CONTRASTIVE HYPERSPHERE FOR ONE-CLASS LINGUISTIC STEGANALYSIS

Yufei Luo[1], *Zhen Yang*[1], *Xin Xu*[2], *Yelei Wang*[1], *Ru Zhang*[1]

[1]Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Tsinghua University, Beijing 100084, China

ABSTRACT

Linguistic steganalysis aims to detect stego texts that conceal hidden information. Existing methods typically rely on large amounts of stego texts for supervised training. However, with the rapid advancement of large language models and steganographic generation techniques, the quality and imperceptibility of stego texts have significantly improved, making them increasingly difficult to obtain in real-world scenarios. To address the challenge of training solely with cover texts, we propose Contrastive Hypersphere for one-class Linguistic Steganalysis (CHLS). Specifically, the method constructs a hypersphere in the feature space to aggregate most cover texts and generates contrastive-negative samples through word-level random swapping as structured regularization signals. During training, the model pulls cover texts toward the hypersphere center while pushing negative samples away using a boundary-aware contrastive loss, thereby preventing representation collapse and capturing shared features. During inference, stego detection is performed by measuring the distance between a test text and the hypersphere center. Extensive experiments demonstrate that CHLS achieves strong detection performance without relying on any stego texts during training.

***Index Terms*—** Linguistic steganalysis, one-class classification, contrastive learning, hypersphere structure.

1. INTRODUCTION

Steganography based on large language models (LLMs) [1], [2] not only possesses powerful generative capabilities but can also produce stego texts that are nearly undetectable [3], [4]. To address this challenge, researchers have proposed various deep learning-based steganalysis methods to mitigate the potential risks of malicious use [5], [6], [7]. However, existing steganalysis models typically rely on the presence of a large number of stego texts in the training set and assume that the training and testing data follow an identical distribution (i.i.d.), in order to achieve high detection performance [8], [9]. In practice, however, obtaining authentic and representative stego texts is extremely challenging [10], [11], [12]. On one hand, steganographic content is highly concealed, making it difficult to identify and extract; on the other hand, the diversity and potential obscurity of steganographic algorithms make it hard for researchers to simulate corresponding stego data [13], [14]. Therefore, the assumption that a large amount of known stego data is available in the training set is rarely feasible in real-world.

To address the shortage of stego texts in training, researchers have proposed few-shot steganalysis methods[15], [16]. However, these methods still rely on a certain amount of labeled or unlabeled stego texts. Wu et al. [17] further explored an extreme scenario where no training data is available and adopted a test-time adaptation strategy to enhance stego text detection. With the advancement of NLP technologies, cover texts have become easier to obtain and their semantic features remain relatively stable. Therefore, we focus on a more practical problem: how to detect stego texts when only cover texts are available for training, namely, the One-Class Classification (OCC) problem.

Currently, the OCC problem in linguistic steganalysis has received little attention. In contrast, the anomaly detection field has extensively adopted OCC approaches, such as mapping normal samples into a hypersphere and minimizing its volume to enhance anomaly detection. However, due to the diverse styles and complex structures of cover texts, their distribution in the feature space is often scattered or fragmented. Therefore, directly applying these one-class classification methods to linguistic steganalysis makes it difficult to extract shared features of cover texts, which can lead to overfitting and representation collapse, where samples become overly clustered in the feature space and lose their discriminability.

To address the one-class classification (OCC) problem caused by the absence of stego texts during training, this paper proposes a method called Contrastive Hypersphere Linguistic Steganalysis (CHLS). Since the training set completely lacks stego texts and stego-specific features cannot be used for detection, CHLS constructs a discriminative boundary by learning the shared features of cover texts, enabling the identification of potential stego texts. Specifically, CHLS first constructs a hypersphere in the feature space that encloses most cover texts. During training, CHLS minimizes the distance between cover texts and the hypersphere center, encouraging the aggregation of cover texts in the feature space to learn their shared features. On the other hand, to prevent the model from overfitting the distribution of cover texts and avoid representation collapse, CHLS applies word-level random swapping to cover texts located outside the hypersphere boundary to generate contrastive-negative samples

Corresponding author: Zhen Yang (yangzhenyz@bupt.edu.cn)

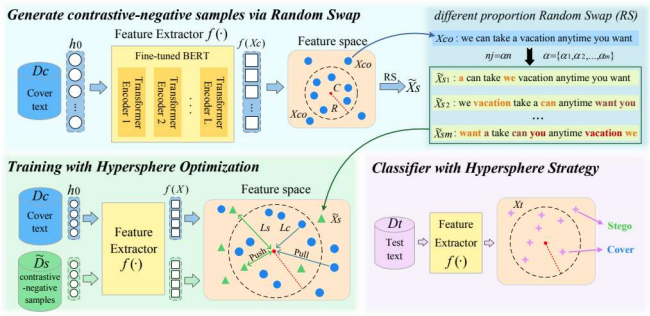


Fig. 1: Overall process of CHLS.

with slight semantic deviations. A boundary-aware contrastive loss is introduced to explicitly push these samples away from the center, incorporating them as structured regularization into the training process. Ultimately, the optimized hypersphere is used as a classifier: the model determines whether a test sample is stego text by calculating its distance to the center of the hypersphere.

2. PROPOSED METHOD

2.1. Feature Extractor and Hypersphere Construction

To extract the shared features and semantic information of cover texts, we use a pre-trained BERT model [20] as the feature extractor, denoted as $f(\cdot)$. Given the training set $D_c = (X_c, Y_c)$, where X_c consists of cover text sentences and Y_c is the corresponding label, each sentence is encoded by BERT to obtain token-level representations. The final output from the L-th layer is the semantic representation $H_L = \{h_{L,1}, h_{L,2}, \ldots, h_{L,n}\}$, where $h_{L,i}$ denotes the vector representation of the i-th word.

To capture the shared features of cover texts and clearly define a decision boundary, we construct a hypersphere S^c encompassing most cover texts. Specifically, the hypersphere center C is defined as the mean of all cover text features in the training set, and the radius R is determined by computing the squared Euclidean distances of these texts to the center C and selecting a specified quantile. The center C and radius R of the hypersphere are defined as:

$$C = \frac{1}{N_c} \sum_{i=1}^{N_c} f(X_c^i) \tag{1}$$

$$R = \mathrm{Q}_\nu(\|f(D_c) - C\|^2) \tag{2}$$

Here, X_c^i represents the i-th text in the training set D_c, where N_c is the total number of cover texts. D_c refers to the cover text training set, and $f(D_c)$ represents the feature representations extracted by $f(\cdot)$. The operator Q_ν selects the ν-th quantile of the squared Euclidean distances from all cover texts to the center C, where the quantile threshold ν controls the proportion of cover texts included within the hypersphere.

2.2. Contrastive-Negative Sample Generation

To prevent representation collapse, we introduce word random swap (RS) perturbations to cover texts located outside the hypersphere, generating contrastive-negative samples. Although these cover texts lie outside the hypersphere and are distant from the main cluster in the feature space, they typically remain semantically similar due to originating from the same corpus, and their deviation is often caused by syntactic "irregularities." The RS operation does not alter the vocabulary itself but disrupts word order to weaken such structural anomalies, thereby shifting the contrastive-negative samples closer to the main cluster in the feature space.

Specifically, for the hypersphere S^c constructed by the feature extractor, we compute the Euclidean distance $d(X_c^i, C)$ between each cover text X_c^i and the center C. We then identify those cover texts that fall outside the hypersphere:

$$d(X_{co}, C) > R \tag{3}$$

We denote the set of cover text samples X_{co}^i outside the hypersphere as $D_{co} = \{X_{co}^1, X_{co}^2, \ldots, X_{co}^M\}$.

We randomly select samples from D_{co} with probability β, and apply word-level random swapping [21] to generate contrastive-negative samples. For each selected sample X_{co}^j, we determine the number of words to swap using $n_j = \alpha n$, where n is the sentence length and α is the perturbation ratio. We define a set of perturbation ratios $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ to generate m distinct contrastive-negative variants. Varying perturbation strengths enhance the diversity of negative samples, enabling the model to learn from different patterns, which helps extract shared features of cover texts and prevents overfitting to specific patterns.

During generation, for each X_{co}^j, we randomly select n_j words, record their indices $\{i_1, i_2, \ldots, i_{n_j}\}$, and shuffle the order of words at those positions to create a new contrastive-negative sample $\tilde{X}_s$. This process yields:

$$\tilde{X}_s = \{x_{co}^1, x_{co}^2, \ldots, x_{co}^{i_{k1}}, x_{co}^{i_{k2}}, \ldots, x_{co}^{i_{kn_j}}, \ldots, x_{co}^n\} \tag{4}$$

where $\{i_{k1}, i_{k2}, \ldots, i_{kn_j}\}$are the positions of the perturbed words. By repeating this process, we generate a set of n_s contrastive-negative samples, denoted as $\tilde{D}_s = \{\tilde{X}_s^j, Y_s\}_{j=1}^{n_s}$, where Y_s represents the label assigned to each $\tilde{X}_s^j$.

2.3. Hypersphere-Based Optimization and Classification

During training, we optimize the hypersphere structure using both cover texts and contrastive-negative samples. The model pulls cover text features toward the center C via Boundary-aware Contrastive Loss (BCL), minimizing the hypersphere volume to capture shared features, while pushing contrastive-negative samples away to regularize and prevent representa-

Table 1: Comparison of different linguistic steganalysis methods across different corpora.

Method			TTem [17]			SESY [18] + CN			GE [5] + CN			SCL [19] + CN			**CHLS**		
corpus	Stego	bpw	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
Movie	AC	0.601	38.28	44.79	41.84	56.44	51.10	7.81	**63.72**	51.71	4.25	58.39	51.41	0.22	61.52	**60.03**	**59.57**
		2.144	40.27	45.91	43.52	55.49	49.50	6.44	**60.30**	49.69	8.03	54.97	51.24	0.19	58.80	**57.41**	**58.35**
		3.313	42.73	47.43	43.97	53.86	51.15	0.54	54.14	50.56	7.65	55.66	50.65	2.31	**57.13**	**55.46**	**54.94**
	ADG	5.081	49.99	52.05	45.33	51.35	50.60	3.71	51.12	51.75	6.45	52.21	51.50	8.45	**53.80**	**53.65**	**53.97**
Twitter	AC	0.572	37.57	45.88	38.22	53.19	51.55	7.21	48.21	51.80	4.65	**61.19**	52.16	10.58	58.97	**58.13**	**54.39**
		2.153	34.90	43.23	36.52	54.46	51.70	7.11	56.22	50.02	10.18	55.51	52.35	13.51	**57.90**	**56.39**	**54.41**
		3.458	39.49	47.15	39.13	53.10	51.85	9.55	54.72	50.65	10.61	54.47	51.40	1.19	**55.34**	**55.30**	**54.76**
	ADG	4.811	48.20	50.75	41.80	50.89	51.10	6.20	51.68	51.94	10.24	49.39	50.85	3.46	**53.88**	**53.82**	**55.69**
News	AC	0.545	37.98	46.30	39.10	58.62	49.35	2.49	62.01	48.95	13.87	59.66	51.10	0.39	**67.13**	**63.55**	**66.67**
		2.123	36.48	45.20	38.35	58.31	50.95	6.77	57.90	51.34	1.38	55.53	52.13	1.47	**64.54**	**59.54**	**55.62**
		3.490	40.89	46.98	39.55	54.11	49.75	2.29	51.07	51.12	0.21	52.85	51.56	0.38	**58.05**	**57.67**	**55.32**
	ADG	5.408	50.11	50.15	40.06	51.12	50.83	5.32	50.15	51.04	3.08	51.67	51.15	0.31	**53.36**	**53.50**	**58.62**
Average			41.41	47.24	40.61	54.25	50.79	5.45	55.10	50.88	6.72	55.12	51.46	3.54	**58.36**	**57.03**	**56.86**

tion collapse. The loss function is defined as follows:

$$L_{BCL} = \frac{1}{N_c}\sum_{i=1}^{N_c}\|f(X_c^i) - C\|^2 + \lambda \cdot \frac{1}{N_s}\sum_{j=1}^{N_s}\max(0, R^2 - \|f(\tilde{X}_s^j) - C\|^2) \tag{5}$$

where N_c is the number of cover texts, N_s is the number of contrastive-negative samples, and λ is a hyperparameter.

In the first part of BCL, the model minimizes the squared Euclidean distance between cover text representations $f(X_c)$ and the center C, encouraging the feature extractor to map cover texts closer to the center, enhancing their aggregation and helping extract shared features. The second part pushes contrastive-negative samples away from the hypersphere center to prevent representation collapse, while only applying repulsion to those within the hypersphere, avoiding excessive penalties for samples far from the center. This strategy ensures that the first part focuses on aggregating cover text features and extracting shared characteristics, while the second part fine-tunes the decision boundary, balancing both aspects to prevent overfitting and ensure effective feature extraction.

During inference, each test sample X_t is classified based on its Euclidean distance to the center C. If the distance is within the radius R, it is classified as a cover text; otherwise, as a stego text:

$$X_t \in \begin{cases} \text{stego text}, & \text{if } d(X_t, C) > R \\ \text{cover text}, & \text{if } d(X_t, C) \leq R \end{cases} \tag{6}$$

3. EXPERIMENTS

3.1. Experimental Setup and Evaluation Metrics

To assess the effectiveness of the proposed method, we follow the experimental setup of Gao et al. [22] and employ three benchmark corpora: Movie [23], Twitter [24], and News [25]. For each corpus, the training set consists of 10,000 cover texts generated by an LSTM model, while the test set contains 1,000 cover texts and 1,000 stego texts generated using the AC [26] and ADG [27] steganographic algorithms. All compared methods are trained and tested on the same datasets. The hypersphere radius R is determined using the quantile threshold $v = 0.8$, and the contrastive loss weight is set to $\lambda = 0.5$. Contrastive-negative samples are generated via controlled word-level substitutions, with the perturbation ratio α selected from 0.10, 0.15, 0.20, 0.25, 0.30, 0.35.

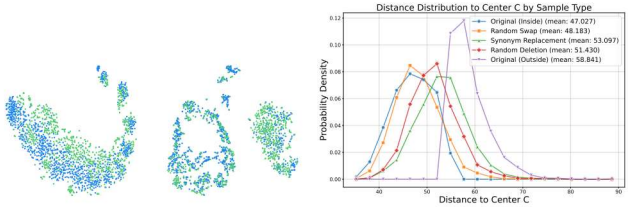
To evaluate the detection performance of CHLS, we adopt three metrics: Area Under the Curve (AUC), Accuracy (ACC), and F1 score. AUC measures the overall ability of the model to distinguish between cover and stego texts across different thresholds; ACC represents the classification accuracy at a specific threshold; and F1 combines precision and recall to assess the balance of the model in detecting stego texts.

3.2. Comparison with Linguistic Steganalysis and One-Class Classification Methods

We compared the proposed CHLS method with four existing linguistic steganalysis approaches: TTem [17], SESY [18], GE [5], and SCL [19]. For fair comparison, all baselines used the same feature extractor as CHLS, namely the pre-trained BERT-base-uncased [20], with other settings following their original configurations. TTem is a test-time adaptive method without training, relying on dynamic contrastive representations during inference. In contrast, SESY, GE, and SCL lack stego texts during training and tend to classify all test samples as cover, yielding a fixed accuracy of 50% and an F1 score of 0%. To improve performance, we introduced CHLS's contrastive-negative (CN) sample strategy into their training. Results are shown in Table 1.

Table 2: Ablation results under the Movie-AC-0.601 bpw, Twitter-AC-0.572 bpw and News-AC-0.545 bpw.

corpus	Movie			Twitter			News		
Method	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
w/o-HY	50.61	50.05	0.99	55.89	49.55	0.27	57.58	47.65	1.13
w/o-CN	42.11	51.40	16.46	31.76	49.95	0.19	44.80	50.10	15.99
w/o-RS	59.61	59.20	58.32	57.31	57.21	53.74	63.45	62.67	61.24
w/o-BC	57.34	55.00	45.44	56.89	56.12	52.35	59.01	60.99	56.14
CHLS	**61.52**	**60.03**	**59.57**	**58.97**	**58.13**	**54.39**	**67.13**	**63.55**	**66.67**



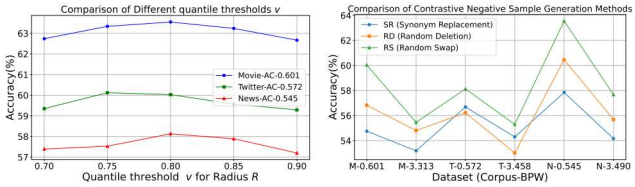
(a) SCL (b) CHLS (c) Distance to C by Sample Type

Fig. 2: (a–b) T-SNE visualizations of feature distributions under the News-AC-0.545 bpw scenario. Blue and green represent cover and stego texts, respectively. (c) Euclidean distance distributions to center C for different contrastive-negative sample types.

Compared to the test-time adaptive method TTem, CHLS achieves significantly better detection performance. Although TTem alleviates the lack of training data through entropy minimization during inference, it lacks explicit modeling of class boundaries in the semantic space, resulting in limited discriminative ability. In addition, CHLS outperforms SESY, GE, and SCL in most scenarios; however, in tasks such as Movie-AC-0.601, these baselines also achieve notable improvements when augmented with contrastive-negative (CN) samples, as CN helps capture stable shared features of cover texts, alleviate overfitting, and enhance discriminability. On the other hand, in semantically complex and structurally diverse scenarios such as Movie, the rigid hypersphere boundary of CHLS struggles to fully cover heterogeneous feature distributions, leading to slight performance drops. In addition, as shown in Figure 2, the t-SNE visualization reveals that SCL exhibits significant feature overlap, making it difficult to distinguish between cover and stego texts; in contrast, CHLS produces a more compact feature distribution and establishes a clearer one-class decision boundary. Moreover, as the embedding rate increases, the accuracy of all methods decreases, because the enlarged candidate space makes the generated distribution closer to cover text, reducing the statistical separability between stego and cover texts. [28]

3.3. Ablation Study

To evaluate CHLS, we performed ablation studies on its four key components: contrastive-negative samples, word-level random swapping, the hypersphere classifier, and the boundary constraint.We tested three variants: replacing the HYpersphere with a linear classifier (*w/o*-HY), applying



(a) (b)

Fig. 3: (a) Effect of different quantile thresholds ν on the hypersphere radius R across datasets. (b) Comparison of contrastive-negative sample generation methods on different corpus-BPW combinations.

synonym replacement to cover texts inside the hypersphere instead of word-level random swapping (*w/o*-RS) [21], and removing Contrastive-Negative samples entirely (*w/o*-CN), and removing the Boundary Constraint that penalizes all contrastive-negative samples equally (*w/o*-BC). As shown in Table 2, removing any component leads to a significant drop in performance, indicating that these modules work synergistically to enhance feature aggregation and discrimination.

We evaluated the impact of the quantile threshold v (ranging from 0.9 to 0.7) on model performance. As shown in the figure 3a, the model achieves the best results in most scenarios when $v = 0.8$, while the differences across values are relatively small, indicating that the model is robust to variations in this parameter within the tested range.

Additionally, we compared three contrastive-negative sample generation strategies, each applied to the same set of out-of-hypersphere cover texts: random swap (RS) used in CHLS, synonym replacement (SR), and random deletion (RD). To ensure fairness, all methods operated with the same word positions, perturbation ratio α, and training process. SR utilizes WordNet for synonym replacement. We conducted two experiments: first, comparing the strategies within the same training process, with results shown in Figure 3b; second, comparing their distances to the center C in the feature space, with results shown in Figure 2c. The results show that RS outperforms the other methods. The negative samples generated by RS are closest to the center, while those generated by SR are the farthest. This indicates that RS perturbations maintain similarity between the negative samples and the original cover text, which helps the model extract shared features.

4. CONCLUSION

This paper proposes CHLS for linguistic steganalysis in settings where stego texts are unavailable during training. By hypersphere modeling with boundary constraints and contrastive negatives, CHLS learns shared feature distributions of cover texts and improves detection. Future work will further optimize the architecture and objectives to better handle more complex stego texts.

5. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. U21B2020) and the 111 Project (Grant No. B21049).

6. REFERENCES

[1] J. Mei, Y. Ren, Y. Dai, and L. Wang, "Generation-based linguistic steganography with controllable security," *Chin. J. Netw. Inf. Secur.*, vol. 8, no. 3, pp. 53–65, 2022.

[2] J. Wu, Z. Wu, Y. Xue, J. Wen, and W. Peng, "Generative text steganography with large language model," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 10 345–10 353.

[3] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE security & privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[4] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[5] Q. Xu, R. Zhang, and J. Liu, "Linguistic steganalysis by enhancing and integrating local and global features," *IEEE Signal Processing Letters*, vol. 30, pp. 16–20, 2023.

[6] Y. Tang, Y. Wang, R. Zhang, and J. Liu, "Linguistic steganalysis via llms: Two modes for efficient detection of strongly concealed stego," *IEEE Signal Processing Letters*, 2024.

[7] J. Wang, Y. Wang, Y. Li, R. Zhang, and J. Liu, "Multi-classification of linguistic steganography driven by large language models," *IEEE Signal Processing Letters*, 2025.

[8] W. Peng, S. Li, Z. Qian, and X. Zhang, "Text steganalysis based on hierarchical supervised learning and dual attention mechanism," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3513–3526, 2023.

[9] Y. Xue, J. Wu, R. Ji, P. Zhong, J. Wen, and W. Peng, "Adaptive domain-invariant feature extraction for cross-domain linguistic steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 920–933, 2023.

[10] Z. Yang, Y. Luo, J. Yang, X. Xu, R. Zhang, and Y. Huang, "Class-aware adversarial unsupervised domain adaptation for linguistic steganalysis," *IEEE Transactions on Information Forensics and Security*, 2025.

[11] Y. Luo, Z. Yang, R. Zhang, and J. Liu, "Pseudo-label based domain adaptation for zero-shot text steganalysis," in *International Conference on Computational & Experimental Engineering and Sciences*. Springer, 2024, pp. 128–142.

[12] Y. Wang, R. Song, L. Li, R. Zhang, and J. Liu, "Dynamically allocated interval-based generative linguistic steganography with roulette wheel," *Applied Soft Computing*, p. 113101, 2025.

[13] Y. Wang, L. Li, Y. Tang, R. Zhang, and J. Liu, "Toward copyright integrity and verifiability via multi-bit watermarking for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[14] Z. Yang, Z. Xu, X. Xu, Z. Yang, and R. Zhang, "Linguistic steganalysis via text dual attention fusing statistical and multi-layer semantic features," *IEEE Signal Processing Letters*, 2025.

[15] H. Wang, Z. Yang, J. Yang, C. Chen, and Y. Huang, "Linguistic steganalysis in few-shot scenario," *IEEE Transactions on Information Forensics and Security*, 2023.

[16] Q. Niu, Z. Yang, Y. Luo, J. Zhao, and Y. Jiang, "Domain-assisted few-shot linguistic steganalysis in imbalanced class scenarios," *IEEE Signal Processing Letters*, 2025.

[17] J. Wu, X. Chen, J. Wen, W. Peng, and Y. Xue, "A test-time entropy minimization method for cross-domain linguistic steganalysis," *IEEE Signal Processing Letters*, 2024.

[18] J. Yang, Z. Yang, S. Zhang, H. Tu, and Y. Huang, "Sesy: Linguistic steganalysis framework integrating semantic and syntactic features," *IEEE Signal Processing Letters*, vol. 29, pp. 31–35, 2021.

[19] J. Wen, L. Gao, G. Fan, Z. Zhang, J. Jia, and Y. Xue, "Scl-stega: Exploring advanced objective in linguistic steganalysis using contrastive learning," in *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security*, 2023, pp. 97–102.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *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)*, 2019, pp. 4171–4186.

[21] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[22] L. Gao, J. Wen, Z. Zhang, G. Fan, and Y. Zhou, "Linguistic steganalysis via probabilistic weighted contrastive learning," *IEEE Signal Processing Letters*, 2024.

[23] 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*, 2011, pp. 142–150.

[24] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[25] A. Thompson, "Kaggle," https://www.kaggle.com/snapcrack/all-the-news/data, 2017.

[26] Z. M. Ziegler, Y. Deng, and A. M. Rush, "Neural linguistic steganography," *arXiv preprint arXiv:1909.01496*, 2019.

[27] S. Zhang, Z. Yang, J. Yang, and Y. Huang, "Provably secure generative linguistic steganography," *arXiv preprint arXiv:2106.02011*, 2021.

[28] 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, 2020.