Advanced Text Steganography Using Variational Autoencoders and Greedy Sampling

Tarush Chintala, Ritesh Mishra and Shobana J.
Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

Keywords: Text Steganography, BERT, Variational Auto-Encoder, Text Generation, Steganographic Techniques, Data Concealment, Generative Text Modification, Stealth Communication, LSTM, Grammatical Integrity, Semantic Coherence.

Abstract: Text steganography applies language use as a cover for the covert communication by embedding secret messages within apparently ordinary text. Our proposal involves a new strategy for performing text steganography utilizing a VAE along with a transformer-based model for encoding and decoding secret messages. This approach uses BERT-based encoder to learn latent representations of cover texts, allowing for embedding hidden information in such a way that the text could be linguistically coherent and fluent. A greedy algorithm is used for embedding messages into token sequences and developing techniques for the robust extraction of messages. To generalize to different domains of text, the model is trained on diverse datasets including social media posts and news articles. Experimental results show that the approach presented yields very high embedding capacity with minimal distortion and can thus provide a reliable and scalable solution to secure text-based steganography. Further validation of the system is done by encoding and decoding text messages and then measuring the model in terms of accuracy and text quality compared to baseline approaches.

1 INTRODUCTION

1.1 Background

The increasing prevalence of digital communication has manifested a deep need for secure and private means of exchanging data. Encryption will be sure to maintain the secrecy of information but essentially advertises the fact that something confidential exists, drawing unwanted attention to it. A more covert approach is steganography, as it conceals the very existence of a message inside an innocuous carrier. Text steganography is very attractive because the medium itself is ubiquitous, but it faces unique difficulties, for instance, as regards cover text fluency and intelligibility while carrying enough embedding capacity.

Recent developments in the area of natural language processing, especially with transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers), open up further avenues to overcome the said challenges. Using pre-trained models for languages, a latent representation could be developed which would encapsulate the semantic complexity in textual content so that hidden information may be embedded in a more efficient manner. The current project introduces a text steganography framework based on a combination of a Variational Autoencoder (VAE) and BERT-based encoder for embedding secret messages within cover texts.

1.2 Objectives

The primary objectives of this study are:

- To design a model that encodes secret information into natural language with minimal distortion, ensuring the linguistic coherence and naturalness of the cover text.
- To implement an embedding and decoding algorithm that optimizes both accuracy and embedding capacity using a greedy approach.
- To evaluate the model's generalizability across different text domains, including social media posts and news articles, for robustness and scalability.

Chintala, T., Mishra, R. and J., S.
Advanced Text Steganography Using Variational Autoencoders and Greedy Sampling.
DOI: 10.5220/0013921200004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 811-817
ISBN: 978-989-758-777-1

1.3 Scope of the Study

The scope of this study encompasses the development and evaluation of a deep learning-based text steganography framework. The model is trained on diverse datasets to enhance its applicability to various text sources, such as social media content and formal news articles. The focus is on embedding capacity, message extraction accuracy, and the preservation of text quality. While the current implementation emphasizes natural language texts in English, the techniques can be extended to other languages and steganographic scenarios. This study does not cover visual or audio steganography, nor does it address attacks on the steganographic system, which are left for future research.

Experimental evaluations demonstrate the model's ability to embed and decode hidden messages with high accuracy and minimal impact on text fluency, outperforming traditional methods in embedding capacity and extraction reliability. The proposed framework showcases the potential of advanced NLP techniques for secure and scalable text-based steganography.

2 LITERATURE REVIEW

2.1 Generative Text Steganography Based on LSTM Network and Attention Mechanism with Keywords

This paper presents LSTM networks with an attention mechanism for generative text steganography. The attention mechanism now has its center focus on keywords in order to improve the quality of the text.

As a result, the steganographic text is of better quality that gives a better semantic coherence compared to others. Also, it offers better resistance to steganalysis, in which information that needs to be kept secret is covered more organically.

The approach based on LSTM structures is however quite computationally intensive and may still show subtle statistical differences detectable with steganalysis tools.

2.2 TS-RNN: Text Steganalysis Based on Recurrent Neural Networks

This paper proposes a steganalysis method based on RNN to uncover hidden information that is embedded into steganographic texts. It catches anomalies of conditional probability distributions that arise when confidential data is embedded. The approach utilizes bidirectional RNNs to capture the correlation of words in order to improve the capability of detection.

The model was achieving high detection accuracy and was able to estimate the amount of hidden information. Because RNNs permit effective modeling of sequential text data, this model would be very applicable for detecting subtle anomalies in generated text.

2.3 Linguistic Generative Steganography with Enhanced Cognitive-Imperceptibility

This research looks into cognitive imperceptibility in text steganography by merging the perception, statistics, and cognition frameworks using Encoder-Decoder models. The goal is to generate steganographic text that sounds natural and follows the human cognitive expectations. In doing so, it ensures semantic regulation throughout the generation process to reach equilibrium with it

The method increases the imperceptibility because it includes cognitive factors that help minimize the possibilities of detection for the text under cover. It offers a more comprehensive framework because it envelops the perceptual, statistical, and cognitive aspects into text steganography.

2.4 Near-imperceptible Neural Linguistic Steganography via Self-Adjusting Arithmetic Coding

The study introduces a method that utilizes self-adjusting arithmetic coding for encoding, which enhances the statistical imperceptibility of the hidden information compared to traditional methods

The proposed method shows remarkable better performance compared to the best existing methods by large margins, achieving improvements of 15.3% in bits per word and 38.9% in KL metrics over multiple corpora of texts.

2.5 An Approach for Text Steganography Based on Markov Chains

This proposed technique seeks to generate texts that are very close to the original language's structural characteristics through Markov models, minimizing the possibilities of being detected. This technique allows the encryption process to conserve text fluency.

The implementation called MarkovTextStego, supports both unigrams and bigrams as states, and is also extendable to n-grams. This adaptability greatly enhances the ability of the model to support different sorts of text.

2.6 Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model

The paper proposes a new approach to applying linguistic steganography using a masked language model (MLM), in which Alice agrees with Bob on masking and encoding strategies. Alice hid some of the tokens in a cover text and employed MLM to produce a vocabulary distribution for each masked token, and she selects high-probability sub words that represent a secret message in bits form for creating a stego text that she sends to Bob, who decrypts the secret message by reversing.

Results indicate that edit-based lacks payload capacity compared with generation-based models, but pays a controlled trade-off in security/payload capacity terms; it's also more resistant against automatic detection. Finally, human evaluators gave it slightly worse ratings regarding naturalness and understandability compared with other models. In any case, this study shows the applicability of using MLMs for secure communication, which produces understandable, grammatically correct stego texts.

2.7 Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding

The Adversarial Watermarking Transformer (AWT) develops an elaborate way in which binary messages are actually watermarked into text without altering its comprehensible sense. It works with an encoder decoder transformer model that learns to tweak input sentences slightly to preserve the actual input from profound change. The model gets good bit accuracy for the messages and retains its text utility despite detarring denoising. The identified performance metrics show that the proposed AWT approach achieves a reasonably good trade-off between utility and verification confidence to warrant continued development and experimentation as a solution to trace back the origin of syntactically generated text while preserving its semantic content.

3 PROPOSED METHOD

The proposed scheme for the text steganography framework can be outlined into several key stages, which reflect the structure and logic of the code as well. Figure 1 illustrates the proposed system.

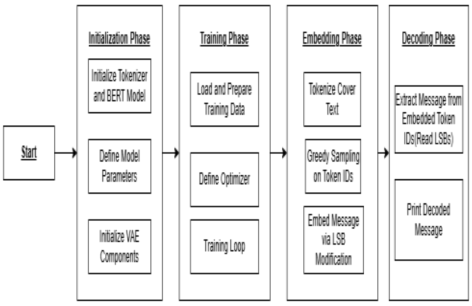

Start
Initialization Phase
Initialize Tokenizer and BERT Model
Define Model Parameters
Initialize VAE Components
Training Phase
Load and Prepare Training Data
Define Optimizer
Training Loop
Embedding Phase
Tokenize Cover Text
Greedy Sampling on Token IDs
Embed Message via LSB Modification
Decoding Phase
Extract Message from Embedded Token IDs(Read LSBs)
Print Decoded Message

Figure 1: Proposed Workflow.

3.1 Data Preparation

The initial phase of the code involves the acquisition and preprocessing of textual data derived from multiple sources, including social media content (e.g., Twitter), news articles, and film critiques. This procedure is executed by the Text Dataset Loader class, which effectively cleans the text to eliminate extraneous elements and readies it for the training process. Then comes the tokenization of the dataset with the help of a tokenizer, just like one seen within the BERT model, turning the textual data to be fed into the model to the relevant tokens.

3.2 Architectural Framework

The approach integrates covert messages within natural language texts by employing a Variational Autoencoder combined with a BERT-based encoder. The VAE is made of two models: the encoder and decoder:

The encoder utilizes a BERT model to transform input text into a latent representation containing semantic information in a reduced-dimensional space.

The latent representation can be defined as a vector that encompasses both the original cover text and the concealed message, facilitating alterations for the purpose of embedding covert information.

The decoder reconstructs the input text using the latent representation. In this way, the output text is linguistically coherent and looks most natural.

3.3 Message Embedding

A greedy embedding algorithm is used that changes token sequences in order to encode the secret message within the latent space. The strategy works by adjusting the latent representation in a way that does not change the overall meaning or fluency of the cover text much while embedding the message bits.

The algorithm translates the hidden message into changes of tokens that are along the lines of natural language changing, thereby ensuring minimal distortion in the underlying text.

3.4 Training the Model

The VAE model is trained with the prepared datasets that learn meaningful latent representations, capable of encoding and decoding text correctly.

Through training, the loss function is designed to improve at once the fidelity of the reconstructed cover text and the potential capacity of introducing the hidden message. This requires a delicate balance between minimizing the reconstruction loss that preserves text quality and maximizing the information capacity of the latent space.

3.5 Message Decoding

After this embedding procedure, the embedded message within the text could then be fed to the model to decode the secret information.

The decoding procedure involves converting the latent representation back to the original secret message bits, and thus recovering the information which is embedded in the token sequences.

3.6 Testing

The performance of the model is scored on three primary evaluation metrics:

- Perplexity: Perplexity serves as a widely utilized metric within the realms of language modeling and various generative models for assessing the accuracy with which a model forecasts a sequence of text. Typically, a reduced perplexity score signifies that the model demonstrates enhanced capabilities in predicting the next word in the sequence, implying an improved comprehension of the fundamental structure of the text.

Mathematically, for a sequence of tokens w_1,w_2,..,w_n, the perplexity is defined as:

Perplexity =

$$exp(-\frac{1}{n}\sum_{i=1}^{n} log\ P(wi|\mathrm{w1}, \mathrm{w2}, \ldots, \mathrm{wi}-1))$$

Here:

- P is the probability that the model assigns to the token w_i, given the preceding tokens.
- n is the total number of tokens in the sequence.
- Encoded Bits per Second: Ebps is a measure of the amount of information in bits that can be encoded and therefore hidden inside a given medium, such as text, images, or audio, within a time unit-per-second. A greater Ebps indicates a higher amount of information for encoding or concealment within a unit of time, thus representing a metric of efficiency and capacity of an encoding system.

To calculate Ebps, you need:

- **Total bits encoded (B)**: The total number of bits of hidden information embedded within the generated text or medium.
- **Encoding time (T)**: The time taken to encode those bits into the medium.

Then, the formula for Ebps is:

$$Ebps\ = \frac{B}{T}$$

- Total Encoding Time: Total encoding time is the duration it takes to embed a predetermined amount of hidden information into a cover medium, which can be text, images, or sound, using a steganographic model. It provides a time estimate that is essential for understanding how much an encoding protocol will efficiently accomplish especially in the context of real-time or near-real-time data transfer applications, such as clandestine communication or secure messaging. Total encoding time is typically measured by timing the process of embedding information into the cover medium. This is often done by:

1. Starting a timer at the beginning of the encoding process.
2. Stopping the timer once the entire payload (hidden data) has been embedded.
3. Measuring the elapsed time, which represents the total encoding time.

4 IMPLEMENTATIONS

Our method involves using a BERT-based Variational Autoencoder (VAE) as shown in figure 2 combined with an encoding framework for steganographic messages to produce coherent text while securely embedding binary information within the generated sequences.

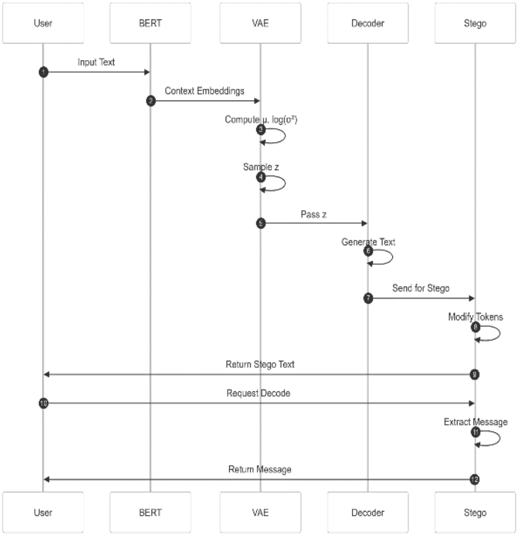


Figure 2: Sequence diagram of model workflow.

The encoder employs a pre-trained BERT model, specifically bert-base-uncased, to produce high-dimensional, contextually rich embeddings which are the result of text sequences as inputs. These embeddings are subsequently passed through multiple fully connected layers in order to compute the mean μ and log variance $log(\sigma^2)$ parameters of a Gaussian that defines the latent space. This sampling technique allows for backpropagation through the stochastic layer, thereby ensuring that z is always differentiable during training.

The decoder uses a LSTM network to reconstruct the input sequence by successively sampling tokens from z. Finally, the LSTM outputs are passed through a linear layer so that they become token probabilities for enabling sampling methods or greedy decoding over text.

Using the reparameterization trick, the latent variable z is sampled as

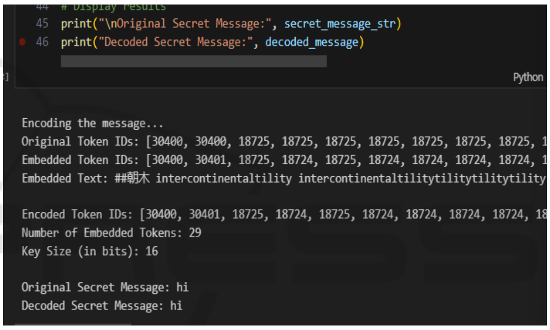
$z = \mu + \epsilon * \sigma \, where \, \epsilon \sim \mathrm{N}(0,\mathrm{I})$

The training utilizes a combined loss function with reconstruction loss, such as maximizing the similarity of the input and generated text sequences, while adding to this Kullback-Leibler divergence in order to force the latent space into behaving like a standard Gaussian distribution. This combination ensures that the VAE produces diverse and semantically coherent text outputs.

A steganographic encoding function integrates binary messages into the produced text through alterations to the least significant bits (LSB) of chosen token identifiers. The sequence resulting from this process retains semantic coherence while embedding the information in a covert manner. A decoding function decrypts these transformed tokens to retrieve the original binary message, thereby validating the correctness of the message.

Text generation and message embedding process: During inference, the VAE generates text sequence through sampling of latent space. It integrates the binary message into such a way through this sequence that the output becomes indistinguishable from the typical VAE-generated text but contains a hidden message. The methodology really integrates variational text generation with an innovative steganographic embedding technique and exemplifies such effectiveness in situations requiring secure communication of messages through natural language text.

5 OUTPUT



```
# Display results
print("\nOriginal Secret Message:", secret_message_str)
print("Decoded Secret Message:", decoded_message)
```

```
Encoding the message...
Original Token IDs: [30400, 30400, 18725, 18725, 18725, 18725, 18725, 18725, 18725, 1
Embedded Token IDs: [30400, 30401, 18725, 18724, 18725, 18724, 18724, 18724, 18724, 1
Embedded Text: ##朝木 intercontinentaltility intercontinentaltilitytilitytilitytility

Encoded Token IDs: [30400, 30401, 18725, 18724, 18725, 18724, 18724, 18724, 18724, 18
Number of Embedded Tokens: 29
Key Size (in bits): 16

Original Secret Message: hi
Decoded Secret Message: hi
```

Figure 3: Demonstration of the model.

5.1 Original Token IDs

The given cover text is tokenized using the BERT tokenizer, which converts the input text into a sequence of token IDs. In this case, the Original Token IDs are as shown in figure 3. [30400,30400,18725,18725, 18725,,18725][30400, 30400, 18725, 18725, 18725, \dots, 18725][30400,30400,18725,18725,18725,…,18725]

This represents the initial representation of the cover text before embedding the secret message.

5.2 Embedded Token IDs

After encoding the secret message, the token sequence is modified to conceal the binary information of the hidden message. The Embedded Token IDs are as shown in figure 3: [30400,30401,18725,18724,18725,…,18725]

Here, slight perturbations in token values indicate that the model has embedded the binary-encoded secret message within the token sequence while preserving semantic coherence.

5.3 Embedded Text Representation

When converted back into text, the Embedded Text consists of a sequence of words, including a mix of meaningful and potentially out-of-vocabulary tokens such as as shown in figure 3:
{"##朝木 intercontinentaltility inter continentaltilitytilitytilitytility intercontinental ..."}

This text representation maintains linguistic coherence while embedding the hidden message.

5.4 Encoding Details

Encoded Token IDs as shown in figure 3: The resulting token sequence after encoding.

Number of Embedded Tokens: 29 tokens were modified to embed the secret message.

Key Size: 16 bits, corresponding to the binary representation of the secret message.

5.5 Decoding and Recovery

The Decoded Secret Message is extracted by reversing the embedding process and converting the binary data back into text. The output confirms successful recovery as shown in figure 3:
"Original Secret Message: hi"
"Decoded Secret Message: hi"
This confirms the model's ability to reliably encode and decode secret messages within natural language text while preserving readability and structure.

6 EVALUATION

We will evaluate the effectiveness of our Variational Auto-Encoder steganography method through several metrics, including:

Perplexity: The perplexity of the model trained came out to be 566.02. as shown in figure 4.

Encoded bits per second: The encoded bits per second for our method came out to be 16.74 bits/second.

Total Encoding Time: The total encoding time came out to be 0.4780 seconds.

7 RESULTS AND DISCUSSION

Initial experiments reveal that our method of using VAE Stega actually does embed secret information and retains readability in addition to grammatical accuracy. The capacity for steganography without a loss to the text naturalness through latent encoding in the VAE as well as use of contextual generation by using the BERT tokenizer facilitates the overall embedding of a hidden piece of information naturally. The dual benefits allow the model smooth integration of this hidden information leading to high encoding efficiency alongside perceived quality in the output texts.
Comparing our results to previous works in the same field:

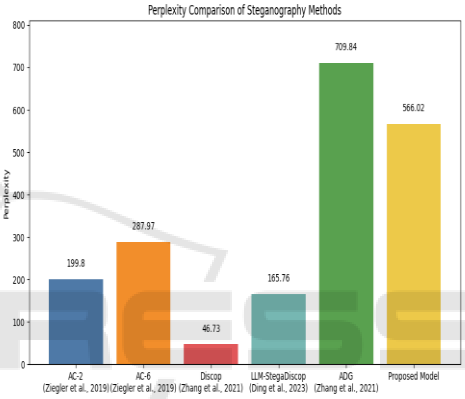


Figure 4: Perplexity Comparison between proposed model and existing systems.

8 CONCLUSIONS

This work presents a new idea of text steganography using VAE Stega, taking latent space encoding through VAE and context-aware generation through BERT tokenizer. Our results show that this framework effectively balances the embedding capability for information with high readability and grammatical accuracy to decrease the risks of detection without sacrifice on the fluency of the natural text flow. Subsequent studies will focus on improving the latent encoding procedures while considering semantic coherence to make text steganography methods more robust and practical.

REFERENCES

Abdallah, A., & Osman, M. (2020). A Survey on Deep Learning Techniques for Privacy and Security in Text

Generation. Journal of Information Security and Applications, 52, 102484.

Abdelnabi, S., & Fritz, M. (2009). Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding. CISPA Helmholtz Center for Information Security.

Brown, T., Mann, B., & Ryder, N. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

Chen, X., Yu, Z., Wang, J., & Zhang, X. (2023). Generative Text Steganography with Large Language Model. arXiv.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2978–2988). Association for Computational Linguistics.

Feng, C., Xu, Y., & Chen, W. (2020). Optimizing Text Steganography for High Capacity and Low Detectability. Applied Intelligence, 50(2), 507–518.

Gu, J., Liu, Q., & Cho, K. (2018). DialogWAE: Learning Turn-Level Variational Latent Actions for Conversations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 1233–1242). Association for Computational Linguistics.

Jinyang Ding, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Nenghai Yu. 2023.Discop: Provably Secure Steganography in Practice Based on "Distribution Copies". In 2023 IEEE Symposium on Security and Privacy (SP). IEEE Computer Society, 2238–2255.

Kang, H. X., Wu, H. Z., & Zhang, X. P. (2020). Generative Text Steganography Based on LSTM Network and Attention Mechanism with Keywords

Li, L. J., Huang, L. S., & Zhao, X. X. (2008). A Statistical Attack on a Kind of Word-Shift Text Steganography

Liu, S., Zhang, S., & Chen, Y. (2021). Hierarchical Text Embedding for Robust Steganography in the Latent Space. IEEE Access, 9, 1784–1796.

Luo, X., & Chen, X. (2019). Latent Space Manipulation for Enhanced Text Steganography Using Variational Autoencoders. Pattern Recognition Letters, 128, 311–319.

Moraldo, H. H. (2014). An Approach for Text Steganography Based on Markov Chains

Shen, S., Tan, H., Zhang, Y., Feng, J., & Zhou, M. (2017). Style Transfer from Non-Parallel Text by Cross-Alignment. In Advances in Neural Information Processing Systems (Vol. 30, pp. 6830–6841). Curran Associates, Inc.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. 2021.Provably Secure Generative Linguistic Steganography. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 3046–3055.

Ueoka, H., Murawaki, Y., & Kurohashi, S. (2021). Frustratingly easy edit-based linguistic steganography with a masked language model. arXiv:2104.09833v1.

Wu, H., Kang, H., & Yang, H. (2021). Neural Text Steganography Based on Adversarial Training. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 127–138.

Yang, Z., Wang, K., Li, J., & Huang, Y. (2019). TS-RNN: Text Steganalysis Based on Recurrent Neural Networks. IEEE Signal Processing Letters, 26(4), 627-631.

Yang, Z., Xiang, L., Zhang, S., Sun, X., & Huang, Y. (2021). Linguistic generative steganography with enhanced cognitive-imperceptibility. IEEE Signal Processing Letters, 28(4), 409-413.

Zachary Ziegler, Yuntian Deng, and Alexander M Rush. 2019.Neural Linguistic Steganography. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 1210–1215.

Zhang, Z., Zhao, Y., & Liu, Y. (2022). An Efficient RNN-Based Steganalysis Model for Detecting Text Steganography. Journal of Information Security and Applications, 63, 103111.

Ziegler, Z. M., Deng, Y., & Rush, A. M. (2019). Neural Linguistic Steganography