# Information Hiding for National Security

## Steganalysis and its Potential Applications for Intelligence

Oscar Hernandez

April 6, 2024

## Contents

**1 Introduction to National Security** **1**

1.1 Communication . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2 Intelligence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**2 Problem** **4**

**3 Methodology** **5**

3.1 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

3.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

**4 Discussion** **8**

4.1 Outlook . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

# 1 Introduction to National Security

This research paper addresses the national security motivations for detecting information hidden in text and advances the state of the art by fine-tuning a transformer architecture based on a distilled pre-trained large language model.

## 1.1 Communication

A communication system consists of a user Alice who transmits messages to a user Bob via a communication channel that is subject to eavesdropping by an adversarial user Carol who is assumed to know the design of the system by Kerkchoff's principle[1]. A (symmetric) secret communication system consists of an additional probability distribution $\mathcal{R}$ which is sampled to produce a shared secret key $k$, known to Alice who uses it to encrypt her message before transmission and to Bob who uses it to decrypt the message after reception but not to Carol; although Carol is assumed to know $\mathcal{R}$, the messages she intercepts appear to be a random sequence of characters. These communication channels are said to be *overt* because Carol can detect the presence of information by eavesdropping on the communication channel itself, although the contents of the intercepted messages are not intelligible.

A communication channel is *covert* if the contents and very presence of information is kept secret from the adversary. A *subliminal* communication channel [5] is a covert communication channel embedded within an overt communication channel which Alice can use overtly by sending a *cover* message directly or covertly by sending a *stego* message produced from a cover message by a *steganography* algorithm which hides the presence of information in such a way that Bob can recover it with a *steganalysis* algorithm which is not known to Carol. The strength of a subliminal communication system is determined by (algorithmic) hardness of *steganalysis*, the problem Carol faces in determining whether a given message is a cover message from the overt channel or a stego message from the embedded covert channel.

The next section addresses the practical applications in the intelligence community of *information hiding*, the arts of steganography and steganalysis.

[1] "The enemy knows the system." [4]

## 1.2 Intelligence

Communication privacy benefits from indirectness and obscurity, in analogy with the anonymous communication principles that underlie the onion network routing protocol of The Onion Router. The contents can be obfuscated by encryption. The mere use of encryption, however, implies some level of sensitivity that may be interpreted in tandem with metadata and other contextual information, e. g. in the case of a federal employee in a position of import to national security who sends encrypted messages via insecure channels like TikTok. One way to obfuscate messages that can be intercepted from social media companies by adversaries is to instead send the messages by other means, such as the chat feature of unpopular mobile phone applications for the Backgammon game which are less likely a priori to be intercepted by adversaries. If this obscurity is ever exploited, however, then a direction relationship may be presumed to exist between the sender and the recipient. The intended recipient of the message can be obfuscated by posting the messages publicly on social media platforms like X, formerly known as Twitter. The task is steganography: to transmit messages on public channels without appearing encrypted and while evading keyword filters and all other methods of detection in potential use by adversaries.

The central applications of steganography in counterintelligence are ensuring that messages intercepted by adversaries cannot be known to contain or not contain sensitive information. Guided on the hand by practical applications for intelligence, steganalysis enables the adversary to reliably distinguish cover text from text with information that has been hidden by steganography. Potential applications include the identification of hidden information in text corpora collected from publicly available information and intercepted signals. Steganalysis is posed as a data-driven problem in Section 2, for which a method is proposed in Section 3, and some implications for intelligence are discussed in Section 4.

## 2 Problem

In the prisoner's problem [5], a prisoner is allowed to communicate with his associate outside of the prison via written letters which are analyzed by the prison warden. The warden feels compelled to deliver a given message, e. g. a bible passage, unless she has a just reason to believe it may be used to commit a crime, e. g. an instruction to attack another inmate. This prison mail system thus uses an overt communication channel that is known to the adversarial warden and that could be host to a covert communication channel. For example, the prisoner could have arranged a code with his associate: bible passage $A$ is an instruction to start a fight at lunch-time and bible passage $B$ is an instruction to expect a shipment of contraband. The warden would forward these letters to the prisoner because they appear to be safe bible passages but, in doing so, she would enable covert communication since the bible passages covertly communicate illicit instructions. To learn to predict the probability that a given letter contains hidden information that connects bible passages to illicit activity, she would start by compiling a dataset of intercepted letters which she labels according to whether she believes they contain hidden information.

Data-driven text steganalysis methods are tasked with using historical lists of cover text and stego text in order to predict the probability in the future that a given message is cover text or stego text. The historical lists are typically combined into a single dataset where each message has a label which is 0 if it is cover text or $l$ if it is stego text generated by an unknown steganography algorithm which is common to all messages with the label $l$. Data-driven methods are trained on one subset of the data and have their performance evaluated on a disjoint subset of the data designated for testing. The performance of a model on a new dataset is likely to be similar to that on the testing set. The next section describes a data-driven model and its performance on a given dataset.

# 3 Methodology

Originally from the Kaggle competition [6], the proposed dataset consists of 1500 samples of cover text with label 0 and 1500 samples of stego text with label 1 (denoting the use of steganography algorithm 1 of the original 12) from [8]; the stego text was produced by a generative adversarial network (GAN), while the cover text was produced by a GAN, a recurrent neural network, and the Tina [2] Long Short-Term Memory neural network.

The testing set consists of 20% of this randomly-shuffled dataset. The task is to develop a supervised machine learning model for text steganalysis to read a message in the testing set and determine the probability that it belongs in class 0 or 1. The testing set is not used for model training, selection, or any other purpose besides a final evaluation of the model's performance. The training set consists of another 60% of the dataset and it is used by the model to approximate the optimal set of parameters. The validation set consists of the remaining 20% of the dataset; while it is not used directly to optimize the parameters, it is used to determine the optimal set of parameters witnessed by the model during training so that it is selected instead of the final set of parameters at the end of training which may have deteriorated after some point in time due to overfitting.

A neural network model is chosen based on the success of a state-of-the-art convolutional neural network *LS_CNN* for text steganalysis [7] and the recent advances in large language models based on the attention mechanism. While it is feasible to design a transformer architecture specifically for this task, pre-trained large language models offer the advantage that the weights learned for their original tasks may be useful for the task at hand.

The proposed model is a pre-trained *TinyBERT* which is densely connected to a binary classification layer. Its implementation is shown in Section 3.1 and the fine-tuned model is compared against a fine-tuned *LS_CNN* in Section 3.2.

### 3.1 Implementation

```python
import torch.nn as nn
from transformers import AutoModel, AutoConfig

class LS_TNN(nn.Module):

  def __init__(self, args, text_field=None):
    super(LS_TNN, self).__init__()
    self.args = args
    self.name = "ls_tnn"
    C = args.class_num
    model_name = "huawei-noah/TinyBERT_General_4L_312D"
    self.tinybert = AutoModel.from_pretrained(model_name)
    D = self.tinybert.config.hidden_size
    self.fc1 = nn.Linear(D, C)
    for param in self.tinybert.parameters():
      param.requires_grad = True
    for param in self.fc1.parameters():
      param.requires_grad = True

  def forward(self, input_ids, attention_mask=None, token_type_ids=None):
    outputs = self.tinybert(input_ids=input_ids, attention_mask=attention_mask, token_type_ids=token_type_ids)
    pooled_output = outputs.pooler_output
    logits = self.fc1(pooled_output)
    return logits
```

Listing 1: Definition of LS_TNN model as a class in Python

## 3.2 Results

The AUROC as a percentage of the maximum value of 1.00, typically desired to be greater than 82%, of the proposed model *ls_tnn* on the testing set is shown to be 99.84% on Figure 3.2 below.

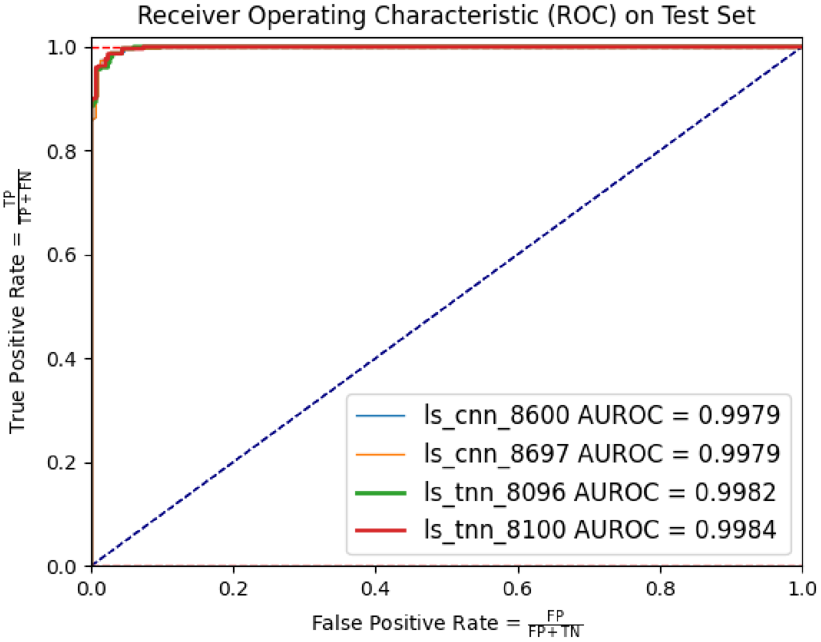

Figure 1: Receiver operating curves and the areas under the curves

This suggests that the model is capable of discriminating between cover text and stego text produced by a steganographic algorithm, and it does so modestly better than the state-of-the-art *LS_CNN* [7]. It required fewer epochs to fine-tune the TinyBERT than it did to train the CNN on the same dataset. It is likely that the attention mechanism in the transformer would fare better than convolutional neural networks at recognizing stego texts with long-range behavior produced by more exotic steganography and that the approach of fine-tuning language models will scale better as a result of the pre-training.

# 4 Discussion

The art of steganalysis can be practiced for intelligence collection to determine whether a given dataset contains hidden information. This leads to a *stego* arms race: knowledge of steganalytic methods may be exploited in the development of steganographic methods which are designed to be specifically resistant to the known methods of steganalysis, which can be exploited for the future development of steganalytic methods which are designed to specifically crack the known methods of steganography, ad infinitum. For this reason, substantial advances in steganalysis ought to be kept secret. Efforts that make use of steganography ought to devote resources to acquiring steganographic methods that are sufficiently resistant to all known methods of steganalysis, just as one would only use cryptographic methods that are resistant to all known methods of cryptanalysis.

One limitation of the method presented is that the stego text under consideration was produced by a single steganographic method based on GANs. Previous work [8] has shown that such methods scale well to multi-class classification. Given a dataset consisting of text messages annotated by whether it is cover text or by class numbers that correspond to steganography algorithms, it is likely that our method would perform well in detecting whether a given text message is cover text or which class it belongs to.

This paper described the relevance of text steganalysis for intelligence operations, and it showed that a fine-tuned transformer neural network model is capable of successfully performing the binary classification in a manner that is likely to perform similarly well on the $(k+1)$-ary classification task required by the use of $k$ different methods of steganography.

## 4.1 Outlook

The proposed method of text steganalysis can be applied to identify subsets of information that is hidden in text corpora of natural language and potentially other kinds of text[2] including, but not limited to: encrypted text, formal source code in any number of programming languages, and compiled binary software.

Since the predictions are probabilities instead of merely classes, the text corpora can be sorted according to the model's level of certainty that the corresponding documents contain hidden information.

Suppose there is a large dataset of messages together with the results of processing them with the same method of steganography. In that case, it may be possible to train a model on that dataset to recover the hidden information from another message that was produced by the same method.[3] This is possible, for example, if the method of steganography is known and very likely to be possible without this stringent requirement. This problem lies at the heart of information hiding, obscurity [3], artificial intelligence safety [1], and national security.

An efficient method of steganalysis can sift through haystacks of data to find potentially valuable needles that are designed with the express purpose of being invisible to humans and machines. These small collections of needles that are predicted with a high probability to contain hidden information can be given to analysts and to more specialized computer programs that may be specially designed to extract the hidden information.

[2] While not explored in this paper, the techniques of deep learning can also be used for steganalysis in images, video, computer network traffic, and other media of information.

[3] This principle is exploited frequently in cryptography, for example in the cryptanalysis of the Enigma machine with a relatively small number of samples.

# References

[1] Scott Aaronson. My AI safety lecture for UT Effective Altruism, 2023. Accessed: 2024-04-01.

[2] Tina Fang, Martin Jaggi, and Katerina Argyraki. Generating steganographic text with LSTMs. In Allyson Ettinger, Spandana Gella, Matthieu Labeau, Cecilia Ovesdotter Alm, Marine Carpuat, and Mark Dredze, editors, *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[3] Dusko Pavlovic. Gaming security by obscurity, 2012.

[4] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.

[5] Gustavus J. Simmons. *The Prisoners' Problem and the Subliminal Channel*, pages 51–67. Springer US, Boston, MA, 1984.

[6] Andrew Thomas. All the News Dataset. `https://www.kaggle.com/datasets/snapcrack/all-the-news`, 2017.

[7] Juan Wen, Xuejing Zhou, Ping Zhong, and Yiming Xue. Convolutional neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(3):460–464, 2019.

[8] Kaiguo Yuan, Yu Yang, Ziwei Zhang, and Juan Wen. Multi-task few-shot text steganalysis based on context-attentive prototypes. *Expert Systems with Applications*, 249:123437, 2024.