Encrypted Rich-data Steganography using Generative Adversarial Networks

Dule Shu
Carnegie Mellon University
Pittsburgh, Pennsylvania
dules@andrew.cmu.edu

Weilin Cong
The Pennsylvania State University
University Park, Pennsylvania
wxc272@psu.edu

Jiaming Chai
The Pennsylvania State University
University Park, Pennsylvania
jjc5958@psu.edu

Conrad S. Tucker
Carnegie Mellon University
Pittsburgh, Pennsylvania
conradt@andrew.cmu.edu

ABSTRACT

Steganography has received a great deal of attention within the information security domain due to its potential utility in ensuring network security and privacy. Leveraging advancements in deep neural networks, the state-of-the-art steganography models are capable of encoding a message within a cover image and producing a visually indistinguishable encoded image from which the decoder can recover the original message. While a message of different data types can be converted to a binary message before encoding into a cover image, this work explores the ability of neural network models to encode data types of different modalities. We propose the *ERS-GAN* (Encrypted Rich-data Steganography Generative Adversarial Network) - an end-to-end generative adversarial network model for efficient data encoding and decoding. Our proposed model is capable of encoding message of multiple types, *e.g.*, text , audio and image, and is able to hide message deeply into a cover image without being detected and decoded by a third-party adversary who is not given permission to access the message. Experiments conducted on the datasets *MS-COCO* and *Speech Commands* show that our model out-performs or equally matches the state-of-the-arts in several aspects of steganography performance. Our proposed ERS-GAN can be potentially used to protect the wireless communication against malicious activity such as eavesdropping.

CCS CONCEPTS

• **Security and privacy** → Human and societal aspects of security and privacy; **Privacy protections**;

KEYWORDS

steganography, neural networks, machine learning

ACM Reference Format:
Dule Shu, Weilin Cong, Jiaming Chai, and Conrad S. Tucker. 2020. Encrypted Rich-data Steganography using Generative Adversarial Networks. In *ACM Workshop on Wireless Security and Machine Learning (WiseML '20), July 13, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3395352.3402626

WiseML '20, July 13, 2020, Linz (Virtual Event), Austria

ACM ISBN 978-1-4503-8007-2/20/07.
https://doi.org/10.1145/3395352.3402626

1 INTRODUCTION

Wireless communication networks are used to transmit a considerable amount of private information. Due to the broadcast nature of wireless communications, eavesdropping attacks often choose wireless networks as a target. Although security measures such as cryptography and friendly jamming prevent direct eavesdropping on the communication, eavesdroppers can still infer messages using alternative techniques such as time analysis, dictionary attacks to break cryptographic keys, and signal cancellation to remove the impact of friendly jamming signals [4]. In case an eavesdropper successfully acquires the targeted messages in the wireless network, a second layer mechanism is needed to protect the secrecy of users' communication. In addition, the fast-emerging Low Probability of Intercept / Detection (LPI/LPD) communication requires messages to be securely and efficiently encoded such that the wireless communication activity remains secretive to a third-party adversary node [25]. One possible solution to enhancing the privacy of wireless communication is steganography. In general, steganography refers to the practice of concealing a piece of information in another piece of information. For example, in image-based steganography, a message is encoded in an image named the cover image before sent to the receiver. When an eavesdropper obtains a cover image containing an encoded message, he or she will consider the image as a regular image and be unaware of the encoded message unless a proper steganalysis on the image is taken. Steganography can also potentially be used for LPI/LPD communication for wireless networks, where, instead of hiding the communication activity, it hides the actual information being transmitted by showing the cover images as the apparent information being transmitted. Motivated by the potential utility of steganography in enhancing the privacy of wireless communication, we propose a deep neural network model for image-based steganography. Our model allows different types of message to be encoded by a cover image without increasing the file size of the image, which increases the efficiency of data transmission at a node.

In image-based steganography, an encoded message is converted to some sufficiently small perturbations of the pixel values of a cover image. The perturbed image (named the *stego image*) remains visually identical to the original image. If a mapping from the stego

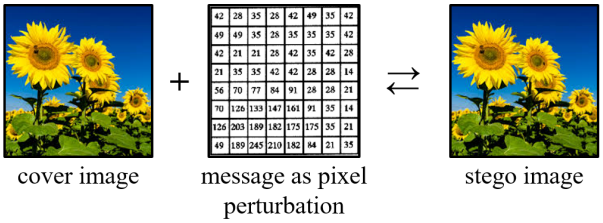


Figure 1: Steganography using cover image.

image to the message is obtained, a secure transmission of the secret message can be achieved. A mapping between a cover image, a message and a stego image using message as pixel perturbation is shown in Fig. 1. The successes in using autoencoders to encode information [8][1][11] and the successes in using adversarial models to synthesize realistic data [5][6] have motivated the use of neural network models for image-based steganography. In particular, Hayes *et al.* [7] proposes a neural network model that transmits encrypted random message using a cover image. The model consists of three modules: an encoder module that encodes the message into the cover image, a decoder module that decodes the message from the stego image, and an adversary module that implements adversarial training of the network model for performance improvement. The random message can be recovered via decryption from the output of the decoder. A similar model for image-based steganography is proposed by [27]. This model follows the three-module design of [7], and introduces a extra noise layer between the encoder and the decoder to mimic the noise in stego image transmission. The model is shown to have robustness in decoding accuracy in the presence of noise.

The two aforementioned models choose random binary data with limited length as the secret message. Although in general, a message of different data types can be converted to a binary message before encoded into a cover image, it is possible that there exists a more efficient way of message encoding for steganography. Since artificial neural networks have been widely used to learn the representations of different data types such as text [26], audio [15], image [2] and video [24], we choose to use a neural network model as an autoencoder for efficient and versatile featurization. By combining the autoencoder for message featurization and a neural network model for steganography, we propose the Encrypted Rich-data Steganography Generative Adversarial Network - a new end-to-end trainable generative adversarial network for image-based steganography. Our model adopts the general structure proposed in [27], but is capable of transmitting multiple types of message ranging from text, audio to image. In order to improve secrecy, we include an encryption module and a decryption module inside our encoder and decoder, respectively. This design preserves the secrecy of the encoded message even if the stego image from our model has been detected by a steganalysis method, as the decryption key is not contained in the model. We demonstrate the performance of our model via numerical experiments. A comparison between our model and some other existing approaches shows that our model has similar or higher performance in several aspects of steganography performance. The contributions of our work are summarized as follows.

- We propose an image-based steganography model that works with multiple types of message data, from text, audio and image.
- Our model introduces a more efficient and versatile message encoding method to improve the capacity of steganography.
- We achieve improved level of secrecy using built-in encryption and decryption modules in our model.
- We report evaluation result of message decoding accuracy by human observers in an experiment involving human participants. To the best of our knowledge, such type of evaluation has not yet been reported by other state-of-the-art works in image-based steganography.

The rest of this paper is organized as follows. In Section 2, we review related works to image-based steganography and explain the difference between our model and some other state-of-the-art models. In Section 3, we provide formal formulation of the image-based steganography problem and our proposed solution. We show the experiment results in Section 4. Section 5 concludes the main points of the paper and discusses future work.

2 RELATED WORKS

2.1 Steganography

A wide variety of steganography methods have been proposed in recent years, which can be categorized into three types: Least-Significant Bit algorithm, Content-Adaptive algorithm, and DL Steganography algorithm.

The Least-Significant Bit (LSB) algorithm can encode the secret message inside the cover image with low computational cost. For convenience and simplicity in implementation, the LSB algorithm hides the secret message to the least significant bits in the channel pixel of an image. The modification of the LSB algorithm is often called ±1-embedding [23][20], because it randomly adds or subtracts the value of 1 from the channel pixel, so that the last bits would match the needed value. The LSB algorithm is relatively easy to detect by a steganalyzer because it systematically alters the statistical distribution of the image [17].

In order to overcome this limitation, the Content-Adaptive steganography algorithms are proposed that utilize a more strategic pixel manipulation technique: carefully picking pixels in the cover image according to the secret message such that the distortion of the encoded image is minimized. In particular, HUGO [16] defines a distortion function domain by assigning costs to pixels based on the effect of embedding some information within a pixel. The distortion is measured by computing weights for local pixel neighborhoods, which results in lower distortion costs along edges and in high-texture regions. WOW [9] encodes information into a cover image according to textural complexity of a region. WOW penalizes distortion to predictable regions of the image using a bank of directional filters and shows that the more complex the image region is, the more pixel values will be modified in this region. S-UNIWARD [10] is similar to WOW but can be used for embedding in an arbitrary domain. Despite the diverse implementation details, the ultimate goals of these Content-Adaptive algorithms are the same, *i.e.,* they are all devoted to minimizing a distortion function by encoding the

secret message into the noise areas or areas with complex texture, and by avoiding the smooth areas. The Content-Adaptive steganography algorithms are computationally expensive and can be easily detected by deep neural network-based steganalyzer.

The Deep-Learning (DL) based steganography can encode the secret message into the cover image with low computational complexity and are less likely to be detected. S-GAN [21] is proposed for generating image-like containers based on Deep Convolutional Generative Adversarial Networks [18]. S-GAN consists of a generator network which produces realistic looking image from noise, a discriminator network which classifies whether an image is synthetic or real, and another discriminator network which determines if an image contains secret messages. Although S-GAN reduces the detection rate of steganalysis algorithms, stego-images generated by S-GAN are warping in semantic and are more easily to draw attention than natural images. Instead of generating image-like containers using GAN, EDS [19] proposes an encoder-decoder based model to conceal a color secret image into a color cover image. Without relying on hand-crafted algorithms, EDS can automatically learn how to merge the cover image and the secret image together using gradient descent. However, stego-images generated by their models are distorted in color and are easily recognized by well trained CNN-based steganalyzer due to the large capacity. To overcome this limitation, IS-GAN [3] is proposed to only hide the secret image in the Y channel of the cover image.

3 METHOD

3.1 Problem formulation

In this work, we aim to hide a secret message (text, audio, or image) inside a cover image without being detected by the adversary. A neural network model is developed to achieve this objective. An overview of the network model is shown in Fig. 2. Our model adopts the general structure proposed in [28], but is capable of transmitting multiple types of message ranging from text, audio to image. In order to improve secrecy, we include an encryption module and a decryption module inside our encoder and decoder, respectively. This design further preserves the secrecy of the encoded message even in case the stego image from our model has been detected by a steganalysis method, as the decryption key is not contained in the model.

The network model consists of an encoder E, a decoder D, an adversary module A, and a noise layer N. The encoder E takes a cover image I_{co} and a secret message M_e as input, and outputs a stego image I_{st}. The stego image I_{st} is passed through a noise layer N to produce a noisy stego image $\tilde{I}_{st}$. $\tilde{I}_{st}$ is sent to the decoder where a message $\hat{M}_d$ is decoded from $\tilde{I}_{st}$. The adversary module A takes the cover image I_{co} and the stego image I_{st} as input, and predicts the probability p that the input image is a stego image. The overall loss function of the steganography model is defined as follows.

$$\mathcal{L}(M_e, I_{co}) = \lambda_1 \ell_{encoder} + \lambda_2 \ell_{decoder} + \lambda_3 \ell_{st} + \lambda_4 \ell_{adv}, \quad (1)$$

where λ_1- λ_4 are user-specified weights (In practice, we recommend the following choice of weights: $\lambda_1 = 0.7$, $\lambda_2 = \lambda_4 = 1$, $0.001 \leq \lambda_3 \leq 0.1$.), $\ell_{\text{encoder}} := \|I_{st} - I_{co}\|_2^2$ is the encoder loss, $\ell_{\text{decoder}} := \|\hat{M}_e - \hat{M}_d\|_2^2 + \ell_{\text{msg}}(M_e, M_d)$ is the decoder loss ($\hat{M}_e$ is the feature representation of M_e), $\ell_{st} := \log(A(I_{st}))$ is a loss function to penalize the encoder for the detection of a stego image, and $\ell_{adv} := \log(1 - A(I_{co})) + \log(A(I_{st}))$ is the adversary loss. Note that ℓ_{st} is equivalent to the second term in ℓ_{adv} in terms of function value. The reason they are defined seperately in $\mathcal{L}$ is that, in the optimization problem formulated in Eq. (4), ℓ_{st} is parameterized by E, D while ℓ_{adv} is parameterized by A. We use the following cross entropy loss function to define the message loss $\ell_{\text{msg}}(M_e, M_d)$,

$$\ell_{\text{msg}} := -\sum_{t=1}^{L_M} \sum_{j=1}^{C} x_j^{(t)} log(p(\tilde{x}_j^{(t)} = 1|x)), \quad (2)$$

where C is the number of words in the dictionary, L_M is the length of the sentence x, $x_j^{(t)}$ is the jth element in the one-hot vector representation of the tth word in sentence x, and $p(\tilde{x}_j^{(t)} = 1|x)$ represents the probability that the tth word in the decoded sentence $\tilde{x}$ is the jth word in the dictionary, conditioned on x. For image and audio message, ℓ_{msg} is defined as an l_2-norm:

$$\ell_{\text{msg}} := \|M_e - M_d\|_2^2 \quad (3)$$

Formally, the network model is proposed as a solution to the following optimization problem.

$$\min_{E,D} \max_{A} \mathbf{E}_{M_e, I_{co}}(\mathcal{L}(M_e, I_{co})), \quad (4)$$

where E, D, and A denotes the sets of weights in the encoder, the decoder, and the adversary module, respectively.

3.2 Encoder

The proposed network model can encode three types of message data: text, audio and image. We choose these three data types for our steganography model because they are commonly-used information media to convey messages. For text and audio data which have a sequential pattern, a Bi-directional Recurrent Neural Network (BRNN) model is used as the feature extraction module. For image data which has a spatial pattern distributed in 2D or 3D space (depending on the number of color channels), a Convolutional Neural Network model is used to extract features.

As shown in Eq. 5, the feature extraction BRNN for text and audio computes the feature vector using some nonlinear function f of the two final hidden states in both directions. Let $M_e = \{x_t\}_{t=1}^{L}$ be an input sequence of length L to the BRNN encoder, the calculation of the feature $\hat{M}_e$ is described by the following equations.

$$\begin{aligned} \overrightarrow{h_t} &= f(\overrightarrow{W} x_t + \overrightarrow{V}\,\overrightarrow{h}_{t-1} + \overrightarrow{b}), \\ \overleftarrow{h_t} &= f(\overleftarrow{W} x_t + \overleftarrow{V}\,\overleftarrow{h}_{t-1} + \overleftarrow{b}), \\ \hat{M}_e &= [\overrightarrow{h}_L; \overleftarrow{h}_L], \quad t \in \{1, 2, ..., L\}, \end{aligned} \quad (5)$$

where the superscripts $\rightarrow$ or $\leftarrow$ indicates that the variable is associated with the BRNN units in a particular direction, the hidden state h_t is calculated by a nonlinear function f using its predecessor hidden state h_{t-1}, the tth element x_t from the input sequence M_e, and the weights W, V and b. The feature vector $\hat{M}_e$ is calculated as the concatenation of the final hidden states in both directions. In the network model proposed in this paper, f is chosen as the Gated Recurrent Unit (GRU).

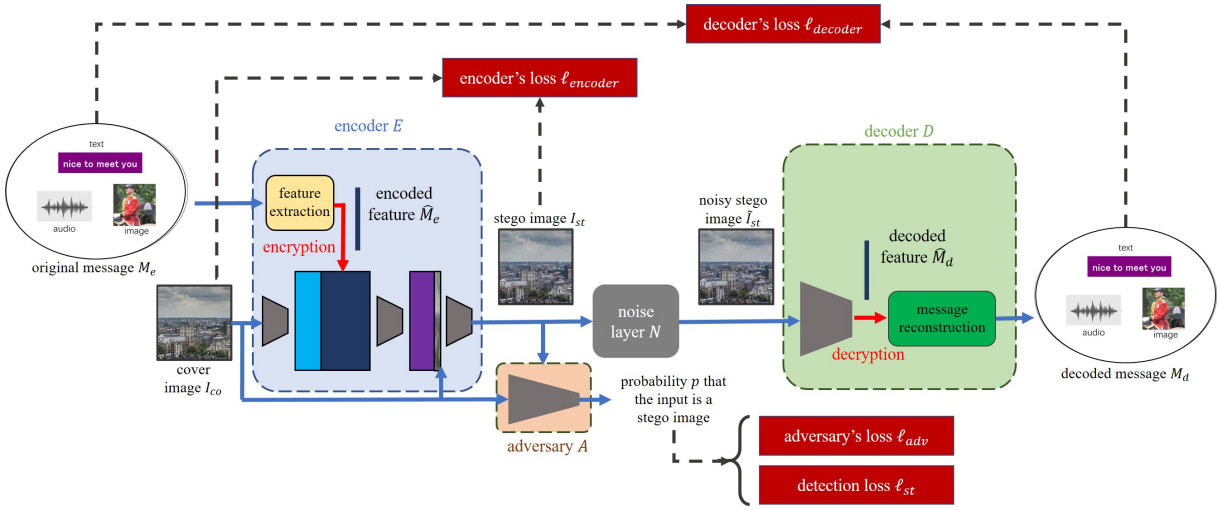


Figure 2: Model overview.

In the feature extraction module for image message, the input message M_e is sent through a sequence of convolutional layers to produce a 3D tensor whose height and width are the same as those of the cover image I_{co}. This 3D tensor is used as the message feature $\hat{M}_e$ to produce the stego image I_{st}.

To demonstrate our model's capability of ensuring secrecy, we apply an encryption process to the feature vector of text message. Given a feature vector $\hat{M}_e$, the encryption process generates a permutation that alters the sequence of elements in the $\hat{M}_e$. The vector obtained after permutation, denoted as $\hat{M}_e^P$, is combined with the cover image I_{co} instead of $\hat{M}_e$. When the corresponding feature vector $\hat{M}_d^P$ is decoded from the noisy stego image $\tilde{I}_{st}$ in the decoder, the elements in $\hat{M}_d^P$ is recovered to a vector $\hat{M}_d$ with the original sequence as in $\hat{M}_e$ via an inverse process of the permutation. The permutation is known only to the sender and the receiver, and is a necessary step to decode the message from the stego image. Thus, it is used as a key for message encryption and decryption.

3.3 Decoder, Adversary and noise Layer

Given a noisy stego image $\tilde{I}_{st}$ as input, the decoder in our model produces a decoded message M_d. In the case when the message M_e is sequential data (text and audio), the decoder first computes a 1-dimensional vector $\hat{M}_d$ from the input $\tilde{I}_{st}$. The vector $\hat{M}_d$ has the same length as the message feature $\hat{M}_e$ in the encoder, and is sent to another BRNN model to reconstruct the encoded message M_d. The BRNN mode for message reconstruction has the same structure as the BRNN model in the encoder. The main difference between the decoder's BRNN and the encoder's BRNN is that the hidden state h_0 in the decoder is set as $h_0 = \hat{M}_d$, while the hidden state h_0 in the encoder is initialized with random number. In the GRU model used by the BRNN decoder, the output of a recurrent neural network unit is the hidden state. Therefore, we have $M_d = \{\tilde{x}_t\}_{t=1}^{L} = \{[\overrightarrow{h}_t; \overleftarrow{h}_t]\}_{t=1}^{L}$, where $\overrightarrow{h}_t$, $\overleftarrow{h}_t$ are the hidden states of decoder's BRNN. The adversary module is a binary classifier used to detect stego image. and applying specially designed filters.

4 EXPERIMENTS

4.1 Model Implementation

The cover images used in network training and testing are from the COCO dataset [14]. In the steganography experiment, three types of message data, text, audio and image, are separately encoded in a cover image. For text data, we use sentences of image description from the COCO dataset in network training. For audio data, we use the Speech Commands dataset [22] to provide audio samples for training. This dataset has 65,000 audio clips of voices from around 1,000 different people with a length of 1 second. For image data, we use the same dataset for cover images to obtain message samples. For each type of data, 2,000 samples of cover images and message samples are used in network training. Adam [12] is used as the optimization algorithm to update the model weights. The parameters for Adam are set as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is chosen as 0.001, and the batch size is chosen as 32. The test dataset has a size of 1,000 data samples. The GAN model is trained for 10 epochs using a computer with an Intel Core i7-8750H CPU, 16 GB RAM, and a GeForce RTX 2060 GPU. The total training time is 3589 seconds. After training, the sampled loss function values on validation data is as follows. $\ell_{encoder} = 0.0046$, $\ell_{decoder} = 0.0005$, $\ell_{st} = 0.9033$, and $\ell_{adv} = 1.1923$. In terms of model complexity, the generator has 21,459,915 number of weights, which can be saved as a Python dictionary object file of 190 MB. Such file size could be considered large if running on a smart phone or other mobile device of similar computational power. Therefore, reducing the scale of a trained model for implementation on a smart phone/mobile device is one of the directions of future work.

4.2 Performance Evaluation

Three metrics are used to evaluate the performance of our network model. The first metric is *capacity*. We define capacity as the size of the encoded message divided by the size of the cover image. The size of a message or a cover image is calculated as the size of the array that represents the data. The second metric is *secrecy*. We evaluate

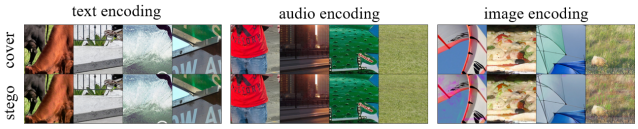


Figure 3: Comparison of cover images and stego images.

secrecy using the detectability of a stego image, *i.e.*, how difficult it is for a human observer or a computer classifier to identify a stego image. The third metric is *accuracy*, which is defined as the fraction of the accurately decoded elements in message M_d. More details of model evaluation is provided in the following part of the section by performance metrics.

4.2.1 Capacity. For image data as the encoded message, both the message and the cover image have the same format which is a color image of size 128×128×3. Therefore, by dividing the size of message by the size of cover image, we get the capacity of our model as 1.000. For text and audio data as the encoded message, the length of the original message M_e varies from sample to sample. Therefore, we use the fixed length of message feature $\hat{M}_e$ to define the message size. In our experiments, the length of $\hat{M}_e$ is chosen as $L = 49,152$, which is equal to the size of the cover image ($49152 = 128 \times 128 \times 3$). Therefore, the capacity of our model for text and audio data is also 1.000. Compared with HiDDeN which operates at a capacity of 0.203 for binary message steganography, our model shows a larger capacity and a higher versatility in terms of encoding different types of message.

4.2.2 Secrecy. We evaluate the secrecy of our model using the detectability of stego image. To show the detectability from human observers' perspective, we compare some cover images samples with the corresponding stego images in Fig. 3. As shown in the figure, the stego images are highly similar to the cover images for all encoded message type, which makes them difficult to detect by human observers. We compare our stego images with the stego images generated by HiDDeN in Fig. 4. As shown in the figure, the stego images generated by our model and by HiDDeN have similar qualities. To evaluate the detectability of stego image to a computer algorithm, we train the steganalyzer ATS [13] to differentiate stego images from cover images. Given a set of cover images and a steganography method, we first use the steganography method to generate a set of stego images. Then, we create a training dataset for ATS using the cover images and the stego images. The numbers of cover images and stego images are chosen as 250. The detection rates of different steganography methods by ATS is shown in Table 1. When the weights are unknown, our method has a detection rate of 52%, which is slightly higher than HiDDeN but lower than the other benchmark steganography methods. When the weights are known, however, our model has a detection rate of 53%, which is the lowest amount all benchmark methods and is significantly lower than HiDDeN. This result indicates that our method has a robust performance in hiding stego images from computer classifiers.

4.2.3 Accuracy. We show the accuracy of message decoding in Fig. 5, where the decoding errors and the message examples are provided. For text data, we use the fraction of incorrect words in a

Figure 4: Comparison of cover image and stego images generated by our method and HiDDeN.

Table 1: Detection rates of different methods by ATS.

Method	Bits per pixel	Detection rate(%)
HUGO	0.200	70
WOW	0.200	68
S-UNIWARD	0.200	68
HiDDeN (weight known)	0.203	97
HiDDeN (weight unknown)	0.203	51
Our method (weight known)	1.000	53
Our method (weight unknown)	1.000	52

decoded sentence to define the decoding error. For image and audio data, we use the l_2 distance between the original message M_e and the decoded message M_d to measure the accuracy. In addition, we evaluate the audio data's decoding accuracy using human listeners' predictions. In each prediction task, a human participant listens to an original audio sample and a corresponding decoded audio sample. Without knowing the ground truth, the participant makes a prediction on which sample he or she hears is the original one. 1,500 prediction results are collected from Amazon Mechanical Turk, out of which 886 predictions are correct, which yields a success rate of $886/1500 \approx 59\%$ for human's prediction.

5 CONCLUSION

In this paper, we propose a deep neural network model for steganography. The network model is able to hide message in the form of text, audio or image under a cover image, send the cover image to a receiver, and decode the message from the image for the receiver. To ensure secrecy in the message transmission against potential neural network-based steganalysis model, an adversarial classifier is used in network training to increase similarity between a stego image and a non-stego image. In addition, permutation of the message feature is introduced as a cryptography method to avoid the message from being decoded by potential adversaries. To allow encoding of different types of data, we propose an autoencoder model to extract features from a message and recover data from the decoded message. Compared with other methods for steganography, our method is able to encode a wider variety of messages, and has a higher capacity and secrecy. For future work, we will work to enrich the types of message in our steganography model. We

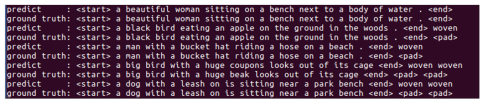
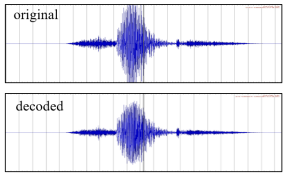
Message type	Decoding error	Example of messages
text	0.017	
audio	0.0003	
image	0.0057	

Figure 5: Decoding accuracy and message examples for different data types and models.

will also study methods to increase the level of secrecy by using multiple layers of encoding. Implementing our proposed method in an actual wireless communication network is another direction of our future work.

ACKNOWLEDGEMENTS

This work is funded by the ARL CRA: MACRO: Models for Enabling Continuous Reconfigurability of Secure Missions grant W911NF-13-20045, and the Air Force Office of Scientific Research (AFOSR) grant FA9550-18-1-0108. Any opinions, findings, or conclusions found in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha T Karhunen. 2015. Bidirectional recurrent neural networks as generative models. In *Advances in Neural Information Processing Systems*. 856–864.

[2] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* 54, 10 (2016), 6232–6251.

[3] Shiqi Dong, Ru Zhang, and Jianyi Liu. 2018. Invisible steganography via generative adversarial network. *arXiv preprint arXiv:1807.08571* (2018).

[4] Song Fang, Tao Wang, Yao Liu, Shangqing Zhao, and Zhuo Lu. 2019. Entrapment for wireless eavesdroppers. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2530–2538.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[7] Jamie Hayes and George Danezis. 2017. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*. 1954–1963.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[9] Vojtěch Holub and Jessica Fridrich. 2012. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*. IEEE, 234–239.

[10] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1.

[11] Myeongjun Jang, Seungwan Seo, and Pilsung Kang. 2019. Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning. *Information Sciences* 490 (2019), 59–73.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Daniel Lerch-Hostalot and David Megías. 2016. Unsupervised steganalysis based on artificial training sets. *Engineering Applications of Artificial Intelligence* 50 (2016), 45–59.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[16] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*. Springer, 161–177.

[17] Jiaohua Qin, Xuyu Xiang, and Meng Xian Wang. 2010. A review on detection of LSB matching steganography. *Information Technology Journal* 9, 8 (2010), 1725–1738.

[18] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:cs.LG/1511.06434

[19] Atique ur Rehman, Rafia Rahim, M Shahroz Nadeem, and Sibt ul Hussain. 2017. End-to-end Trained CNN Encode-Decoder Networks for Image Steganography. arXiv:cs.MM/1711.07201

[20] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. 1994. A digital watermark. In *Proceedings of 1st International Conference on Image Processing*, Vol. 2. IEEE, 86–90.

[21] Denis Volkhonskiy, Ivan Nazarov, Boris Borisenko, and Evgeny Burnaev. 2017. Steganographic generative adversarial networks. *arXiv preprint arXiv:1703.05502* (2017).

[22] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).

[23] Raymond B Wolfgang and Edward J Delp. 1996. A watermark for digital images. In *Proceedings of 3rd IEEE International Conference on Image Processing*, Vol. 3. IEEE, 219–222.

[24] Chenggang Yan, Yunbin Tu, Xingzheng Wang, Yongbing Zhang, Xinhong Hao, Yongdong Zhang, and Qionghai Dai. 2019. STAT: spatial-temporal attention mechanism for video captioning. *IEEE transactions on multimedia* (2019).

[25] Shihao Yan, Xiangyun Zhou, Jinsong Hu, and Stephen V Hanly. 2019. Low probability of detection communication: Opportunities and challenges. *IEEE Wireless Communications* 26, 5 (2019), 19–25.

[26] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine* 13, 3 (2018), 55–75.

[27] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. HiDDeN: Hiding Data With Deep Networks. arXiv:cs.CV/1807.09937

[28] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 657–672.