ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240698

Domain Adaptational Steganographic Text Detection Using Few-Shot Adversary-Refinement Framework

Zhang Ziwei[a], Wen Juan[a,*], Zhou Yinghan[a], Gao Liting[a] and Xue Yiming[a]

[a]College of Information and Electrical Engineering, China Agricultural University
ORCID (Zhang Ziwei): https://orcid.org/0000-0002-6194-2419, ORCID (Wen Juan): https://orcid.org/0000-0002-4199-2988, ORCID (Zhou Yinghan): https://orcid.org/0009-0006-6366-9838, ORCID (Gao Liting): https://orcid.org/0009-0004-6659-382X, ORCID (Xue Yiming): https://orcid.org/0000-0001-6500-3868

Abstract. Text steganography involves discreetly concealing sensitive messages within natural text, while text steganalysis serves as its counterpart by aiming to detect suspicious text that may contain embedded secret information. Detecting steganographic text has become increasingly difficult because evolving steganographic algorithms produce ever-changing text distributions. Consequently, few-shot text steganalysis, which identifies steganographic text with scarce examples regardless of its distribution has become a research hotspot. The state-of-the-art few-shot text steganalysis relies on the inter-class variance between classes, i.e., they behave satisfactorily in detecting large-variance classes while being incompetent in distinguishing confusable samples from similar steganographic settings. In this paper, we propose an **A**dversary-**R**efinement Framework for **T**ext **S**teganalysis, namely ARTS, which employs a task-invariant extractor and a task-relevant projector to implement an "attract and repel" process. Specifically, in the "attract" stage, we align task-invariant features through adversarial training to shorten the intra-class distance. Afterward, the refined prototypes are projected to a new space in the "repel" stage, and then a refined penalty item is applied to enlarge the inter-class distance. Extensive experiments conducted in six datasets with different inter-class variances demonstrate the superiority of the proposed model over the SOTA models.

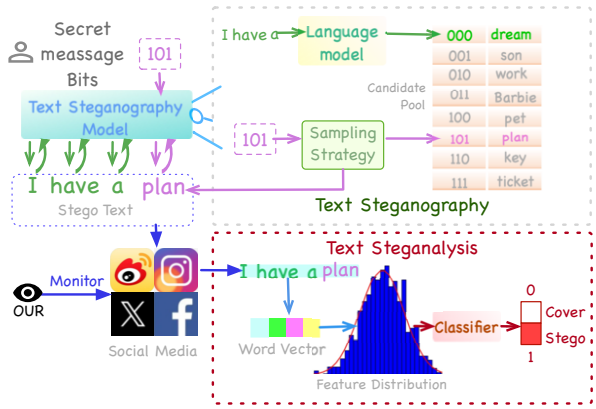


Figure 1. The framework of text steganography and text steganalysis. In the text steganography model, a language model is utilized to compute the probability distribution of the next token and generate a candidate pool based on the chosen sampling strategy. The selection of the subsequent token is then determined by the secret message bits, resulting in an innocuous stego text. Conversely, text steganalysis involves monitoring and capturing suspicious messages from social media. The captured message is then converted into a word vector and subjected to classification based on feature distributions to identify the cover text or steganographic text.

1 Introduction

The seemingly innocuous text that nevertheless hides sensitive data is known as steganographic text (or stego) [5, 26, 38]. Currently, the most hard-to-detect steganographic text is created by generative text steganography models [27, 48, 54, 2, 52], which utilize a language model to generate the probability of the next token, but instead of sampling by probability, they encode words and select the ones matching the secret bits. In Figure 1, consider a language model that generates the statement "I have a" and is about to determine the next word. The probability distribution indicates that the expected word is "dream". However, if a criminal intends to hide the secret bitstream "101" in the following words after "a", they may select the word "plan" from the candidate pool, which corresponds to the code "101". As the interception technique, text steganalysis utilizes high-precision detection of natural text (cover) or stego text. As shown in Figure 1, both "I have a dream" and "I have a plan", generated by the same language model, appear coherent and complete, but the latter contains a hidden message encoded in the word "plan". This technique poses a challenge in distinguishing between steganographic and natural text, particularly considering features such as semantics and affective bias. Consequently, text steganalysis focuses on analyzing the probability distribution of words to detect malicious entities, that conceal harmful information within seemingly natural text. In summary, text steganalysis is essential for ensuring network security.

To combat the evolving text steganography, neural-network-based text steganalysis attracts researchers' attention and plays a key role in detecting stego texts by modeling the statistical differences between the normal text and the stego text [39, 49, 31, 55, 32]. One commonality of these methods is that they train the networks by using a large amount of training data independent and identically distributed with the testing data. However, stego texts obtained from different steganographic settings, such as steganography methods, sampling strategies, or language models, may vary in distribution. Evidence shows that when one aspect of the setting changes, the detection per-

* Corresponding Author. Email: wenjuan@cau.edu.cn

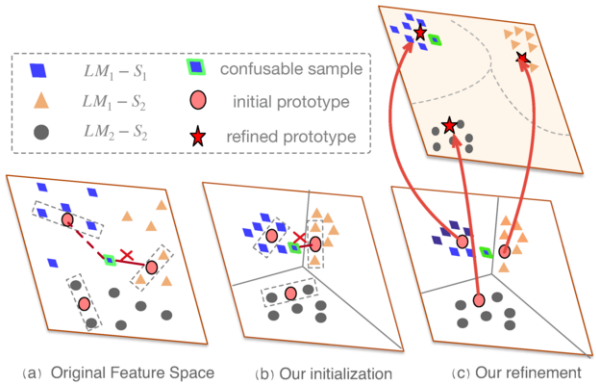


Figure 2. The illustration of the "attract and repel" process. Points of different shapes are steganographic samples generated by different language models (LM) or sampling strategies (S), represented as "$LM_i - S_j$" for simplicity. Initial prototypes are calculated from randomly selected samples framed with dotted lines.

formance drops greatly [44, 41]. This problem is also known as domain mismatch in linguistic steganalysis.

To detect multi-source steganographic texts with target distributions, Xue et al. [44] design a domain adaptation text steganalysis model to align domain-invariant features between the target and source domain. To reduce the number of required labeled samples, Wen et al. [41] propose FS-Stega based on the meta-learning framework, which enhances the model's performance to learn over diverse origins of stego texts, enabling the model to quickly adapt with a few training samples.

Although the current studies have contributed to mitigating the text steganalysis mismatch issue by extracting useful task-invariant features, we argue that excessive concern on task-invariant features will fail to detect confusable samples. In Figure 2(a), the confusable sample belonging to the parallelogram class will be misclassified to the triangle class based on the feature space of the previous work since this sample has less inter-class variance (note that these two classes are obtained by the same language model LM_1).

To tackle this, in this paper, we propose ARTS - an **A**dversary-**R**efinement Framework for **T**ext **S**teganalysis, consisting of a task-invariant extractor and a text-relevant projector to complete a "attract and repel" process for better feature representations based on meta-learning strategy in few-shot scenario. Specifically, in the "attract" stage, we use adversarial training to get task-invariant features to diminish the intra-class gap between samples and enhance model adaptability in shifts of distribution space (Figure 2(b)). Next, in the "repel" stage, we refine the prototypes of the first stage and map it into a new space via a learned matrix, while containing the intra-class distance. In addition, we propose the refined penalty item to magnify the prototype distances (Figure 2(c)). Our main contributions are summarized as follows:

- We propose ARTS for few-shot text steganalysis. Aligning reference set with the query set by supervision loss in adversarial training promotes the model adaptability for gaining better task-invariant metric space and tight intra-class distance.
- We refine prototypes by projecting them to a task-relevant metric space through a sharing matrix. We also apply refined penalty items to magnify the inter-class divergence.
- We evaluate the performance of the proposed model on six datasets with different variances. Experimental results demonstrate that ARTS achieves significant enhancement over other SOTA methods in few-shot scenarios.

2 Related Work

2.1 Text Steganalysis

Text steganalysis stabs at detecting stego text from normal text. Initial methods rely on designed handcrafted features [4, 46, 43, 29, 42]. Afterward, neural networks draw researchers' attention. LS-CNN [39] utilizes words' local correlations, while Niu et al. [31] are devoted to capturing long-term semantic features. In addition, constructing multi-scale representation in steganalysis models [32, 55, 47, 44] continues to be a hot topic.

Nevertheless, when testing samples are differently distributed from the training text, it would cause performance degradation. To enhance model adaptability Xue et al. [44] apply transductive learning to match different domain embeddings. Wen et al. [40] draw on the ideas of lifelong learning to solve multi-task text steganalysis. Besides, a few researchers contribute to fine-grand text steganalysis that distinguishes various stego texts from different origins. Yang et al. [49] utilize Recurrent Neural Networks (RNNs) to prove effectiveness in detecting stego with diverse embedding capacities. Jia et al. [18] propose HAM-Stega to detect the hierarchical information of stego text.

To avoid large target data to guarantee the fine-tuned model's performance, Wen et al. [41] propose FS-Stega, which utilizes a meta-learning framework in few-shot text steganalysis to detect stego text and cover text. Motivated by this, we propose a fine-grand few-shot text steganalysis method that detects multiple categories of steganographic text based on a meta-learning framework, solving the domain mismatch problem caused by diverse distributional stego text.

2.2 Meta-Learning

Meta-learning intends to enhance the model's learning ability to unseen tasks through a few examples. The existing meta-learning approaches consist of three main categories: (1) Gradient-based methods, which use backpropagation to learn a proper neural network initialization [10, 33, 22]; (2) Model-based methods, which are easier to optimize than gradient-based methods. MANNs [35] and SNAIL [30] are the representative methods. (3) Metric-based methods, such as PROTO [36] and Siamese Network [21] aim to optimize the transferable embedding by metric learning.

In particular, meta-learning has been applied to tackle few-shot text classification. Bao et al. [1] and Han et al. [15] design specific feature generators. Geng et al. propose an Induction Network and DMIN [12, 13] utilizing dynamic routing to adapt a support set. Another group of methods improves accuracy with the help of additional knowledge [7, 3, 17]. There are also existing methods that further revised PROTO [11, 8, 34, 16, 23]. In this study, we improve the feature representation of PROTO-based model under a small inter-class variance scenario.

3 Methodology

In this section, we first discuss the problem definition of text steganalysis. Then, the overview of ARTS is presented in section 3.2. The model details are described in the following sections.

3.1 Problem Definition

Our goal is not only to differentiate normal text and stego text but also to distinguish different stego texts originating from diverse distributions. According to the characteristics of generative text steganog-

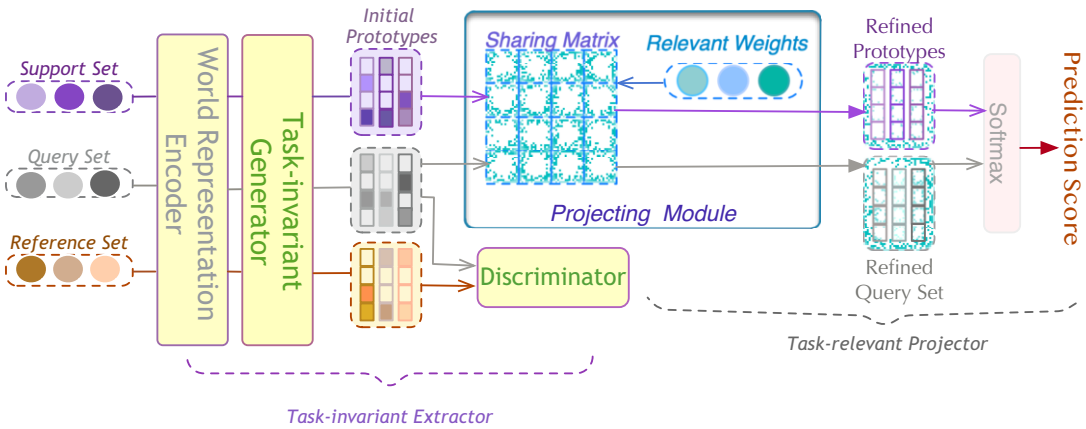


Figure 3. The overall structure of ARTS.

raphy, the probability distribution of a certain type of stego text is determined by the language model lm, the embedding capacity c, and the steganographic algorithm a involving a sampling strategy. In this way, a stego sentence $\boldsymbol{x}$ can be derived from a given generative distribution p:

$$\boldsymbol{x} \sim p\left(lm, c, a\right) \tag{1}$$

where the embedding capacity c is a parameter to control the number of bits embedded in one word.

Task set A task set is a collection of samples derived from the same distribution. e.g., a specific task set T_i can be denoted as:

$$T_i = \left\{\left(x_i^1, y_i\right), \left(x_i^2, y_i\right), \cdots \left(x_i^{n_i}, y_i\right)\right\} \tag{2}$$

where $x_i^j \sim p_i$ denotes the jth text sample in task T_i, $i \in \{1 \cdots |p|\}$, $|p|$ denotes the size of probability spaces. n_i is the number of samples in T_i. Note that all samples in T_i share the same task label y_i, which takes the value of a unique integer.

Meta-set A meta-set is a set of task sets. Let's say we have a total of $|p|$ task sets, meta-set can be denoted as $\mathcal{T} = \{T_i\}_{i=1}^{|p|}$. On account of the effectiveness of the prior works [1, 15], we fully refer to their episode training and testing strategies and split $\mathcal{T}$ into two subsets with non-overlapping label space: meta-training set $\mathcal{T}_{train}$ and meta-test set $\mathcal{T}_{test}$.

Meta-training we sample N tasks from $\mathcal{T}_{train}$ and select K and Q instances from each task to contribute to the support set T^s and the query set T^q, respectively. To leverage more statistical characteristics in the "attract" stage, we extract other $Q \times N$ samples from N different tasks to form the reference set T^r. The goal of model training is to determine the task label $\hat{y}$ for a given sample x.

Meta-testing During meta-testing, We sample N new tasks from $\mathcal{T}_{test}$ and select K and Q instances from each task to contribute to the support set and the query set, respectively. Finally, we calculate the average classification performance across all meta-testing episodes.

3.2 Overview

Our model consists of two main components: a task-invariant extractor and a task-relevant projector (Shown in Figure 3). The former follows an adversarial procedure to encourage the query set to be aligned to the reference set, promoting the model to learn better task-invariant features. The latter projects the original prototype to a new metric space to magnify the relative distances of sampled classes based on a sharing matrix, alleviating the negative impact of the smaller inter-class variance of support sets. Finally, the prediction is performed by assigning each text a class label of the closest prototype in the new metric space.

3.3 Word Representation Encoder

We obtain the initial word embeddings by applying a pre-trained BERT with B transformer layers [6]. Given the sentences $t^s \in T^s$, $t^q \in T^q$, and $t^r \in T^r$, we map them to three matrices $W_s \in R^{b_1 \times 768}$, $W_Q \in R^{b_2 \times 768}$, and $W_r \in R^{b_3 \times 768}$ as the common features, where b_1, b_2 and b_3 are the number of words in t^s, t^q, and t^r, respectively.

3.4 Task-invariant Extractor

To obtain the aligned and imperative common features shared among different distributions, we design a task-invariant feature extractor through an adversarial learning process with a generator and a discriminator. The generator confuses the discriminator in distinguishing whether the sample is from the query or reference set. In this way, they would curtail the gap among the same labeled samples and gain better contextual embeddings to boost the prediction accuracy.

Generator $G_\alpha(\cdot)$ The generator is a CNN-based structure [39] consisting of convolution kernels with different sizes h, empirically valued 3, 4, and 5. Based on the generator, the task-invariant features of support set, query set, and reference set can be expressed as $G_\alpha(W_s)$, $G_\alpha(W_q)$, and $G_\alpha(W_r)$, respectively, where α is the generator parameters. For convenience, we denote:

$$X_s = G_\alpha\left(W_s\right) \tag{3}$$

$$X_q = G_\alpha\left(W_q\right) \tag{4}$$

Discriminator $D_\beta(\cdot)$ The discriminator is to guide the generator to get better task-invariant features. Treating W_s and W_q as the target domain and W_r as the source domain, the discriminator determines whether the sample is from the source or the target domain. We employ a three-layer feed-forward neural network to construct the discriminator. The output probability of the ith sample $d^i = D_\beta\left(G_\alpha\left(W_i\right)\right)$ is calculated by $softmax$ function. We utilize

the cross-entropy loss to execute gradient updates, shown as follows:

$$\mathcal{L}_D(\beta) = -\frac{1}{2len}\sum_{i=1}^{2len}[dl^i \log d^i + \left(1 - dl^i\right)\log(1 - d^i)] \tag{5}$$

where β donates the parameters of the discriminator, dl^i represents the ground-truth label of the ith sample, and len is the number of samples in the query or the reference set.

3.5 Task-relevant Projector

The adversarial training can shorten the intra-class distance; however, it will also compress the inter-class distance. Consequently, the query instance may be misclassified to the nearest estimated prototype class. To enlarge the distance between prototypes, we introduce the task-relevant projector to map prototypes to a refined space, where the inter-class distance between different prototypes is magnified apart, while samples and the corresponding prototypes are still tight.

Concretely, in N-way K-shot text steganalysis scenario, assume the task-invariant features from support set W_s with label l construct W_s^l, and we can obtain the initial prototype s_l for each label l:

$$s_l = \frac{1}{|W_s^l|}\sum_{w_i \in W_s^l} G_\alpha(w_i) \tag{6}$$

Overall, There are three steps involved in converting initial prototypes into the refined space. Firstly, We utilize the linear layer L_θ to learn the auxiliary points $\{a_1, \ldots, a_N\}$, which contributes to the formation of the weight matrix $A = \left[\frac{a_1}{\|a_1\|}, \ldots, \frac{a_N}{\|a_N\|}\right]$. It is crucial to note that the auxiliary points are vector representations that play a vital role in the calculation of A, which assigns weights to each extracted class. In the second step, the sharing matrix M_θ is computed using the equation $X_s M_\theta = A$, where $M_\theta \in R^{768\times768}$ and $A \in R^{N\times768}$. Here, X_s^+ represents the generalized inverse of the non-square domain-invariant features matrix of the support set X_s, and it is calculated as follows:

$$X_s^+ = \left\{X_s^T X_s\right\}^{-1} X_s^T \tag{7}$$

Moreover, M_θ is trained to enhance the discriminability among different classes, facilitating accurate identification of confusable samples (as shown in Figure 2), while also preserving a compact intra-class distance. Essentially, M_θ can learn task-specific features while retaining domain-invariant characteristics. Finally, based on M_θ, we transform the query set vector X_q to the refined query vector by calculating $X_q M_\theta$. It is worth mentioning that the sharing matrix and relevant weights together constitute the **Projecting Module (PM)**.

The $softmax$ function is used to estimate the probability that each query vector $x_q \in X_q$ belongs to a certain refined prototype in the new mapping space. Concretely, given the initial prototypes $S = \{s_l\}_{l=1}^N$, the posterior probabilities is computed as:

$$p(y = l \,|x_q) = \frac{exp\left(-dis\left(x_q M_\theta, s_l M_\theta\right)\right)}{\sum_{s_l \in S} exp\left(-dis\left(x_q M_\theta, s_l M_\theta\right)\right)} \tag{8}$$

The distance function dis can be adopted either the squared Euclidean distance or the cosine distance. Moreover, $s_l \in S$ signifies that the variable s_l iterates over all prototypes in $\mathcal{S}$ to optimize the distance between x_q and all other prototypes. The classification loss is formatted as:

$$\mathcal{L}_c = \frac{1}{|X_q|}\sum_{x_q \in X_q}[dis\left(x_q M_\theta, s_l M_\theta\right) + \log\sum_{s_l \in S} exp\left(-dis\left(x_q M_\theta, s_l M_\theta\right)\right)] \tag{9}$$

Algorithm 1 Training Procedure for ARTS

Require:

Meta-training set $\mathcal{T}_{train}$ and the corresponding label Y_{train}; R episodes and ep epochs
Randomly initialize the parameters of the generator α, discriminator β, and the projector θ.
for each $i \in [1, ep]$ **do**
  Randomly sample N task sets from $\mathcal{T}_{train}$ and the corresponding label from Y_{train}.
  for each $j \in [1, R]$ **do**
    $X_s, X_q, X_r \leftarrow \emptyset, \emptyset, \emptyset$
    Sample K and Q disjoint data to form the support set T^s and the query set T^q, respectively. The remaining data in $\mathcal{T}_{train}$ form the reference set T^r
    Input T^s to the model
    Compute the initial prototype s_l by eq. 6
    Update weight $A \leftarrow \left[\frac{a_1}{\|a_1\|}, \ldots, \frac{a_N}{\|a_N\|}\right]$
    Compute X_s^+ by the eq. 7
    Compute the sharing matrix $M = X_s^+ A$
    Input T^q to the model
    Compute $\mathcal{L}_c$ and $\mathcal{L}_{RPI}$ by eq.9 and eq.10
    Input T^r to the model
    Fix α, θ, update β by minimizing eq.5
    Fix α, β, update θ by minimizing eq. 11
    Fix θ, β, update α by minimizing eq.11

3.6 Training by refined penalty item

Note that the task-relevant projector focuses on pushing samples away from prototypes of different classes. When the distance between different prototypes in the mapping space is relatively close, the performance may be greatly compromised. Thus, we propose the Refined Penalty Item (RPI) to drive refined prototypes further apart from each other, shown as follows:

$$\mathcal{L}_{RPI} = \sum_{i\neq j, s\in S} -dis\left(s_i M_\theta, s_j M_\theta\right) \tag{10}$$

Accordingly, the total loss for ARTS is:

$$\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_{RPI} - \sigma\mathcal{L}_D. \tag{11}$$

Empirically, we set $\lambda = 0.5$ and $\sigma = 1$ in the subsequent experiments for satisfied performance. The training procedure is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Basis

To evaluate the proposed model, we choose seven mainstream text steganography algorithms [9, 48, 51, 53, 50, 52, 45] to generate

Methods		M-Mix		N-Mix		T-Mix		AVG			M-Mix		N-Mix		T-Mix		AVG	
		acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$		acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$
PROTO	1 -shot	50.7	48.7	56.3	55.7	50.5	52.7	52.5	52.4	5 -shot	63.6	60.9	66.9	69.2	57.2	60.1	62.6	63.4
MAML		48.0	46.3	50.6	51.3	49.7	52.7	49.1	50.1		58.5	61.5	59.8	60.0	58.3	57.8	58.9	59.8
Induct		61.2	62.1	58.6	57.4	68.3	62.4	62.7	60.6		66.7	65.1	61.2	71.4	71.1	69.9	66.3	68.8
DS-FSL		41.8	45.6	40.9	45.2	45.8	43.0	42.8	44.6		51.0	56.2	51.9	52.0	59.9	62.5	54.3	56.9
MLADA		63.0	59.8	55.1	51.3	66.7	65.3	61.6	58.8		70.0	72.2	70.7	68.1	**78.4**	71.1	73.0	70.5
FS-Stega		36.8	42.2	50.8	52.2	54.2	52.1	47.2	48.8		62.0	65.0	64.8	66.7	59.3	61.2	62.0	64.3
meta-sn		57.2	58.0	51.5	56.9	57.8	56.5	55.5	57.1		66.0	64.6	67.4	68.5	62.8	66.0	65.4	65.8
TART		61.9	60.4	64.3	56.0	60.5	62.9	62.2	59.7		72.7	65.3	72.2	72.2	74.6	68.3	73.2	68.6
OUR		**66.9**	**67.7**	**73.4**	**74.2**	**72.0**	**72.9**	**70.8**	**71.6**		**74.4**	**75.6**	**80.5**	**80.6**	77.9	**78.3**	**77.6**	**78.2**

Table I. The comparison of 3-class detection performance (%) on larger-variance datasets under 1-shot and 5-shot scenarios. The "AVG" column is the average performance of each model across three datasets.

the cover text (embedding capacity c = 0) and stego text ($c = 1, 2, \cdots, 5$). We use four widely-used English corpora to train these steganography models: Twitter [14], COCO [24], Movie Review [28], and News [19]. Consequently, We construct six cross-capacity and cross-algorithm datasets, each of which has a total of 1120 samples and is divided into 11, 5, and 7 non-overlapping classes for training, validation, and testing. Based on their cross-corpora property, we named the six datasets M-Mix, N-Mix, T-Mix, M-M, M-N, and M-T. As we analyzed in section 1, less inter-class variance in the testing samples would reduce the model's performance when the target domain mismatches the source domain. Specifically, M-Mix, N-Mix, and T-Mix, which comprise single-corpus texts in training and mixed corpora in testing, can be considered **larger-variance datasets**. While M-M, M-T, and M-N, which take Movie Review as the training corpora and single corpus in testing, are regarded as **smaller-variance datasets**. To evaluate the performance of ARTS, we calculate acc and $f1$ [44].

Dataset	Training corpora	Testing corpora	AVG
M-Mix	Movie Review	ALL	13.22
N-Mix	News	ALL	14.94
T-Mix	Twitter	ALL	9.42
M-M	Movie Review	Movie Review	14.06
M-T	Movie Review	Twitter	10.69
M-N	Movie Review	News	15.52

Table II. The detail of the datasets. The "AVG" column reveals the average length of the sentences. "ALL" means containing all the four chosen corpora.

4.2 Baselines

We compare our ARTS with eight SOTA methods grouped into three categories: (1) metric-based methods: **Meta-SN** [16] utilizes siamese network and **TART** [23] introduces transfer module to improve **PROTO** [36]. (2) Gradient-based methods: **MAML** [10] uses prior over-model parameters to update through a few gradient steps. (3) Model-based methods: **Induction Networks** [12] proposes a dynamic routing algorithm to simulate human-like induction. **DS-FSL** [1] integrates the distribution features into the attention mechanism. **MLADA** [15] utilizes domain adversarial networks to promote the domain adaptation ability. **FS-Stega** [41], which is the representative method for meta-learning text steganalysis, takes advantage of attentional meta-learner to detect stego samples in few-shot scenarios.

4.3 Implementation Details

Our experiments are implemented by Pytorch with a Python interface. To accelerate the training process, we use the GeForce GTX 2080Ti GPU and CUDA 10.0. We use bert-based-uncased [6] as the word representation encoder and maintain the parameters of the BERT layer during training. Besides, the dropout of the convolutional layer is 0.5, the initial learning rate is 5e-5, and the small batch gradient descent of the Adam [20] is applied for optimization. The detection threshold is set to 0.5, the number of kernels with different sizes is 100, and the dimension of the word vector is 768. In total, the number of parameters is 219.79M. Our code and datasets [1] are available.

4.4 Comparisons

We first use the larger-variance datasets (M-Mix, N-Mix, and T-Mix) to evaluate the performance of ARTS in a 3-way few-shot text steganalysis task. The detection results are reported in Table I. Intuitively, our model surpasses SOTA methods across all datasets, except for accuracy in T-Mix with 1-shot setting. In particular, the average accuracy shows that ARTS outperforms TART by 8.6% and 4.4% in 1-shot and 5-shot experiments, respectively. Compared with FS-Stega, our method delivers a substantial improvement of about 23.6% and 15.6% in 1-shot and 5-shot scenarios, respectively. These results demonstrate the superiority of ARTS over the mainstream text steganalysis models on larger-variance dataset scenarios.

Moreover, we verify the proposed model on hard-to-detect scenarios involving more confusable samples. Specifically, we fix the training corpora as Movie Review (M). The test corpora are: 1) the same as training (M-M), 2) different from training but from a single distribution (M-T, M-N). We compare the results with M-Mix since they all share the same training corpora. We choose four best-performing baseline models in Table I for comparison, and the results are given in Table III. As expected, compared to the performance on M-Mix, most models experience a performance decrease on these small-variance datasets. We compute the average performance degradation AVGΔ for each model and find that ARTS has the minimum decline among all the models. One interesting finding is that the performance of ARTS on the M-M dataset is better than that on M-Mix, which is contrary to the other models. It's interpretable because ARTS has superior feature alignment ability thanks to the discriminator-guided task-invariant extractor. When the training and test sets are consistent, the alignment features learned from the training procedure can be transferred to the testing stage.

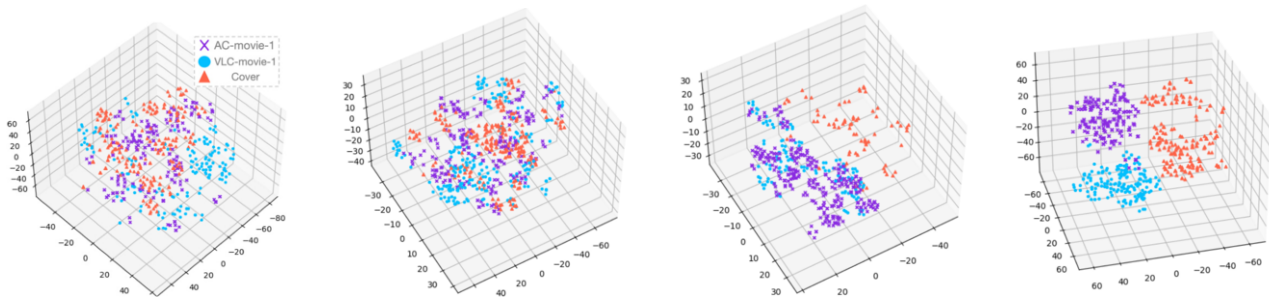
4.5 Ablation Study

We apply two additional structures to subtitle our CNN-based task-invariant generator, including a Bi-direction recurrent neural network

[1] https://github.com/zz9wa/A-R-T-S.

Methods		M-M		M-T		M-N		M-Mix		AVGΔ			M-M		M-T		M-N		M-Mix		AVGΔ	
		acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$		acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$	acc	$f1$
MLADA	1 -shot	55.7	58.3	59.8	53.5	41.4	39.9	63.0	59.8	-10.7	-9.3	5 -shot	65.4	52.4	59.7	54.0	59.7	57.3	70.0	72.2	-10.2	-16.0
FS-Stega		33.1	30.7	30.7	35.7	29.2	31.9	36.8	42.2	-5.8	-9.4		51.8	49.3	51.4	50.2	53.3	49.8	62.0	65.0	-11.4	-15.1
meta-sn		54.2	51.2	47.9	50.2	53.0	49.4	57.2	58.0	-5.5	-7.8		59.8	55.3	58.9	56.0	60.8	59.2	66.0	64.6	-7.7	-7.0
TART		55.1	50.7	55.7	58.7	59.4	60.2	61.9	60.4	-5.2	-3.8		63.6	60.2	60.0	59.5	67.7	64.3	72.7	65.3	-10.2	-3.0
OUR		**70.0**	**70.4**	**58.8**	**62.4**	**62.3**	**69.7**	**66.9**	**67.7**	**-3.2**	**-3.3**		**77.8**	**76.0**	**62.0**	**63.6**	**69.2**	**72.4**	**74.4**	**75.6**	**-3.2**	**-2.4**

Table III. 3-way few-shot detection results in small-variance datasets. AVGΔ denotes the mean value of the model performance on M-M, M-T, and M-N subtracts the performance score on M-Mix. It examines the performance degradation on datasets with smaller variances compared to those with larger variances.



(a) AVG (b) MLADA (c) TART (d) ARTS

Figure 4. The t-SNE visualization of sentence embeddings in the testing episode (3-way 5-shot 25-query). All three classes are unseen in the training procedure. Note that (a) shows the average of the word embeddings obtained by BERT without assigning weights.

with self-attention (LSTM-att) [25] and a Bi-direction recurrent neural network (bi-LSTM) [1]. We additionally evaluate the model performance by deactivating a specific module and recording the corresponding average time consumption in seconds (Time). Experiments are conducted in a 3-way 5-shot scenario on three datasets and the $f1$-scores are shown in Table IV. Plainly, the solution of taking CNN as the task-invariant generator has the optimal performance compared to the models equipped with bi-LSTM and LSTM-att, with an improvement of about 5.4% and 5.7% in terms of $f1$, respectively.

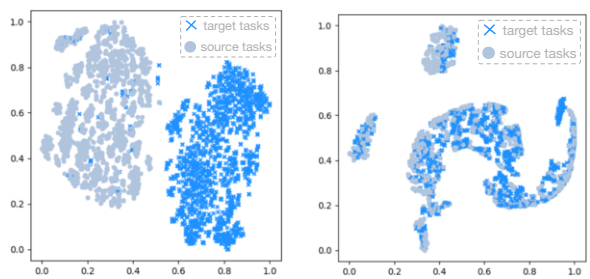
	M-M	M-T	M-N	Avg	Time
bi-LSTM	69.8	55.7	68.3	64.6	5423
LSTM-att	72.7	55.1	65.2	64.3	5903
$-\mathcal{L}_D$	73.5	59.4	64.6	65.8	**4986**
$-\mathcal{L}_{RPI}$	71.5	56.7	69.6	65.9	5313
$-PM$	64.8	60.5	62.8	64.0	5128
ARTS	**76.0**	**63.6**	**72.4**	**70.6**	5577

Table IV. Comparison of different model structures. $-\mathcal{L}_D$, $-\mathcal{L}_{RPI}$, and $-PM$ represents deactivating discriminator, task-invariant training, and the Projecting Module.

From Table IV, we can see that by removing the Discriminator $\mathcal{L}_D$ and the refined penalty item $\mathcal{L}_{RPI}$, the average $f1$-score on the three datasets declines by 4.2% and 4.1%, respectively. Moreover, By substituting the Projecting Module with PROTO, ARTS has decreased by 5.4%. In terms of efficiency, ARTS achieves a proficient balance between model performance and runtime.

4.6 *Visualization*

To intuitively inspect the ARTS's ability to distinguish samples from different classes based on the combination of the task-invariant extractor and task-relevant projector, we utilize t-SNE [37] to visualize refined sentence embeddings of the query set which input to the classifier. The detail is shown in Figure 4. Compared with the initial feature space acquired by BERT shown in Figure 4(a), MLADA does not significantly make improvements to feature spaces, only bringing a few cover samples closer (Figure 4(b)). In contrast, from Figure 4(c) we can see that TART has done a great job pushing the cover class apart from the stego classes. However, TART does not distinguish between the two types of stego samples (samples denoted by dots and crosses) because their differences were relatively small. In comparison, ARTS can construct a separate feature space for each class (Figure 4(d)). It not only perfectly pushes away three classes but also pulls samples of the same class more tightly. This verifies the effectiveness of the proposed "attract and repel" strategy, which enables ARTS to perform better in fine-grained multi-task steganalysis and classify stego samples from different distributions.



(a) $-\mathcal{L}_D$ (b) ARTS

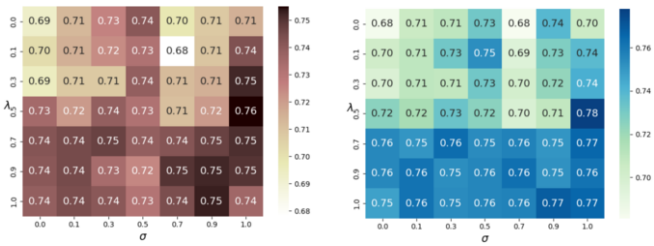
Figure 5. Visualization of feature alignment from the reference samples to the target samples.

In addition, we visually test the task-invariant extractor in aligning the features of the target sets to the features learned from the reference set. From Figure 5(a), we can see that without the supervision of

the discriminator, the features of different domains are hardly overlapped. In contrast, the feature representations of the two domains obtained by ARTS are well aligned (Figure 5(b)), which confirms that extracting task-invariant features by adversarial training can effectively improve the model's feature transfer ability and strengthen its adaptability in detecting novel tasks.

4.7 Hyper-parameter Sensitivity Analysis

We end the experiments section with a sensitivity analysis of the hyper-parameters. Specifically, we conduct on two main hyper-parameters: the weight σ and λ controlling $\mathcal{L}_D$ and $\mathcal{L}_{RPI}$, respectively. We depict hotmaps of accuracy and $f1$ score in Figure 6.



(a) acc (b) $f1$

Figure 6. Hotmaps of acc and $f1$ when changing σ and λ in 3-way 5-shot 25-query scenario.

We discover that $\sigma = 1$ and $\lambda = 0.5$ yield the optimum performance, and both two evaluation metrics have roughly the same performance trends. Besides, the color is more saturated when $\sigma > 0.5$ and $\lambda > 0.3$, signifying improved performance. It is presumably because $\mathcal{L}_D$ tights the divergence of the intra-class while $\mathcal{L}_{RPI}$ pushes different prototypes away. However, more concerns on $\mathcal{L}_{RPI}$ means paying more attention to task-relevant features that are not transferable. By the way, changes in hyperparameters within the selected range have little impact on performance, indicating ARTS is not very sensitive to hyperparameter changes.

5 Conclusion

In this paper, we propose a few-shot text steganalysis model consisting of two components: a task-invariant extractor and a task-relevant projector, to achieve fine-grained stego text detection. First, we utilize adversarial training to obtain task-agnostic features. The intra-class gap is diminished, dubbed the "attract" stage. Next, we map initialized prototypes to a new space and introduce a refined penalty item to urge the prototypes further away. This is the "repel" stage. The proposed model is evaluated on both larger-variance and smaller-variance datasets, and our ARTS outperforms previous work by a large margin in few-shot scenarios. Future work includes applying ARTS to other fields, e.g., cross-lingual scenarios, and exploring other effective meta-learning methods in text steganalysis.

6 Ethics statement

Our research is motivated by the need to prevent the misuse of steganography for public safety. We acknowledge the potential ethical implications, particularly concerning the impact on individuals' freedom and privacy. We pledge to prioritize the protection of personal privacy and the respect for civil liberties in the digital realm as we ethically advance text steganalysis technology.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62272463 and No. 61802410)

References

[1] Y. Bao, M. Wu, S. Chang, and R. Barzilay. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*, 2020.

[2] Y. Bao, H. Yang, Z. Yang, S. Liu, and Y. Huang. Text steganalysis with attentional lstm-cnn. In *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, pages 138–142. IEEE, 2020.

[3] J. Chen, R. Zhang, Y. Mao, and J. Xu. Contrastnet: A contrastive learning framework for few-shot text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10492–10500, 2022.

[4] Z. Chen, L. Huang, H. Miao, W. Yang, and P. Meng. Steganalysis against substitution-based linguistic steganography based on context clusters. *Computers & Electrical Engineering*, 37(6):1071–1081, 2011.

[5] A. Desoky. Comprehensive linguistic steganography survey. *International Journal of Information and Computer Security*, 4(2):164–197, 2010.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[7] T. Dopierre, C. Gravier, and W. Logerais. Protaugment: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. *CoRR*, abs/2105.12995, 2021. URL https://arxiv.org/abs/2105.12995.

[8] M. Fan, Y. Bai, M. Sun, and P. Li. Large margin prototypical network for few-shot relation classification with fine-grained features. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2353–2356, 2019.

[9] T. Fang, M. Jaggi, and K. Argyraki. Generating steganographic text with LSTMs. In A. Ettinger, S. Gella, M. Labeau, C. O. Alm, M. Carpuat, and M. Dredze, editors, *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL https://aclanthology.org/P17-3017.

[10] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[11] T. Gao, X. Han, Z. Liu, and M. Sun. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6407–6414, 2019.

[12] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun. Induction networks for few-shot text classification. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1403. URL https://aclanthology.org/D19-1403.

[13] R. Geng, B. Li, Y. Li, J. Sun, and X. Zhu. Dynamic memory induction networks for few-shot text classification. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1087–1094, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.102. URL https://aclanthology.org/2020.acl-main.102.

[14] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[15] C. Han, Z. Fan, D. Zhang, M. Qiu, M. Gao, and A. Zhou. Meta-learning adversarial domain adaptation network for few-shot text classification. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1664–1673, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.145. URL https://aclanthology.org/2021.findings-acl.145.

[16] C. Han, Y. Wang, Y. Fu, X. Li, M. Qiu, M. Gao, and A. Zhou. Meta-learning siamese network for few-shot text classification. In *International Conference on Database Systems for Advanced Applications*, pages 737–752. Springer, 2023.

[17] S. Hong and T. Y. Jang. Lea: meta knowledge-driven self-attentive document embedding for few-shot text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–106, 2022.

[18] j. Jia, Z. Zhang, L. Gao, J. Wen, and Y. Xue. Hierarchy-aware matching network for text steganalysis (in chinese). 2023.

[19] kaggle. Bbc news dataset. 2018. URL https://www.kaggle.com/c/learn-ai-bbc.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[21] G. Koch, R. Zemel, R. Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[22] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10657–10665. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01091. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Lee_Meta-Learning_With_Differentiable_Convex_Optimization_CVPR_2019_paper.html.

[23] S. Lei, X. Zhang, J. He, F. Chen, and C.-T. Lu. TART: Improved few-shot text classification using task-adaptive reference transformation. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11014–11026, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.617. URL https://aclanthology.org/2023.acl-long.617.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJC_jUqxe.

[26] 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*, pages 99–104, 2017.

[27] Y. Luo, Y. Huang, F. Li, and C. Chang. Text steganography based on ci-poetry generation using markov chain model. *KSII Transactions on Internet and Information Systems (TIIS)*, 10(9):4568–4584, 2016.

[28] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[29] P. Meng, L. Hang, Z. Chen, Y. Hu, and W. Yang. Stbs: A statistical algorithm for steganalysis of translation-based steganography. In *Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12*, pages 208–220. Springer, 2010.

[30] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B1DmUzWAW.

[31] Y. Niu, J. Wen, P. Zhong, and Y. Xue. A hybrid r-bilstm-c neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(12):1907–1911, 2019.

[32] W. Peng, J. Zhang, Y. Xue, and Z. Yang. Real-time text steganalysis based on multi-stage transfer learning. *IEEE Signal Processing Letters*, 28:1510–1514, 2021.

[33] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[34] H. Ren, Y. Cai, X. Chen, G. Wang, and Q. Li. A two-phase prototypical network model for incremental few-shot relation classification. In *Proceedings of the 28th international conference on computational linguistics*, pages 1618–1629, 2020.

[35] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[36] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[37] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[38] J. Wen, X. Zhou, M. Li, P. Zhong, and Y. Xue. A novel natural language steganographic framework based on image description neural network. *Journal of Visual Communication and Image Representation*, 61:157–169, 2019.

[39] J. Wen, X. Zhou, P. Zhong, and Y. Xue. Convolutional neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(3):460–464, 2019.

[40] J. Wen, Y. Deng, J. Wu, X. Liu, and Y. Xue. Lifelong learning for text steganalysis based on chronological task sequence. *IEEE Signal Processing Letters*, 29:2412–2416, 2022.

[41] J. Wen, Z. Zhang, Y. Yang, and Y. Xue. Few-shot text steganalysis based on attentional meta-learner. In B. S. Manjunath, J. Butora, B. Tondi, and C. Vielhauer, editors, *IH&MMSec '22: ACM Workshop on Information Hiding and Multimedia Security, Santa Barbara, CA, USA, June 27 - 28, 2022*, pages 97–106. ACM, 2022. doi: 10.1145/3531536.3532949. URL https://doi.org/10.1145/3531536.3532949.

[42] M. Wu and S. Jin. Text steganalysis method- breaking steganographic utility of stego. *Jisuanji Gongcheng/ Computer Engineering*, 32(24):10–12, 2006.

[43] L. Xiang, X. Sun, G. Luo, and B. Xia. Linguistic steganalysis using the features derived from synonym frequency. *Multimedia tools and applications*, 71:1893–1911, 2014.

[44] Y. Xue, B. Yang, Y. Deng, W. Peng, and J. Wen. Domain adaptational text steganalysis based on transductive learning. In *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*, pages 91–96, 2022.

[45] B. Yang, W. Peng, Y. Xue, and P. Zhong. A generation-based text steganography by maintaining consistency of probability distribution. *KSII Transactions on Internet & Information Systems*, 15(11), 2021.

[46] H. Yang and X. Cao. Linguistic steganalysis based on meta features and immune mechanism. *Chinese Journal of Electronics*, 19(4):661–666, 2010.

[47] H. Yang, Y. Bao, Z. Yang, S. Liu, Y. Huang, and S. Jiao. Linguistic steganalysis via densely connected lstm with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2020.

[48] Z. Yang, S. Jin, Y. Huang, Y. Zhang, and H. Li. Automatically generate steganographic text based on markov model and huffman coding. *CoRR*, abs/1811.04720, 2018. URL http://arxiv.org/abs/1811.04720.

[49] Z. Yang, K. Wang, J. Li, Y. Huang, and Y. Zhang. Ts-rnn: Text steganalysis based on recurrent neural networks. *IEEE Signal Processing Letters*, PP(99):1–1, 2019.

[50] 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*, 14(5), 2019. doi: 10.1109/TIFS.2018.2871746.

[51] 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*, 16:880–895, 2020.

[52] S. Zhang, Z. Yang, J. Yang, and Y. Huang. Provably secure generative linguistic steganography. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3046–3055, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.268. URL https://aclanthology.org/2021.findings-acl.268.

[53] X. Zhou, W. Peng, B. Yang, J. Wen, Y. Xue, and P. Zhong. Linguistic steganography based on adaptive probability distribution. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[54] Z. Ziegler, Y. Deng, and A. Rush. Neural linguistic steganography. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1210–1215, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1115. URL https://aclanthology.org/D19-1115.

[55] J. Zou, Z. Yang, S. Zhang, Y. Huang, et al. High-performance linguistic steganalysis, capacity estimation and steganographic positioning. In *International Workshop on Digital Watermarking*, pages 80–93. Springer, 2021.