concurrently handle diverse tasks with a constructed model. In the proposed framework, convolutional neural networks (CNNs) are utilized as private base models to extract sensitive features for each steganalysis task. Besides, a shared CNN is built to capture potential interaction information and share linguistic features among all tasks. Finally, LS-MTL incorporates the private and shared sensitive features to identify the detected text as steganographic or non-steganographic. Experimental results demonstrate that the proposed framework LS-MTL outperforms the baseline in the multi-category linguistic steganalysis task, while average Acc, Pre, and Rec are increased by 0.5%, 1.4%, and 0.4%, respectively. More ablation experimental results show that LS-MTL with the shared module has robust generalization capability and achieves good detection performance even in the case of spare data.

Keywords: Linguistic steganalysis; multi-task learning; convolutional neural network (CNN); feature extraction; detection performance

1 Introduction

Steganography [1,2] conceals secret messages within a carrier and then transmits them through public channels, so that prospective eavesdroppers are unaware of the existence of the hidden confidential information. Since the text is the most popular and frequently used information interaction

media in people's daily lives, linguistic steganography [3–5], using text as the carrier for covert communication, has excellent research value and practical significance.

Currently, linguistic steganography has attracted wide attention from researchers and emerged as an intriguing area. It mainly includes two types of methods modification-based [6] and generation-based [7]. In modification-based methods, such as synonym substitution, the content of natural texts is selected by a human and slightly modified to achieve the purpose of hiding information. Nevertheless, it has the disadvantage of low hiding capacity due to the low redundancy of the texts. The generation-based steganography automatically generates steganographic texts with the assistance of the trained neural network-based language model, which increases the hiding capacity by generating unlimited text. Moreover, texts generated by a well-trained language model are more natural, implying that generation-based steganographic methods can generate higher-quality steganographic texts [8]. At present, generation-based linguistic steganography is the dominant and promising branch of linguistic steganography.

As the counter-technique of linguistic steganography, linguistic steganalysis [9] aims to reveal whether secret messages are hidden in detected texts (a text that carries secret information is called a steganographic text, and a text that does not is a non-steganographic text), preventing the seemingly regular communication between criminal offenders. Namely, linguistic steganalysis methods need to find out as much as possible about the difference in linguistic characteristics between steganographic and non-steganographic texts, and then classify them correctly. Over the past decades, linguistic steganalysis has proliferated. Early methods [10] mainly relied on manually designed statistical features to capture the artifacts of embedding operations, such as context fitness and word frequency distribution. However, they are developed based on the statistical changes targeted by specific steganography and required various heuristic features designed by the domain specialists, which hinders their performance and universality. In particular, the prevailing generation-based linguistic steganography can automatically generate higher-quality steganographic texts that are so statistically and linguistically close to natural texts, thus, it is much more difficult to distinguish them using hand-crafted features [11].

Due to the unprecedented success of deep learning, neural network models have been widely introduced in linguistic steganalysis [12]. The strong capability of models to automatically learn and extract discriminative features, such as the semantic and syntactic features inside detected texts, drives the detection performance of linguistic steganalysis tasks. Although these approaches achieve excellent performance and demonstrate tremendous potential for detecting generation-based linguistic steganographic methods, they perform almost a single binary steganalysis task to distinguish between steganographic and non-steganographic texts targeted by specific steganography. In more detail, different kinds of steganographic and non-steganographic texts are obtained by multiple means. For example, generated non-steganographic texts (generated without secret messages) and natural non-steganographic texts (selected from text corpora) are both referred to as non-steganographic texts; at the same time, steganographic texts also can be classified as generative steganographic texts (generated under the control of secret messages [4]) or modified steganographic texts (created by linguistic steganography through slightly modifying a natural text [13]). However, existing works in this field have yet to perform steganalysis tasks for a mixture of four categories of steganographic or non-steganographic texts. In summary, the detection performance of the prevailing binary classification methods will be limited when various types of texts coexist.

Intuitively, steganographic and non-steganographic texts from different sources carry various text properties, which may provide helpful information with varying characteristics for steganalysis

tasks. For example, modified steganographic texts are obtained by slightly modifying natural non-steganographic texts. In contrast, generative steganographic texts are automatically generated under the constraint of secret messages, meaning they contain statistical and linguistic properties specific to the texts. When the number of tasks is large, and the learned features are required for each task, multitask learning is an appropriate approach, which provides a convenient way to combine information from multiple tasks and uses the correlation features between these related tasks to help improve the classification performance of a single task. Given the above point, we opt to use multi-task learning [14–16] to mine potential correlation information among several steganalysis tasks. Experimental results demonstrate that the proposed approach enhances the feature learning and generalization ability of the prevailing single-task-based methods and improves the detection performance of each steganalysis task in a general framework. In summary, the main contributions of this paper are as follows:

- We incorporate multi-task learning into linguistic steganalysis to enhance the performance of detecting steganographic texts and propose a novel general linguistic steganalysis framework, which takes advantage of the private and shared features with CNNs from multiple categories of tasks. Furthermore, the interactive shared features between tasks can effectively alleviate the data sparsity issue.
- To the best of our knowledge, this paper is the first to perform a four-category linguistic steganalysis task in a general framework, improving the universality and generalization of linguistic steganalysis methods.
- Experimental results show that the proposed framework can simultaneously detect various texts by training a separate model and achieve excellent steganalysis detection performance.

2 Related Work

Benefiting the rapid development of deep learning, the steganographic texts generated by generation-based linguistic steganographic methods are increasingly statistically similar to non-steganographic texts. Thus, the traditional steganalysis that relies solely on the use of hand-crafted features becomes a less feasible solution for detecting steganographic texts. Considering the decisive role played by linguistic features on classification accuracy, researchers have begun to introduce deep learning into linguistic steganalysis and employ neural network models to automatically capture linguistic features for text detection. The current linguistic steganalysis methods based on the neural networks are summarized in Table 1 below, where steganographic texts are called stego texts, and non-steganographic texts are represented as non-stego texts. Subsequently, the methods mentioned in Table 1 will be elaborated.

CNN is the most representative one among the many existing neural network models. The success of CNN-based generative linguistic steganographic methods has motivated researchers to apply CNN to linguistic steganalysis. Wen et al. [12] used the word embedding layer to retrieve semantic and syntax features and employed rectangular convolution kernels of varying sizes to extract discriminative features for steganalysis tasks. To detect the generated steganographic poetry, Yang et al. [17] proposed the single-feature and multi-feature fusion TS-CNN to capture the distribution differences in the semantic space before and after the information is hidden. In addition, Yang et al. [18] used convolutional sliding windows of multiple sizes to obtain relevant features, distorted before and after embedding between the generated non-steganographic and steganographic texts. Xiang et al. [19] presented a two-stage cascaded CNN-based linguistic steganalysis to improve the system's ability to recognize steganographic texts generated via synonym substitution.

Moreover, other neural networks have also been applied to tasks of linguistic steganalysis. Yang et al. [20] introduced the idea of feature pyramids into steganalysis, developing an approach that employs densely connected Long Short-Term Memory networks (LSTMs) with feature pyramids to incorporate more low-level features and thereby achieve effective detection of steganographic texts. To capture the differences in conditional probability distributions, Yi et al. [21] proposed two pre-trained methods for linguistic steganalysis, based on recurrent neural network (RNN) or a sequence autoencoder, to improve the detection performance. To address the shortcomings of traditional networks, Li et al. [22] presented insightful explorations of using capsule network-based dynamic routing to extract and analyze semantic feature differences. To fully consider the global information of the text, Wu et al. [11] introduced a Graph Convolutional Neural Network (GCN) to collect contextual information to update the node representation and further employed a global shared matrix to obtain better text representation.

The linguistic steganalysis methods described above primarily employ a single neural network to construct a steganalysis model that can detect whether the text is steganographic or non-steganographic. To further improve the detection accuracy, some researchers opted to combine multiple neural network models to design steganalysis methods. Li et al. [23] proposed a two-stage text steganalysis method, which employs Bi-LSTM to obtain sentence vectors that preserve the strong correlations between word information, and also uses Graph Neural Network (GNN) to extract anomalous features from both intra-sentence and inter-sentence levels. To address the limitations of RNN and CNN in preserving semantic features, Niu et al. [24] proposed a hybrid linguistic steganalysis scheme by combining Bi-LSTM and CNN to capture both local features and long-term semantic information. For their part, Bao et al. [25] introduced an attention mechanism to facilitate an additional focus on suspicious information. Jiao et al. [26] took this a step further by introducing a multi-head attention mechanism, connecting word representations with a multi-headed self-attentive representation for further classification. Subsequently, Zou et al. [27] employed Bidirectional Encoder Representation from Transformers (BERT) and Global Vectors for Word Representation (Glove) to capture inter-sentence contextual association relationships, then extracted context information using Bi-LSTM and finally obtained the sensitive semantic features via the attention mechanism for steganographic text detection. Xu et al. [28] employed a pre-trained BERT language model to obtain initial contextually relevant word representation, after which the extracted features were fed into an LSTM with attention to obtain the final sentence representation used to classify the detected texts. Besides, [28] also mixes the steganographic texts generated by several steganographic methods.

In general, there are multiple kinds of steganographic and non-steganographic texts (i.e., several steganalysis tasks). However, as shown in Table 1, the above-mentioned linguistic steganalysis methods implement only a single binary steganalysis task, which strictly limits the generalizability of the classification model. Accordingly, motivated by multi-task learning, this paper proposes a linguistic steganalysis framework named LS-MTL, which extends the binary steganalysis task into a four-category task to enable the detection of multiple steganalysis tasks. The proposed framework can be used to implement steganalysis tasks by employing various neural networks. Benefiting from the tremendous development in the field of deep neural networks in the past two years, related researchers have used CNN to extract high-level semantic features and subtle distribution differences of different categories of texts [17]; CNN can capture complex dependencies and automatically learn feature representations from the texts. Thus, LS-MTL employs the CNN as the base model, enhancing the extraction and generalization ability by fusing the private and shared features of various categories of steganalysis tasks, thus reducing the impact of data sparsity on detection performance. It is

experimentally demonstrated that the proposed steganalysis method can achieve superior detection performance.

Table 1: The comparison and analysis of the existing linguistic steganalysis methods

Method	Detected texts		Characteristics
	Types of texts	Stego texts	
TS-CNN [17]	Generative stego texts and natural non-stego texts.	CT-stega [17]	• Exploring the sentence-level and whole poem-level linguistic features; collecting a large corpus of steganographic poetries; • The pooling layer of CNN only retains significant semantic features, so most of the subtle differences will be filtered in the low embedding rate; specifical for steganographic poetry, which is not universal.
TS-CSW [18]		T-stega [18]	• Designing special convolutional sliding windows with multiple sizes to obtain anomalous word association features triggered by secret information embedding; estimating the amount of secret information in steganographic texts; collecting a large corpus of steganographic poetries and texts; • Specifical for steganographic poetry, not universal.
GCN [11]		T-stega [18]	• Using GCN to extract global information first; • Building graphs for each text, which is costly.
LSTM-Pyramid [20]		T-stega [18]	• Employing LSTM with feature pyramids to incorporate more low-level features; • Neglecting the coarse granularity of semantic text units, such as sentences.
Pre-train [21]		T-stega [18], RNN-based [4]	• Introducing pre-train methods by pre-training a language model based on RNN or a sequence autoencoder; • Analyzing statistical distribution differences of texts, without considering semantic features.
Dynamic-routing [22]		RNN-based [4]	• Employing capsule network with dynamic routing to extract subtle differences of semantic distribution in the low embedding rate; • Ignoring the apparent difference in texts, such as syntactic features.
LSTM-CNN [25]		T-stega [18]	• Employing an attention mechanism to identify important cues in suspicious sentences; extracting the contextual and semantic features; • Relying on CNN to extract high-level local features from the global semantic space and neglecting the coarse granularity of semantic text units such as words, sentences, etc.
Multi-Attention [26]		T-stega [18]	• Adding the attention mechanism to extract correlation linguistic features; • Only considering semantic interactions between words.

(Continued)

Table 1: Continued

Method	Detected texts		Characteristics
	Types of texts	Stego texts	
HLS [27]		T-stega [18]	• Locating where secret information may be embedded; employing BERT or Glove as an embedding layer; • Focusing on contextualized association relationships of words and context information, but ignoring inter-sentence semantic relationships.
CNN-Syn [19]	Modified stego texts and natural non-stego texts.	T-lex	• Propounding a two-stage CNN, a sentence-level, and a text-level CNN; • Only applicable to linguistic steganography based on synonyms, not universal.
LSTM-GNN [23]		T-lex	• Proposing two phrases to extract intra-sentential and inter-sentential features; • Only extensive extracting features, without more fine-grained extraction of the text's semantic, syntactic, or statistical features.
LS-CNN [12]	Generative stego text (Markov-based) and natural stego texts; modified stego (T-lex) texts and natural stego texts.		• Utilizing CNN with a decision strategy to capture the semantic and synaptic features of long texts; • The effectively extracted features are local.
R-BILSTM-C [24]	Generative stego text (T-stega [18]) and natural stego texts; modified stego (T-lex) texts and natural stego texts.		• Capturing local features and long-term semantic text features by asymmetric convolution kernels and residual shortcuts block; • Neglect the coarse granularity of semantic text units, such as words.
BERT-LSTM [28]	Conducting a three-category task, generative (T-stega [18]), modified stego texts (T-lex), and natural non-stego texts.		• Introducing the idea of transfer learning; • Increasing data noise when multiple types of steganographic texts are mixed.

3 Proposed Method

Supposing that a complex problem needs to be solved, it can be decomposed into several simple and mutually independent subproblems; multiple results can then be integrated to obtain the results of the initial complex problem. We observed that the individual subproblems are interrelated, and the rich correlation information enriched between the problems is ignored when treating the problems as a single independent task. Existing linguistic steganalysis tasks typically detect a single category of steganographic and non-steganographic texts. In contrast, multiple categories of steganographic and non-steganographic texts coexist, and we need to use various steganalysis models to achieve multi-category text detection. Notably, it is worth noting that it ignores the interactive sensitive linguistic features, which still need to perform better detection performance.

To alleviate the problem mentioned above, this paper introduces the idea of multi-task learning to share the interaction information and sensitive linguistic features among multiple linguistic steganalysis tasks while aiding the private features of each steganalysis task. By obtaining more comprehensive and meaningful auxiliary details in this way, LS-MTL can significantly enhance the detection performance of each task in a multi-category steganalysis method. If there are M categories of

steganographic texts and N categories of non-steganographic texts, LS-MTL will build k steganalysis tasks, and $k = M * N$. In this paper, we construct four linguistic steganalysis tasks by considering the four categories of texts (generative and modified steganographic texts, generative and natural non-steganographic texts). As shown in Fig. 1, LS-MTL obtains the word embedding representation at the pre-processing step; then constructs private and shared feature spaces by CNNs for steganalysis tasks; finally, the globally comprehensive features are obtained by fusing each steganalysis task's private and shared features of each steganalysis task, which are used by the classifier to determine whether the detected text is a steganographic text (stego text) or a non-steganographic text (non-stego text). It is worth adding that the private feature space is employed to retain the linguistic features specific to each task, which are extracted by a private CNN severally; while another is utilized to capture shared inter-textual interaction features in the corpus of all steganalysis tasks, which a shared CNN constructs. The proposed general framework of LS-MTL is presented in more detail below.

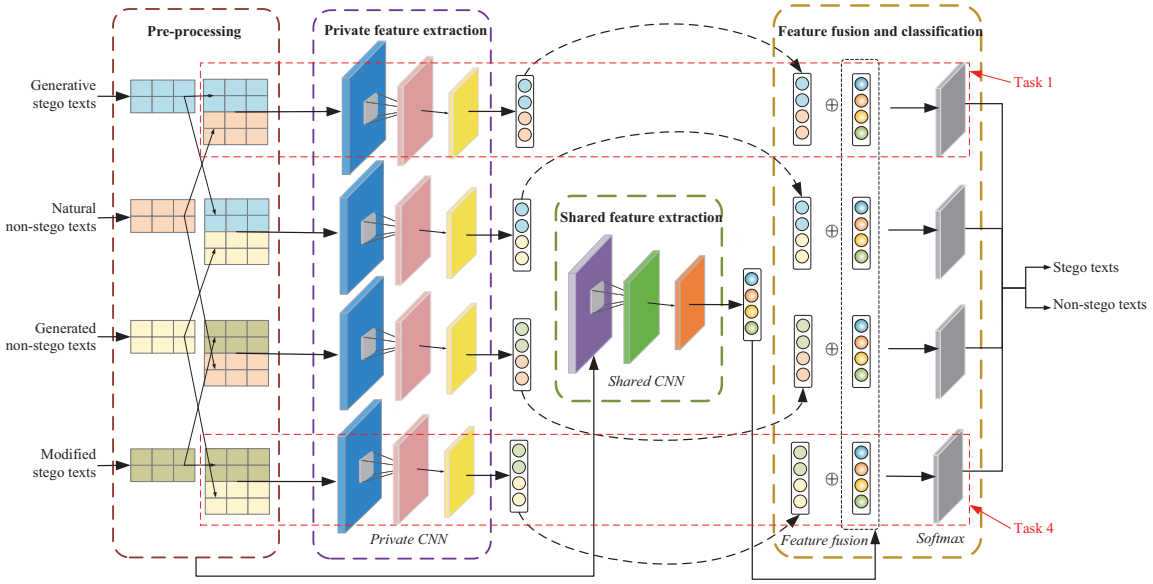

Figure 1: The overall framework of the proposed linguistic steganalysis framework, LS-MTL

3.1 Pre-Processing

LS-MTL first uses a word embedding model to preprocess the corpus and then obtains the word vector matrix x^k based on the text of the k-th steganalysis task, as shown in Eqs. (1) and (2) below:

$$D_k = \left\{\left(x_i^k, y_i^k\right)\right\}_{i=1}^{N}, \tag{1}$$

$$x_i^k = \left[V_1^k, \ldots, V_l^k\right]\left(V_j^k \in \mathbb{R}^d, 1 \le j \le l\right), \tag{2}$$

where k represents the k-th steganalysis task; N represents the number of texts; D_k denotes a corpus containing N_k texts for the k-th steganalysis task; x_i^k and y_i^k indicate the detected text and the corresponding label of the text x_i^k in the k-th steganalysis task, respectively; V_j^k is the vector representation of the j-th word in the i-th text for the k-th steganalysis task, and all the V_j^k denotes the representation of the detected text x_i^k; l is the length of the text; $\mathbb{R}^d$ is the vector space with d dimension.

3.2 Private Feature Extraction

For obtaining the specific features contained in the k-th single steganalysis task, this paper employs k CNNs to get the respective private feature representation of the texts for each steganalysis task respectively, as outlined in more detail below.

First, the word vector matrix of the detected text is input into the private CNN of the steganalysis task to which the text belongs. LS-MTL then leverages multiple convolutional kernels of various widths to extract features and generate the candidate feature representation for the k-th steganalysis task. c_i^k is denoted as in Eq. (3):

$$c_i^k = f\left(W \cdot x_{i:i+h-1}^k + b\right), \tag{3}$$

where k represents the k-th steganalysis task; c_i^k is the candidate feature representation for the k-th steganalysis task; W is a convolutional kernel weight matrix; h is the width of the convolutional kernel matrix; $x_{i:i+h-1}^k$ represents the candidate feature representation of the selected text to be detected controlled by the convolutional kernel; b is the bias term; $f(\cdot)$ is the nonlinear activation function.

For the features obtained from the different convolution kernel matrices, a maximization pooling operation $\max(\cdot)$ is performed, respectively. In short, the maximum value is taken for each candidate feature representation to obtain the maximum pooled feature $\hat{c}_i^k$ for the k-th steganalysis task in Eq. (4):

$$\hat{c}_i^k = \max\left(c_1^k, \ldots, c_{l-h+1}^k\right). \tag{4}$$

The private feature representation z^k of the detected text for the k-th steganalysis task is obtained by concatenating the maximum pooled features obtained from convolution kernel matrices together. Meanwhile, z^k is represented as following Eq. (5):

$$z^k = \left[\hat{c}_1^k, \ldots, \hat{c}_m^k\right], \tag{5}$$

where m is the number of convolutional kernel matrices of the private CNN.

3.3 Shared Feature Extraction

Multi-task learning aims to enhance detection performance by learning tasks in parallel, leveraging the correlative features between multiple-category steganalysis tasks. Thus, to simultaneously obtain the sensitive interaction features between each steganalysis task and other related tasks, this paper designs a shared CNN to extract shared features for multiple linguistic steganalysis tasks, while simultaneously stabilizing the impact of diverse data on the detection performance of steganalysis tasks. The shared features are obtained by following the steps presented below.

When the word vector matrix x^k of the detected text for a specific steganalysis task is input to the corresponding private CNN, x^k will also be input to the pre-constructed shared CNN. The shared CNN employs multiple convolutional kernel matrices of different widths to extract features and simultaneously generate the candidate feature representations. s_i is calculated by Eq. (6):

$$s_i = f\left(W \cdot x_{i:i+h-1} + b\right), \tag{6}$$

where s_i is the candidate feature representations; W is a convolutional kernel weight matrix; x represents the word vector matrix of the selected text to be detected; $x_{i:i+h-1}^k$ represents the candidate feature representation of the selected text to be detected controlled by the convolutional kernel, and h is the width of the convolutional kernel matrix; b is the bias term; $f(\cdot)$ is the nonlinear activation function.

A maximization pooling operation is performed for the features obtained from the convolution kernel matrices. Briefly, each candidate feature is represented by taking the maximum value to obtain the maximum pooled feature $\hat{s}$, which can be described as Eq. (7):

$$\hat{s} = \max(s_1, \ldots, s_{l-h+1}). \tag{7}$$

The maximum pooled features obtained from convolutional kernel matrices are concatenated together to obtain a shared feature representation o for all steganalysis tasks, which is expressed as Eq. (8):

$$o = \left[\hat{s}_1, \ldots, \hat{s}_n\right], \tag{8}$$

where n is the number of shared convolutional kernel matrices of the CNN.

3.4 Feature Fusion and Classification

After obtaining the private features z^k for the k-th steganalysis task and the shared features o, LS-MTL concatenates the two types of features to form the combined feature vector for the k-th steganalysis task in Eq. (9), that is:

$$H^k = z^k \oplus o, \tag{9}$$

where $\oplus$ is the concatenation operator; the H^k refers to the concatenation of the private features z^k for the k-th steganalysis task and the shared features o.

Next, the combined feature vector H^k is processed by a specific classifier, and the detection probability distribution can be obtained by an activation function *softmax*. The computation is performed as Eq. (10):

$$\hat{y}^k = softmax\left(W \cdot H^k + b\right), \tag{10}$$

where $\hat{y}^k$ is the probability distribution of the text to be detected for the k-th steganalysis task; H^k denotes the combined feature vector of private and shared features; W and b are the trainable model parameters. Finally, the predicted label will be obtained by comparing the probability distributions of the categories.

This paper leverages a supervised learning framework and minimizes the loss function via back-propagation iterative optimization to obtain the best model. The loss function calculates the average cross-entropy between the predicted and actual label as the prediction error, is defined as follows in Eq. (11):

$$Loss = -\sum_{k=1}^{K}\sum_{i=1}^{N_k} \beta_k y_i^k \log\left(\hat{y}_i^k\right), \tag{11}$$

where K denotes the number of steganalysis tasks; N_k represents the number of samples in the corpus of the k-th steganalysis task; β_k is the weight of the k-th steganalysis task; y_i^k is the actual label of the i-th sample in the k-th steganalysis task; and $\hat{y}_i^k$ denotes the predicted label of the i-th sample in the k-th steganalysis task. The prediction error will be progressively smaller through the model, meaning that the prediction result will be closer to the actual label.

4 Experimental Results and Analysis

4.1 Datasets

In the experiments, this paper utilized the current promising steganographic methods to produce the corresponding categories of steganographic texts, which are universal, highly concealed, and challenging to be detected by statistical analysis. The detected texts are referred into four categories: modified steganographic texts, generative steganographic texts, generated non-steganographic texts, and natural non-steganographic texts. Modified steganographic texts are produced by a steganographic tool T-Lex [19], which modifies natural texts from the Gutenberg corpus by synonym substitutions to hide secret messages. Generative steganographic texts are automatically generated by the generative linguistic steganography in [4], using the steganographic encoding methods FLC and VLC with the Movie dataset. Texts generated by the same method proposed in [4], but without embedding secret messages, are treated as generated non-steganographic texts. While natural non-steganographic texts are randomly selected from the original Movie dataset. For each steganalysis task, we choose 10,000 steganographic texts and 10,000 non-steganographic texts, for generative steganographic texts, we selected 5,000 generated by each of the two encoding methods (FLC and VLC) to form the whole generative steganographic text dataset. Notably, we choose and mix 8,000 steganographic texts and 8,000 non-steganographic texts as the training set, and other steganographic texts and non-steganographic texts are blended as well.

We constructed four-category steganalysis tasks, corresponding to four types of datasets, to detect whether a text is a steganographic text. The first type of dataset, named Dataset 1, includes generative steganographic texts and generated non-steganographic ones; Dataset 2 consists of generative steganographic texts and natural non-steganographic ones; Dataset 3 is composed of modified steganographic texts and generated non-steganographic ones; the dataset that comprises of modified steganographic texts and natural non-steganographic texts is called Dataset 4. In our experiments, we train LS-MTL with all four categories of datasets one time and input the test texts into the trained LS-MTL to distinguish whether the corresponding tested text is steganographic or non-steganographic text.

4.2 Setup and Metrics

In this paper, the structure of the CNN used in this paper is as follows: an embedding layer, we pre-trained a word embedding model, Word2vec, on the Google News corpus to produce a 300-dimensional dense vector for each word; a convolutional layer with 3, 4, 5 three different sizes of convolutional kernels, and each size has 100 convolutional kernels; a pooling layer and a fully connected layer following with a softmax classifier. Besides, the learning algorithm is the minibatch gradient descent with the Adam algorithm, and the learning rate is initialized as 0.00001 and the epoch as 100. Note that the hyper-parameters are finally obtained through training, and we adopted the model performing best during the training process to evaluate test sets.

To evaluate the performance of our method, we employ several evaluation metrics commonly used in linguistic steganalysis tasks: Accuracy (Acc), Precision (Pre), and Recall (Rec). Among them, Acc reflects the ratio of the number of correctly detected texts to the total number of samples; Pre is the ratio of the detection accuracy to actual detection, and Rec reflects the percentage of correct detection to what should have been detected. These metrics are calculated as shown in Eqs. (12)–(14) below:

$$\text{Acc} = \frac{TP + TN}{TP + FP + FN + TN}, \tag{12}$$

$$\text{Pre} = \frac{TP}{TP + FP}, \tag{13}$$

$$\text{Rec} = \frac{TP}{TP + FN}, \tag{14}$$

For each steganalysis task, TP (True Positive) is the number of steganographic texts that are correctly detected as steganographic ones; TN (True Negative) is the number of non-steganographic texts that are rightly detected as non-steganographic ones; FP (False Positive) denotes the number of non-steganographic texts that are incorrectly detected as steganographic ones; FN (False Negative) represents the number of steganographic texts that are incorrectly detected as non-steganographic ones.

4.3 Results and Analysis

4.3.1 Comparison Experiments

To verify the effectiveness of the proposed LS-MTL in this paper, we opt to use the linguistic steganalysis proposed in [12], referred to as S-CNN, as the comparison method. S-CNN employs CNN as its primary neural network to learn features for the detection of generative steganographic texts, which is dedicated to linguistic steganalysis for single-based binary classification. In this paper, we trained and tested for each steganalysis task with the corresponding dataset, while we trained LS-MTL with all the datasets and tested with the corresponding dataset for each steganalysis task. The results of the comparison experiments between LS-MTL and S-CNN on the four categories of datasets for linguistic steganalysis are shown in Table 2 and Fig. 2.

Table 2: The comparison of experimental results between LS-MTL and S-CNN on different steganalysis tasks

Datasets	Model	Acc	Pre	Rec
Datasets 1	S-CNN	0.927	**0.962**	0.890
	LS-MTL (Ours)	**0.943**	0.947	**0.938**
Datasets 2	S-CNN	0.878	0.857	**0.908**
	LS-MTL (Ours)	**0.888**	**0.927**	0.841
Datasets 3	S-CNN	0.996	**0.998**	0.994
	LS-MTL (Ours)	**0.997**	0.996	**0.998**
Datasets 4	S-CNN	**0.940**	0.945	0.935
	LS-MTL (Ours)	0.937	**0.948**	**0.967**
Average	S-CNN	0.936	0.941	0.932
	LS-MTL (Ours)	**0.941**	**0.955**	**0.936**

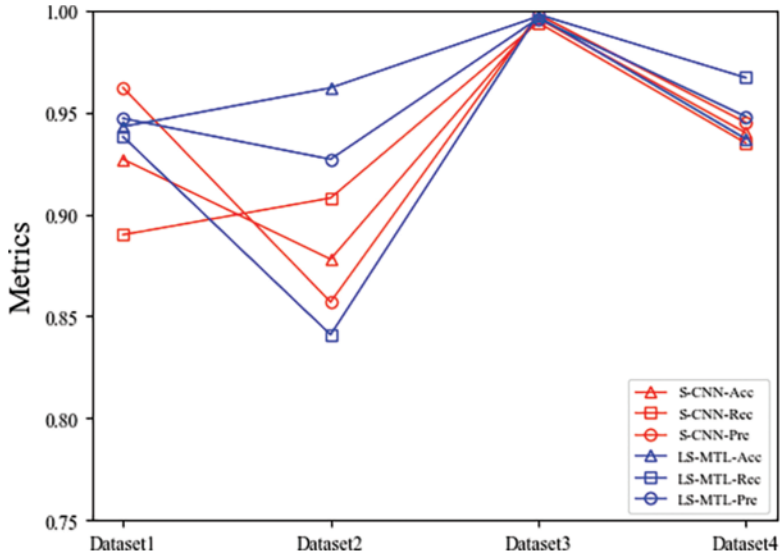

Figure 2: The comparison results of S-CNN and LS-MTL

As shown in Table 2, S-CNN is demonstrated excellent detection performance on certain metrics on datasets. We analyze that S-CNN is a single-based binary classification specifically for linguistic steganalysis, which purposefully extracts linguistic features for particular steganalysis tasks. Therefore, S-CNN also achieves excellent detection performance, the average Acc is 93.53%, the average Pre is 94.05% and the average Rec is 93.175%. Meanwhile, as illustrated in Table 2 and Fig. 2, the proposed method achieves an average Acc of 94.10%, an average Pre is 95.5% and average Rec is 93.60%, which exhibits better average detection performance for all metrics compared to the baseline, with Acc improving by 0.5%, Pre by 1.4%, and Rec by 0.4%. LS-MTL is proven to detect steganographic and non-steganographic texts more correctly by obtaining the higher Acc in Datasets 1, 2, and 3 of steganalysis tasks, and it achieves the highest Acc of 99.7%. In the steganalysis tasks on Datasets 3 and 4, LS-MTL and S-CNN achieve detection performance of Pre that is not very different from each other. However, in the steganalysis task on Dataset 2, our method is far ahead of the baseline by 7% Pre improvement. For Rec, LS-MTL exceeds the baseline by about 3% in most cases, achieving better detection performance than S-CNN. Generally, results convincingly show that LS-MTL can effectively improve detection performance by obtaining relevant interaction linguistic features from multiple steganalysis tasks. Besides, it implements multiple linguistic steganalysis tasks in the same model LS-MTL and improves detection performance, which is sufficient to prove that the proposed framework has robust generalization capability.

It is worth noting that LS-MTL and S-CNN exhibit varying detection performance on several steganalysis tasks. However, both methods achieve remarkable detection performance in the steganalysis tasks on Datasets 3. We determine that the two datasets are extremely easy to classify due to the significant differences in their text style and linguistic features. Specifically, the modified steganographic texts are derived from the Gutenberg corpus, while the generated non-steganographic texts are from the Movie dataset. Moreover, the writing styles used in these datasets are distinctive, making them very easy to detect and distinguish.

4.3.2 Ablation Experiments

To analyze the effect of different factors on the detection performance of LS-MTL, we conduct the ablation experiments mainly from two perspectives: model structure and dataset. First, to measure the practical effectiveness of the shared module in LS-MTL on the detection performance, we constructed two models based on LS-MTL: the LS-MTL prototype and the LS-MTL variant without the shared CNN module. Since the LS-MTL without the shared module does not share mutual information among multiple tasks, it implements a single binary classification task on the corresponding dataset, so we train a separate model with the corresponding dataset for the eliminated shared CNN to implement the corresponding separate binary classification task. Meanwhile, we named the LS-MTL variant as the single-task, and the prototype of LS-MTL as the multi-task. Theoretically, the shared feature capturing from multi-category steganalysis tasks can effectively mitigate the impact of data sparsity on detection performance. To argue the above issue, we simulate the existing data sparsity situation by cropping the number of steganographic and non-steganographic texts to 5,000 for each steganalysis task separately. In this paper, we train the LS-MTL model on the cropped and uncropped datasets, respectively. While the model trained on the cropped dataset is denoted as task-1, the model trained on the uncropped dataset is represented as task-2. In summary, we implemented the comparison experiments in four scenarios: single-task-1, single-task-2, multi-task-1, and multi-task-2. The experimental results are presented in Table 3 and Fig. 3.

Table 3: Comparison results of ablation experiments for LS-MTL on four categories of datasets

Datasets	Models	Acc	Pre	Rec
Datasets 1	Single-task-1	0.883	0.914	0.846
	Single-task-2	0.904	0.920	0.884
	Multi-task-1	**0.922**	**0.902**	**0.947**
	Multi-task-2	**0.943**	**0.947**	**0.938**
Datasets 2	Single-task-1	0.810	0.886	0.711
	Single-task-2	0.838	0.940	0.722
	Multi-task-1	**0.872**	**0.914**	**0.821**
	Multi-task-2	**0.888**	**0.927**	**0.841**
Datasets 3	Single-task-1	0.942	0.997	0.886
	Single-task-2	0.997	0.997	0.998
	Multi-task-1	**0.996**	**0.994**	**0.998**
	Multi-task-2	**0.997**	**0.996**	**0.998**
Datasets 4	Single-task-1	0.884	0.895	0.870
	Single-task-2	0.926	0.930	0.921
	Multi-task-1	**0.931**	**0.909**	**0.956**
	Multi-task-2	**0.937**	**0.948**	**0.967**
Average	Single-task-1	0.900	0.923	0.828
	Single-task-2	0.916	0.947	0.881
	Multi-task-1	**0.930**	**0.930**	**0.931**
	Multi-task-2	**0.941**	**0.955**	**0.936**

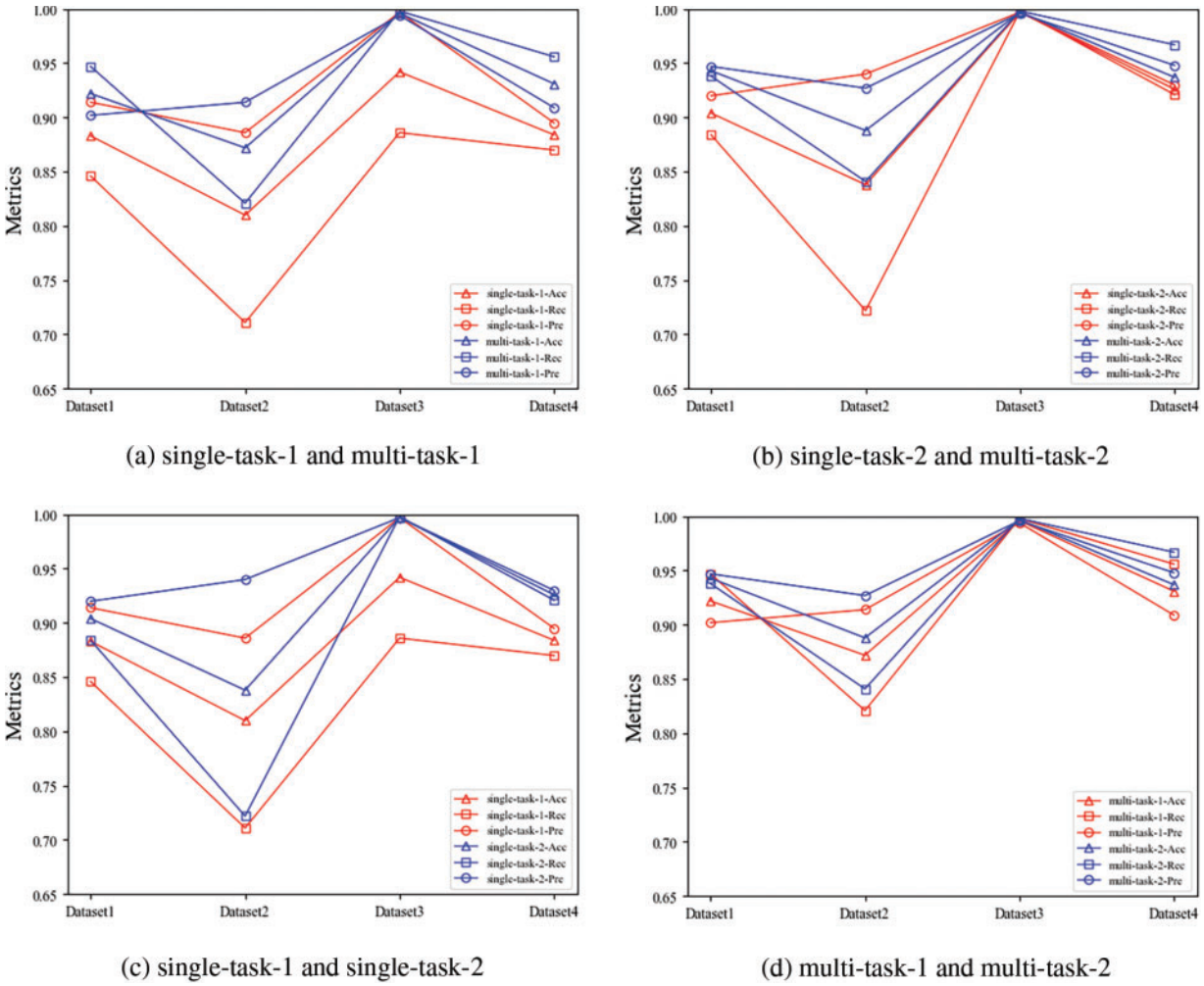

Figure 3: The comparison results of ablation experiments

As shown in Figs. 3a and 3b, all the multi-task learning methods outperform single-task classification methods in all steganalysis tasks, i.e., the multi-task-1 methods are superior to single-task-1, and the multi-task-2 are preferable to single-task-2. As can be analyzed from Table 3, the multi-task learning approaches achieved excellent detection performance compared to the single-task approaches, where multi-task-1 outperformed single-task-1 by 3% in average Acc, 0.7% in average Pre, and 10.3% in average Rec; multi-task-2 outperformed single-task-2 by 2.5% in Acc, 0.8% in Pre, and 10.3% in Rec higher. Experimental results demonstrate that shared features between multiple tasks by introducing the idea of multi-task learning can significantly and consistently enhance the performance of linguistic steganalysis methods and the generalization capability of the proposed framework.

From the experimental results in Figs. 3c and 3d, we noticed that the detection performance of single-task-1 is lower than that of single-task-2, while the performance of multi-task-1 is also lower than that of multi-task-2. Experimental results indicate that data sparsity problem affects the performance of linguistic steganalysis tasks to some extent, regardless of single-task or multi-task text classification. From Table 3, we find a big difference between the single-task-1 and single-task-2

models, with average variance values of 1.6%, 1.6%, and 5.3% for the Acc, Pre, and Rec three indicators, respectively. Besides, it even shows a difference in Rec value of up to 11.2% on Datasets 3 between single-task-1 and single-task-2.

From Table 3, the difference in detection performance between the multi-task-1 and multi-task-2 models is relatively small, with an average difference of 1%, 2%, and 0.5% for metrics, respectively. The differences in detection performance between the multi-task-1 and multi-task-2 models are far less than those between the single-task-1 and single-task-2 models under the reduced dataset. Notably, for the last two types of tasks in Table 3, multi-task-1 achieves about the same detection performance as single-task-2 with fewer data. These findings conclusively demonstrate that steganalysis based on multi-task learning is less dependent on the amount of available data than single-task learning. The multi-task learning-based approach has better generalization ability and mitigates the data sparsity problem to a remarkable extent.

5 Conclusion and Future Work

This paper presents a general multi-task learning-based framework for linguistic steganalysis, LS-MTL, which makes it possible to implement multi-category linguistic steganalysis tasks with a single model. Compared to related works, LS-MTL provides a solution for a real-world scenario where there are more types of steganographic and non-steganographic texts. LS-MTL exhibits better average detection performance for all metrics compared to the baseline, with Acc improving by 0.5%, Pre by 1.4%, and Rec by 0.4%. Meanwhile, ablation experiments are conducted in terms of model structure, and it argues that the proposed framework has a strong generalization ability. In the case of sparse data, LS-MTL also achieves stable and promising detection performance. Although the proposed framework achieves excellent text detection performance, there are still some limitations. For example, the different linguistic features from multi-tasks are distinct, so using the same neural network CNN as the base model is inappropriate. In future work, we will employ diverse and appropriate neural networks to capture more effective and sensitive linguistic features for linguistic steganalysis.

Funding Statement: This paper is partly supported by the National Natural Science Foundation of China under Grants 61972057 and 62172059, Hunan Provincial Natural Science Foundation of China under Grant 2022JJ30623 and 2019JJ50287, Scientific Research Fund of Hunan Provincial Education Department of China under Grant 21A0211 and 19A265.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. F. Yang, X. Y. Luo, J. C. Lu and F. L. Liu, "Extracting hidden messages of MLSB steganography based on optimal stego subset," *Science China Information Sciences*, vol. 61, no. 11, pp. 237–239, 2018.

[2] Y. Luo, J. Qin, X. Xiang and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2779–2791, 2021.

[3] L. Y. Xiang, R. Wang, Z. L. Yang and Y. L. Liu, "Generative linguistic steganography: A comprehensive review," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 3, pp. 986–1005, 2022.

[4] Z. L. Yang, X. Q. Guo, Z. M. Chen, Y. F. Huang and Y. J. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1280–1295, 2019.

[5] C. L. Wang, Y. L. Liu, Y. J. Tong and J. W. Wang, "GAN-GLS: Generative lyric steganography based on generative adversarial networks," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 1375–1390, 2021.

[6] C. Y. Chang and S. Clark, "Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method," *Computational Linguistics*, vol. 40, no. 2, pp. 403–448, 2014.

[7] Z. L. Yang, S. Y. Zhang, Y. T. Hu, Z. W. Hu and Y. F. Huang, "VAE-stega: Linguistic steganography based on variational auto-encoder," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 880–895, 2021.

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

[9] M. T. Ahvanooey, Q. M. Li, J. Hou, A. R. Rajput and Y. N. Chen, "Modern text hiding, text steganalysis, and applications: A comparative analysis," *Entropy*, vol. 21, no. 4, pp. 355, 2019.

[10] 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.

[11] H. Z. Wu, B. Yi, F. Ding, G. R. Feng and X. P. Zhang, "Linguistic steganalysis with graph neural networks," *IEEE Signal Processing Letters*, vol. 28, pp. 558–562, 2021.

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

[13] T. Y. Liu and W. H. Tsai, "A new steganographic method for data hiding in microsoft word documents by a change tracking technique," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 24–30, 2007.

[14] P. Liu, X. Qiu and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. of the Twenty-Fifth Int. Joint Conf. on Artificial Intelligence (IJCAI)*, New York, NY, United States, pp. 2873–2879, 2016.

[15] G. Q. Lu, J. Z. Gan, J. Yin, Z. P. Luo, B. Li *et al.,* "Multi-task learning using a hybrid representation for text classification," *Neural Computing and Applications*, vol. 32, pp. 6467–6480, 2020.

[16] J. Peng, C. Xia, Y. Xu, X. Li, X. Wu *et al.,* "A multi-task network for cardiac magnetic resonance image segmentation and classification," *Intelligent Automation & Soft Computing*, vol. 30, no. 1, pp. 259–272, 2021.

[17] Z. L. Yang, N. Wei, J. Y. Sheng, Y. F. Huang and Y. J. Zhang, "TS-CNN: Text steganalysis from semantic space based on convolutional neural network," arXiv:1810.08136v1 [cs.CR], 2018.

[18] Z. L. Yang, Y. F. Huang and Y. J. 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.

[19] L. Y. Xiang, G. Q. Guo, J. M. Yu, V. S. Sheng and P. Yang, "A convolutional neural network-based linguistic steganalysis for synonym substitution steganography," *Mathematical Biosciences and Engineering*, vol. 17, no. 2, pp. 1041–1058, 2020.

[20] H. Yang, Y. J. Bao, Z. L. Yang, S. Liu, Y. F. Huang *et al.,* "Linguistic steganalysis via densely connected LSTM with feature pyramid," in *Proc. of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, New York, NY, United States, pp. 5–10, 2020.

[21] B. Yi, H. Z. Wu, G. R. Feng and X. P. Zhang, "Exploiting language model for efficient linguistic steganalysis: An empirical study," in *2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, pp. 3074–3078, 2022.

[22] H. Li and S. Jin, "Text steganalysis based on capsule network with dynamic routing," *IETE Technical Review*, vol. 38, no. 1, pp. 72–81, 2021.

[23] E. L. Li, Z. J. Fu, S. Y. Chen and J. F. Chen, "A two-stage highly robust text steganalysis model," *Journal of Cybersecurity*, vol. 2, no. 4, pp. 183–190, 2020.

[24] Y. Niu, J. Wen, P. Zhong, and Y. M. Xue, "A hybrid R-bilstm-C neural network based text steganalysis," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1907–1911, 2019.

[25] Y. J. Bao, H. Yang, Z. L. Yang, S. Liu and Y. F. Huang, "Text steganalysis with attentional LSTM-CNN," in *2020 5th Int. Conf. on Computer and Communication Systems (ICCCS)*, Shanghai, China, pp. 138–142, 2020.

[26] S. M. Jiao, H. F. Wang, K. Zhang and Y. Q. Hu, "Neural linguistic steganalysis via multi-head self-attention," *Journal of Electrical and Computer Engineering*, vol. 2021, Article ID 6668369, 5 pages, 2021.

[27] J. Zou, Z. L. Yang, S. Y. Zhang, S. U. Rehman and Y. F. Huang, "High-performance linguistic steganalysis, capacity estimation, and steganographic positioning," in *Int. Workshop on Digital Watermarking*, Melbourne, VIC, Australia, pp. 80–93, 2020.

[28] M. Xu, L. R. Yang, T. Y. Zhao and P. Zhong, "A novel linguistic steganalysis method for hybrid steganographic texts," *Journal of Physics Conference Series*, vol. 1873, no. 1, pp. 012053, 2021.