RLS-DTS: Reinforcement-Learning Linguistic Steganalysis in Distribution-Transformed Scenario

Yihao Wang, Ru Zhang, and Jianyi Liu

Abstract—**When the data undergo a distribution change, existing linguistic steganalysis often struggles to effectively capture the statistical characteristics of the transformed cover or stego, resulting in a drop in performances. To address this issue and fully use the information from the original data before the change, this letter proposes a reinforcement learning-based method for linguistic steganalysis. This method employs an agent (steganalyzer) to interact within an observation space, enabling adaptation to the characteristics of the transformed data and capturing steganalysis features. Specifically, we map the texts to the GloVe observation space and construct an agent comprising Actor module and Critic module to provide action, state, and other information. In the pre-training phase, agent trains and reinforces the Actor and Critic modules using the original data before the change. In the fine-tuning phase, agent optimizes these two modules to extract steganalysis feature through reinforcement training with instant reward in the transformed data. Experiments show that the proposed method exhibits better performances than the baseline in the transformed scenarios. Furthermore, this method offers a more autonomous training solution for linguistic steganalysis.**

Index Terms—**Linguistic steganalysis, distribution transformation, reinforcement learning, steganalyzer, observation space.**

I. Motivation

With the widespread adoption of social platforms, various forms of cover texts have emerged, represented by Twitter and news. The ubiquity of text as a common medium of communication has fueled the rapid advancement of linguistic steganography [1], [2], [3], [4]. To effectively detect stego texts, steganalysis requires the accurate capture of statistical distribution differences between cover texts and stego texts [5]. The statistical distribution of stego texts is influenced by the forms of cover texts and the employed steganography schemes [2], [3]. So as the forms of cover texts or steganography schemes undergo changes, the distributional differences between cover texts and stego texts will be changed. The distribution transformed scenario refers to a situation where the distribution differences undergo changes, resulting in obstacles to effectively capturing steganalysis features of the transformed data.

In the distribution-transformed scenario, existing traditional steganalysis features [5], [6], [7] exhibit limited richness and lack robust quantification of steganographic disturbances, resulting in poor adaptability and an inability to detect different steganography schemes [8]. Although existing deep-learning steganalysis [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] can detect various steganography schemes, the features learned from the original data are not readily applicable to transformed data, leading to a performance decline. Specifically, existing deep-learning steganalysis faces two main challenges. Firstly, these methods lack the ability to continuously learn new text characteristics through interaction with the observation space, making it difficult to adapt to statistical changes. Secondly, these methods are built on the assumption that text statistical properties are fixed, when the distribution is transformed due to changes in the forms of cover texts or steganography schemes, these methods struggle to capture the features of the transformed cover or stego texts. This leads to a decrease in the precision or recall of detection, ultimately resulting in a lower F1 score for detection. Thus, the transformed scenario poses significant challenges to steganalysis. Furthermore, with the growing demand for steganalysis in the distribution-transformed scenario, this issue becomes increasingly prominent.

To overcome these challenges, we propose a method called RLS-DTS. RLS-DTS incorporates the ideas of reinforcement learning, enabling the agent to adjust its strategy through interaction with the observation space to adapt to the new distribution. This method captures the distributions of the transformed cover and stego texts, resulting in improved performance in the distribution-transformed scenarios.

The main work of this letter can be summarized as follows.

1) A reinforcement learning-based model for linguistic steganalysis is constructed. RLS-DTS builds a GloVe observation space that provides state information and loss component to the agent, enabling the effective extraction of steganalysis features from distribution-transformed data through interaction with the observation space.
2) The Actor and Critic modules for linguistic steganalysis are proposed. RLS-DTS introduces a concise Actor module for providing action-related information and the instant reward to optimize the features extracted by Critic and improve the detection performance. Additionally, the deep-learning Critic is introduced to provide the probability estimation of stego texts, capture and optimize the features of the transformed texts effectively.
3) The distribution-transformed scenarios are designed. We design three datasets to evaluate the performances.

Manuscript received 27 June 2023; revised 22 August 2023; accepted 28 August 2023. Date of publication 30 August 2023; date of current version 13 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grants U21B2020 and U1936216 and in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2023120. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Minoru Kuribayashi. *(Corresponding author: Ru Zhang.)*

The authors are with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yh-wang@bupt.edu.cn; zhangru@bupt.edu.cn; liujianyi1987@163.com).

Digital Object Identifier 10.1109/LSP.2023.3310380

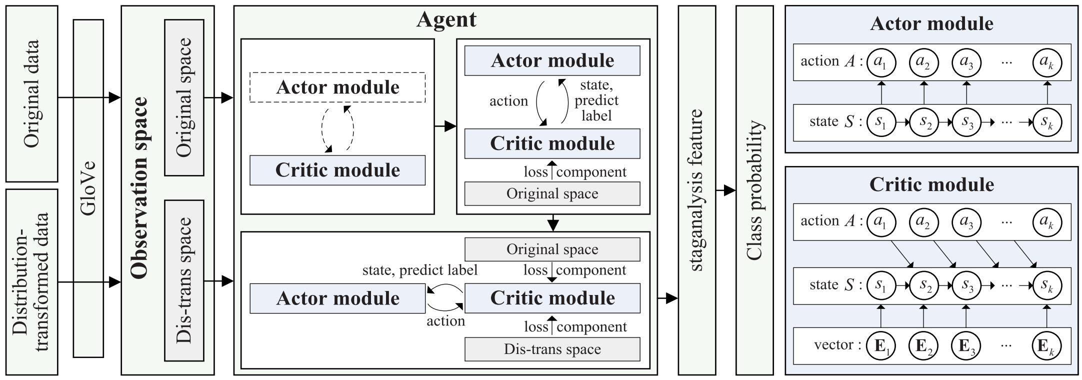

Fig. 1. Overall process of the RLS-DTS method. (The model consists of an observation space and an agent (steganalyzer). The Actor module in the agent provides actions based on the current state. The aim of this module is selected actions that can optimize the overall performance of the model. The Critic module in the agent predicts the label, state and other information based on vector inputs and optimizes the prediction accuracy. This module provides feedback signals and guidance to the Actor module. Moreover, the Critic module optimizes the extraction of steganalysis features by receiving loss component from the observation space. "Original space" and "Dis-trans space" represent the original observation space and distribution-transformed observation space. The dashed boxes and dashed lines in the figure indicate the modules and interactions not used in the current part.).

The rest of this letter is as follows: Section II provides the description of the overview and the details. Experiments are presented in Section III, comparing the methods and analyzing the results. Finally, Section IV concludes the letter.

II. RLS-DTS Proposed

A. Overview

The overall process of the method is illustrated in Fig. 1.

B. Model Details

During the optimization and feature extraction stage, it is necessary to construct an observation space that can interact with the agent. The input of this observation space is a word sequence $T = \{t_1, t_2, \ldots, t_i, \ldots, t_n\}$ of length n, where, $t_i, i = \{1, 2, \ldots, n\}$ represents the ith word in the sequence. The output is the encoding $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_i, \ldots, \mathbf{e}_n]$ of each sentence, where, $\mathbf{e}_i, i = \{1, 2, \ldots, n\}$ represents the encoding of the ith word. The formula is as follows.

$$\mathbf{e}_i = \mathbf{E}(t_i) = M[i], \tag{1}$$

where, M represents the word vector representation matrix obtained by the optimized function. $M[i]$ represents the ith row of M, which corresponds to the vector representation of the word t_i. The constructed GloVe [21] observation space is obtained using Formula 1. The reason for not constructing observation spaces based on large-scale models [22], [23] like BERT [22] is that BERT is difficult to fully use dynamically generated feedback in reinforcement learning. In contrast, GloVe can capture statistical characteristics by leveraging the global word co-occurrence matrix. The GloVe observation space enhances the robustness of handling dynamic text data.

Once the observation space is constructed, the agent (steganalyzer) interacts with the observation space [24] to optimize the extraction of steganalysis features. We designed an agent that consists of a concise Actor module and a deep-learning Critic module, providing information such as action probability distribution, state value, gradients, and other information. Different from the role of action in decision control and text classification, the action A of the RLS-TLS method does not need to provide a clear action, but provides information for calculating the gradient in the Critic module, and better guides the optimization of the Critic module. This enables the agent to optimize its actions and decision-making strategies at specific moments and states. The formulas of the Actor module and the Critic module are as follows.

$$y_{\text{Actor}} = \sigma[(h_t - 1) \times \mathbf{W}_1 + \mathbf{E} \times \mathbf{W}_2 + \mathbf{b}], \tag{2}$$

$$y_{\text{Critic}} = \text{Linear}(\text{LSTM}(\mathbf{E}, h_t - 1, c_t - 1)), \tag{3}$$

where, h_t and c_t represent the hidden unit and memory unit of the tth time step. $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{b}$ represent the weight matrix and bias vector. $\sigma(\cdot)$, $\text{LSTM}(\cdot)$, and $\text{Linear}(\cdot)$ represent the Sigmoid, LSTM, and linear operation. As can be seen from Formula 3, Critic module is composed of LSTM architecture, which extracts steganalysis features according to the supervised way used in linguistic steganalysis [8], [9], [10], [11], [12], [13], [14], [15], [16].

To evaluate the effectiveness of action policies and guide the adjustment of the agent's strategy, [25], [26] provide reward feedback signals for image steganography. In the RLS-DTS method, an instant reward R is introduced to calculate the gradient for the entire model [27]. The formula is as follows.

$$R = \alpha \times (\mathcal{L}^l - \mathcal{L}), \tag{4}$$

where, α is a hyperparameter. $\mathcal{L}^l$ is the lth $\mathcal{L}$, and the calculation of $\mathcal{L}$ is detailed in the description in II-C.

Upon receiving the policy evaluation, the Actor module uses reinforcement policy to provide specific guidance for action selection. To effectively detect stego texts under distribution transformation, the designed model employs policy gradient methods [28] to iteratively optimize the policy. The gradient formula of the policy function is as follows.

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{t=0}^{N} \nabla_\theta \log \pi_\theta(a_t | s_t)(r_t - b), \tag{5}$$

where, θ and b are the parameters of the policy function, s_t, a_t, and r_t represent the current state, action and reward $r_k \in R, k = \{1, 2, 3, \cdots\}$, and $\pi_\theta(a|s)$ represents the probability of $s_t \in S$ mapping to the selected action $a_t \in A$.

Algorithm 1: The Specific Training Process of RLS-DTS.

Input: The sentences of the original and transformed data.
Output: The probability of detecting stego texts.
1: Encode the sentences to generate the observation space.
2: Initialize the parameters of the Actor and Critic modules.
3: Evaluate and optimize the performance of the Critic module on the original space by Formula 3, 5, and 6, obtaining preliminary optimization parameters for the Actor module.
4: With the optimized Critic module obtained in Step 3, optimize the Actor module on the original space.
5: Fine-tune the performance of the Critic and Actor modules on the distribution-transformed observation space by reinforcement learning and update the parameters of modules.
6: **repeat**
7:   Update the parameters of the Critic module based on the difference between the currently predicted labels and the true labels.
8:   Update the parameters of the Actor module using the policy gradient methods.
9:   Evaluate the performance of the updated Actor module on the validation set in the distribution-transformed observation space.
10:   Save the parameters of the Actor and Critic modules based on the best accuracy achieved on the validation set in the distribution-transformed observation space.
11: **until** Convergence **or** Max number of iterations.
12: Extract steganalysis features by capturing the statistical characteristics of the distribution-transformed observation space using the parameters of the fine-tuned Critic module and Actor module.
13: Obtain the probability of detecting stego texts by the Softmax classifier.

C. Training

The training process of RLS-DTS is shown in Algorithm 1.

The whole training strategy of RLS-DTS differs from traditional supervised linguistic steganalysis [8], [9], [10], [11], [12], [13], [14], [15], [16] as it incorporates a reinforcement-learning framework. Benefiting from prior research on reinforcement learning loss function designed [27], our loss function $\mathcal{L}$ consists of two parts: the cross-entropy loss $\mathcal{L}_C$ and the regularization loss $\mathcal{L}_R$. $\mathcal{L}_C$ measures the difference between the predictions and the true labels, while $\mathcal{L}_R$ encourages the model to better capture the features of stego texts. These operations are performed S times, and their values are summed and averaged to obtain the final loss value $\mathcal{L}$. The formulas are shown as follows.

$$\mathcal{L}_C = -\frac{1}{m}\sum_{n=1}^{m}[y_i log\widehat{y}_i + (1-y_i)log(1-\widehat{y}_i)],$$

$$\mathcal{L}_R = (Rlength/length)^2 \times 0.15,$$

$$\mathcal{L} = \frac{1}{S}\sum_{p=0}^{S}\mathcal{L}^l = \frac{1}{S}\sum_{p=0}^{S}(\mathcal{L}_C^l + \mathcal{L}_R^l), \tag{6}$$

TABLE I
THE SPECIFIC INFORMATION OF THREE DATASETS IN THE CONSTRUCTED TRANSFORMED SCENARIO

Datasets	Steganography	Form of cover texts
SSDC	F (R, V) → F (R, V)	M → T, N; T → M, N; N → M, T
DSSC	F → R, V	M (T, N) → M (T, N)
	R → F, V	
	V → F, R	
Datasets	Steganography & Form of cover texts	
DSDC	$F_M \rightarrow R_{T,N}$; $F_T \rightarrow R_{M,N}$; $F_N \rightarrow R_{M,T}$	
	$R_M \rightarrow V_{T,N}$; $R_T \rightarrow V_{M,N}$; $R_N \rightarrow V_{M,T}$	
	$V_M \rightarrow F_{T,N}$; $V_T \rightarrow F_{M,N}$; $V_N \rightarrow F_{M,T}$	

Note: F, R, and V represent the Fang [1], RNN [2], and VAE [3] steganography schemes; M, T, and N represent Movie, Tweets, and News forms of the cover texts. In ”✓✓ → ✓”, ✓✓ represents the steganography scheme or cover texts of the original data; ✓ represents the steganography scheme or cover texts of the distribution-transformed data. In ”$**_{\times\times}\rightarrow *_{\times}$”, ** and ×× represent the steganography scheme and form of cover texts in the original data, while * and × represent the steganography scheme and the form of cover texts in the transformed data.

where, m represents the number of texts in a batch, y_i and $\widehat{y}_i$ represent the true label and the probability of belonging to stego text for the ith sample. $length$ and $Rlength$ represent the maximum text length in a batch and the length of the current stego text, respectively.

III. EXPERIMENTS

A. Settings

We constructed three datasets of distribution-transformed scenario for evaluation: SSDC, DSSC, and DSDC datasets, where, “·S·C” represents the **S**ame/**D**ifferent **S**teganography schemes in the **S**ame/**D**ifferent forms of the **C**over texts. The used steganography schemes are Fang [1], RNN [2], and VAE [3]. The forms of the cover texts are Movie, Tweets, and News. The size of the original data is 1,600 samples (800 covers and 800 stegos), and the size of the transformed data is 400 samples (200 covers and 200 stegos). Table I provides detailed information about these datasets.

To train and test the performance of the methods, the cover and stego texts are divided into training, validation, and testing sets in a ratio of 6:2:2. In the proposed RLS-DTS method, the hidden layer size is set to 300 dimensions. We use the Adam algorithm [29] as the gradient descent optimizer with an initial learning rate of 0.001. α is set as 0.1. The batch size is configured as 32. For the pre-training of the Critic modules, the training of the Actor modules, and the fine-tuning of these modules, the epochs are set to 30, 5, and 20, respectively.

We compared the RLS-DTS method with seven high-performance linguistic steganalysis methods selected as baseline discriminators. These methods include non-BERT-based methods: FCN [8], TS_BiRNN [9], TS_CSW [10], BiLSTM_Dense [11], and BERT-based methods: KD&FI [12], BERT_LSTM [13], SSLS [14]. The evaluation of these methods is based on $F1$ score, the formula is shown as follows.

$$F1 = 2 \times (P \times R)/(P + R), \tag{7}$$

where, P and R represent the Precision and Recall of detection. Since the Acc (Accuracy) will be too large with a certain value of P or R being too large, the $F1$ score is employed as the evaluation metric to determine if the methods successfully capture the characteristics of cover texts and stego texts.

TABLE II
$F1$ COMPARISON OF DETECTION PERFORMANCE BETWEEN RLS-DTS METHOD AND BASELINE METHODS

SSDC	Fang [1]						RNN [2]						VAE [3]					
	M→N	M→T	N→M	N→T	T→M	T→N	M→N	M→T	N→M	N→T	T→M	T→N	M→N	M→T	N→M	N→T	T→M	T→N
[8]	0.6154	0.5937	0.5429	0.4912	0.5373	0.6667	0.6000	0.5455	0.5397	0.5965	0.6111	0.6769	0.4815	0.4412	0.5075	0.4706	0.5205	0.5172
[9]	0.5667	0.6349	0.6000	0.6579	0.6465	0.6341	0.6535	0.7907	0.7129	0.7826	0.6667	0.7407	0.6750	0.4615	0.5946	0.5538	0.5672	0.5455
[10]	0.6923	0.6286	0.6667	0.7042	0.6170	0.7342	0.7222	0.7059	0.7273	0.8193	0.7292	0.8140	0.5333	0.6216	0.5946	0.5714	0.6269	0.5397
[11]	0.6000	0.6364	0.6279	0.6957	0.6053	0.6102	0.6279	0.7692	0.7391	0.7179	0.7059	0.8041	0.6723	0.6667	0.6598	0.5773	0.6392	0.6667
[12]	0.7317	0.6316	0.6076	0.5937	0.6154	0.6835	0.7200	0.7164	0.7356	0.8471	0.6506	0.8046	0.6452	0.5714	0.6374	0.5952	0.6579	0.6944
[13]	0.7532	0.6591	0.6591	0.6563	0.6316	0.6957	0.5937	0.5517	0.7216	0.8276	0.7097	0.7879	0.8060	**0.7302**	0.6667	0.7397	0.6972	0.6667
[14]	0.7595	0.6667	0.6458	0.6866	0.6316	0.7229	0.7013	0.7632	0.7184	0.8132	0.7174	0.7290	0.7879	0.7188	0.6667	0.7429	0.6984	0.7302
Ours	**0.8276**	**0.7200**	**0.6835**	**0.7805**	**0.7126**	**0.8046**	**0.8395**	**0.8791**	**0.8000**	**0.9176**	**0.7551**	**0.8696**	**0.8132**	0.7216	**0.7527**	**0.7865**	**0.7416**	**0.7708**

(SSDC Dataset).

TABLE III
$F1$ COMPARISON OF DETECTION PERFORMANCE BETWEEN RLS-DTS METHOD AND BASELINE METHODS

DSSC	Movie						News						Tweets					
	F→R	F→V	R→F	R→V	V→F	V→R	F→R	F→V	R→F	R→V	V→F	V→R	F→R	F→V	R→F	R→V	V→F	V→R
[8]	0.6129	0.5479	0.5867	0.4928	0.5169	0.6747	0.6452	0.3462	0.6765	0.4364	0.7297	0.6230	0.6230	0.6410	0.4074	0.5397	0.6250	0.5862
[9]	0.6809	0.5882	0.6364	0.6038	0.5750	0.6824	0.7529	0.5231	0.8090	0.6250	0.6555	0.6667	0.7105	0.6139	0.6133	0.5400	0.5897	0.5301
[10]	0.7273	0.6126	0.6757	0.6182	0.6111	0.6000	0.8780	0.6723	0.8571	0.6610	0.6727	0.6733	0.7912	0.6306	0.5085	0.6022	0.6747	0.6739
[11]	0.5882	0.5577	0.6377	0.5870	0.5652	0.6667	0.7761	0.6087	0.7800	0.5588	0.6957	0.3333	0.7353	0.6346	0.4912	0.5075	0.6588	0.6512
[12]	0.6970	0.5794	0.6575	0.5859	0.5591	0.6481	0.8831	0.6316	0.8049	0.6496	0.6792	0.6600	0.8140	0.6275	0.3846	0.4789	0.6535	0.5435
[13]	0.4151	0.5574	0.6389	0.6275	0.6383	0.6789	0.8571	0.6897	0.8493	0.6724	0.6408	0.5495	0.7714	0.6796	0.1875	0.1961	0.6735	0.5057
[14]	0.5902	0.5905	0.6479	0.5941	0.6327	0.6408	0.8000	0.6897	0.8395	0.6723	0.6526	0.5745	0.5846	0.6735	0.2328	0.2174	0.5641	0.3478
Ours	**0.7848**	**0.7089**	**0.7013**	**0.7525**	**0.6667**	**0.7105**	**0.9195**	**0.7660**	**0.8750**	**0.7865**	**0.7901**	**0.7826**	**0.9333**	**0.6829**	**0.7105**	**0.6909**	**0.6923**	**0.7532**

(DSSC Dataset).

TABLE IV
$F1$ COMPARISON OF DETECTION PERFORMANCE BETWEEN RLS-DTS METHOD AND BASELINE METHODS

DSDC	Fang [1] → RNN [2]						RNN [2] → VAE [3]						VAE [3] → Fang [1]					
	M→N	M→T	N→M	N→T	T→M	T→N	M→N	M→T	N→M	N→T	T→M	T→N	M→N	M→T	N→M	N→T	T→M	T→N
[8]	0.3265	0.4314	0.4138	0.4314	0.5758	0.5763	0.3600	0.5070	0.4776	0.5294	0.5143	0.4828	0.7500	0.5122	0.5176	0.5588	0.5412	0.6607
[9]	0.6000	0.6216	0.6786	0.7200	0.6889	0.6265	0.6400	0.5634	0.6789	0.6542	0.6275	0.6481	0.6667	0.6325	0.6250	0.6667	0.6374	0.7317
[10]	0.6329	0.6410	0.7312	0.7952	0.7071	0.7216	0.6542	0.6588	0.6667	0.6000	0.6168	0.6724	0.5510	0.5825	0.5455	0.6207	0.6422	0.6364
[11]	0.5979	0.6392	0.6400	0.6885	0.6742	0.5882	0.6000	0.5974	0.6422	0.6606	0.6275	0.6607	0.6667	0.6667	0.6667	0.6555	0.5500	0.5507
[12]	0.6579	0.7105	0.7342	0.7895	0.6947	0.6829	0.6095	0.5526	0.6126	0.6122	0.6239	0.6435	0.4731	0.6047	0.6055	0.6422	0.5882	0.5510
[13]	0.6512	0.6567	0.7327	0.7463	0.7037	0.6737	0.7606	0.6575	0.7018	0.6800	0.6667	0.6724	0.5155	0.5060	0.4167	0.3768	0.6606	0.6415
[14]	0.6667	0.6579	0.7423	0.7105	0.7059	0.7229	0.7158	0.6374	0.6972	0.6726	0.6607	0.6723	0.5909	0.6047	0.4000	0.3509	0.5918	0.6522
Ours	**0.7648**	**0.8675**	**0.7532**	**0.8736**	**0.7819**	**0.8414**	**0.7826**	**0.6939**	**0.7292**	**0.6889**	**0.6931**	**0.7059**	**0.7174**	**0.7143**	**0.6753**	**0.7750**	**0.6667**	**0.8101**

(DSDC Dataset).

B. Results and Analysis

The comparison of the detection performance between the proposed RLS-DTS method and the baseline methods in the SSDC, DSSC, and DSDC datasets is presented in Table II to Table IV. M, T, and N represent Movie, Tweets, and News forms of the cover texts. F, R, and V represent the Fang [1], RNN [2], and VAE [3] steganography schemes. **Bold** is the best detection performance ($F1$ score).

The results in Table II to Table IV show that the RLS-DTS method can improve the performance of transformed datasets to varying degrees. It is worth noting that the performance of the BERT-based methods does not exhibit a notable advantage over the non-BERT-based methods, and in certain cases, it even falls behind the non-BERT-based methods. This is because in distribution-transformed steganalysis, the ability to adapt to the altered distribution is more crucial than capturing semantic features. When employing BERT for fine-tuning, the distribution of the original data diverges from those of the transformed data, and the model is difficult to adapt to the distribution well. At the same time, it is evident that the performance of the comparative methods on the SSDC and DSSC datasets is generally higher than that of the DSDC dataset. Nevertheless, there are instances where the performance of some methods on the SSDC and DSSC datasets is relatively subpar, for example [13] in the Tweets (R → F) dataset, the $F1$ score is very low. This is because the $F1$ value becomes lower due to the lower value of P or R. The existing methods only effectively capture the distribution of cover texts or stego texts, resulting in a diminished $F1$ value. Considering that there is a distribution transformed in the dataset, effective detection needs to capture the distribution of cover texts and stego texts. Therefore, we opt for using the $F1$ score as the performance evaluation metric for experiments.

IV. CONCLUSION

Considering the impact of distribution-transformed data on the performance of steganalysis, this letter proposes a novel RLS-DTS method to address this issue. This method uses the interaction between the observation space and the steganalyzer to capture the distribution of the original data. Then RLS-DTS uses the policy gradient method to strengthen the characteristics of the transformed data and optimize steganalysis features extraction. Experiments demonstrate that the RLS-DTS method exhibits excellent detection performance in distribution-transformed scenarios. Additionally, it provides a more autonomous and efficient solution for steganalysis.

REFERENCES

[1] T. Fang, M. Jaggi, and K. Argyraki, "Generating steganographic text with LSTMs," 2017, *arXiv:1705.10742*.

[2] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y. Zhang, "RNN-Stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1280–1295, May 2019.

[3] Z. Yang, S. Zhang, Y. Hu, Z. Hu, and Y. Huang, “VAE-Stega: Linguistic steganography based on variational auto-encoder,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 880–895, 2021.

[4] X. Zhou, W. Peng, B. Yang, J. Wen, Y. Xue, and P. Zhong, “Linguistic steganography based on adaptive probability distribution,” *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 5, pp. 2982–2997, Sep./Oct. 2021.

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

[6] Y. Yang, M. Lei, J. Wang, and B. Liu, “A SVM based text steganalysis algorithm for spacing coding,” *China Commun.*, vol. 11, no. 1, pp. 108–113, 2014.

[7] S. Samanta, S. Dutta, and G. Sanyal, “A real time text steganalysis by using statistical method,” in *Proc. IEEE Int. Conf. Eng. Technol.*, 2016, pp. 264–268.

[8] Z. Yang, Y. Huang, and Y. Zhang, “A fast and efficient text steganalysis method,” *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 627–631, Apr. 2019.

[9] Z. Yang, K. Wang, J. Li, Y. Huang, and Y. Zhang, “TS-RNN: Text steganalysis based on recurrent neural networks,” *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1743–1747, Dec. 2019.

[10] Z. Yang, Y. Huang, and Y. Zhang, “TS-CSW: Text steganalysis and hidden capacity estimation based on convolutional sliding windows,” *Multimedia Tools Appl.*, vol. 79, pp. 18293–18316, 2020.

[11] H. Yang, Y. Bao, Z. Yang, S. Liu, Y. Huang, and S. Jiao, “Linguistic steganalysis via densely connected LSTM with feature pyramid,” in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2020, pp. 5–10.

[12] W. Peng, J. Zhang, Y. Xue, and Z. Yang, “Real-time text steganalysis based on multi-stage transfer learning,” *IEEE Signal Process. Lett.*, vol. 28, pp. 1510–1514, 2021.

[13] J. Zou, Z. Yang, S. Zhang, S. Rehman, and Y. Huang, “High-performance linguistic steganalysis, capacity estimation and steganographic positioning,” in *Proc. Int. Workshop Digit. Watermarking*, 2021, pp. 80–93.

[14] Y. Xu, T. Zhao, and P. Zhong, “Small-scale linguistic steganalysis for multi-concealed scenarios,” *IEEE Signal Process. Lett.*, vol. 29, pp. 130–134, 2022.

[15] Y. Wang, R. Zhang, and J. Liu, “V-A3tS: A rapid text steganalysis method based on position information and variable parameter multi-head self-attention controlled by length,” *J. Inf. Secur. Appl.*, vol. 75, 2023, Art. no. 103512.

[16] J. Wen, Y. Deng, J. Wu, X. Liu, and Y. Xue, “Lifelong learning for text steganalysis based on chronological task sequence,” *IEEE Signal Process. Lett.*, vol. 29, pp. 2412–2416, 2022.

[17] J. Yang, Z. Yang, J. Zou, H. Tu, and Y. Huang, “Linguistic steganalysis towards social network,” *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 859–871, 2023.

[18] C. Yang, X. Luo, J. Lu, and F. Liu, “Extracting hidden messages of MLSB steganography based on optimal stego subset,” *Sci. China Inf. Sci.*, vol. 61, no. 11, pp. 1–3, 2018.

[19] Y. Xue, L. Kong, W. Peng, P. Zhong, and J. Wen, “An effective linguistic steganalysis framework based on hierarchical mutual learning,” *Inf. Sci.*, vol. 586, pp. 140–154, 2022.

[20] Y. Xue, B. Yang, Y. Deng, W. Peng, and J. Wen, “Domain adaptational text steganalysis based on transductive learning,” in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2022, pp. 91–96.

[21] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[22] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, 2019, Art. no. 9.

[24] P. Ladosz, L. Weng, M. Kim, and H. Oh, “Exploration in deep reinforcement learning: A survey,” *Inf. Fusion*, vol. 85, pp. 1–22, 2022.

[25] X. Mo, S. Tan, W. Tang, B. Li, and J. Huang, “ReLOAD: Using reinforcement learning to optimize asymmetric distortion for additive steganography,” *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1524–1538, 2023.

[26] W. Tang, B. Li, M. Barni, J. Li and J. Huang, “An automatic cost learning framework for image steganography using deep reinforcement learning,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 952–967, 2021.

[27] T. Zhang, M. Huang, and L. Zhao, “Learning structured representation for text classification via reinforcement learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 6053–6060.

[28] R. Sutton, D. McAllester, and S. Singh, “Policy gradient methods for reinforcement learning with function approximation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, vol. 12, pp. 1057–1063.

[29] D. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.