electronics

MDPI

Article

A Coverless Audio Steganography Based on Generative Adversarial Networks

Jing Li, Kaixi Wang * and Xiaozhu Jia

College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
* Correspondence: kxwang@qdu.edu.cn

Abstract: Traditional audio steganography by cover modification causes changes to the cover features during the embedding of a secret, which is easy to detect with emerging neural-network steganalysis tools. To address the problem, this paper proposes a coverless audio-steganography model to conceal a secret audio. In this method, the stego-audio is directly synthesized by our model, which is based on the WaveGAN framework. An extractor is meticulously designed to reconstruct the secret audio, and it contains resolution blocks to learn the different resolution features. The method does not perform any modification to an existing or generated cover, and as far as we know, this is the first directly generated stego-audio. The experimental results also show that it is difficult for the current steganalysis methods to detect the existence of a secret in the stego-audio generated by our method because there is no cover audio. The MOS metric indicates that the generated stego-audio has high audio quality. The steganography capacity can be measured from two perspectives, one is that it can reach 50% of the stego-audio from the simple size perspective, the other is that 22–37 bits can be hidden in a two-second stego-audio from the semantic. In addition, we prove using spectrum diagrams in different forms that the extractor can reconstruct the secret audio successfully on hearing, which guarantees complete semantic transmission. Finally, the experiment of noise impacts on the stego-audio transmission shows that the extractor can still completely reconstruct the semantics of the secret audios, which indicates that the proposed method has good robustness.

Keywords: audio steganography; coverless steganography; GAN; covert communication; information hiding

Citation: Li, J.; Wang, K.; Jia, X. A Coverless Audio Steganography Based on Generative Adversarial Networks. *Electronics* **2023**, *12*, 1253. https://doi.org/10.3390/electronics12051253

Academic Editors: Krzysztof Szczypiorski and Xiao Yu

Received: 22 December 2022
Revised: 12 February 2023
Accepted: 28 February 2023
Published: 5 March 2023

1. Introduction

The many rapidly developed internet technologies make our lives more convenient. However, in some cases, there is a strong demand for a private secure transmission that can prevent a message from being disclosed. Information hiding is an important technology to protect private information. Currently, information hiding technology has been developed into two branches: digital steganography and digital watermarking technology [1]. The former is mainly used for covert communication, and the latter is mainly used for copyright protection. Steganography is an art and science that hides a secret in one or more covers and then the stego-objects are transmitted in public channels without being noticed.

At present, the digital covers for steganography include texts, images, audios, videos, and network traffic. With the prevalence of social networks supporting short audio files [2] and the popularization of VoIP services [3], audio has become a popular media. This prevalence is one feature of the best covers. Our hearing system is not very sensitive, and audio files have large redundancy. Therefore, audio has become a main cover for digital steganography. Traditional audio steganography mainly utilizes the redundancy of audio to embed a secret [4]. This kind of embedding generally performs some modifications to the cover, which, in turn, leads to changes to the cover features.

A third party can detect the existence of a secret in such a stego-audio by analyzing the changes to the statistical features. More importantly, with the emergence of various

new detection tools, especially those based on the neural networks and deep-learning technologies, the detection accuracy has become very high [5]. Table 1 shows the detection accuracy of different steganalysis methods based on deep-learning technologies for the traditional steganography methods by cover modification with the difference embedding rates. The high detection accuracy shows that the traditional audio steganography by cover modification has lost its utility. Compared to steganography by cover modification, steganography by cover generation aims to generate a natural cover to cope with the detection issues caused by cover modification.

Generative Adversarial Networks (GANs) [6] are an important technology to realize steganography by cover generation, and they have been applied in audio covers. Ref. [7] first proposed audio steganography based on adversarial examples to modify an existing audio cover. Ref. [8] encoded the cover audio and the secret audio to generate a stego-audio automatically based on a GAN. Ref. [9] proposed a framework based on a GAN to achieve optimal embedding for audio steganography in the temporal domain. Ref. [10] employed LSBM steganography to embed a secret into a generated cover with high security.

In [11], the secret audio and cover audio were preprocessed using the time-domain zero-padding method and then input into the encoder to generate the stego-audio, which improved the security greatly. Ref. [12] proposed end-to-end audio steganography, and the encoder encoded the secret message into the audio cover. However, the encoder generated a modified vector of the audio sample value instead of stego-audio, which greatly reduced the distortion caused by message embedding.

Although these previous works utilized GANs for audio steganography, they performed modifications on an existing or generated cover, and a third party can still detect the existence of a secret in the stego-audio by analyzing the changes to the statistical features of the generated cover. As far as we know, there are currently no steganography methods that directly generate a stego-audio without a cover-audio.

Table 1. The detection accuracy of different steganalysis methods based on deep learning under the different embedding rates.

Steganalysis Methods	Steganography Methods	Embedding Rates	Accuracy
Spec-ResNet [13]	LSB-EE [4]	0.1	0.9151
		0.2	0.946
		0.3	0.9608
		0.5	0.9724
LARxNet [5]	SIGN [14]	0.1	0.9427
		0.2	0.9665
		0.3	0.9854
		0.5	0.9912
WASDN [15]	MIN [16]	0.1	0.9357
		0.2	0.9572
		0.3	0.9643
		0.5	0.9881
MultiSpecNet [17]	LSB-EE [4]	0.1	0.9433
		0.2	0.957
		0.3	0.9675
		0.5	0.9796

In this paper, a coverless audio steganography that does not perform any modification to an existing or generated cover is proposed. Herein, being coverless does not mean that there is no media to carry a secret, instead, it means that there is not any embedding operation on the existing or generated cover [18].

Consequently, the stego-audio is directly generated without a cover in this work. Therefore, the proposed method only needs to transmit the stego-audio. For attackers, there is no cover to refer to. This application scenario has better security than the traditional audio steganography by cover modification. The main contributions of this paper are summarized as follows:

(1) A coverless audio-steganography model is proposed based on the general audio synthesis framework WaveGAN [19] to directly synthesize a stego-audio instead of modifying an existing or generated audio to generate a stego-audio. A post-processing layer is replenished in the original WaveGAN generator to improve the quality of the generated stego-audio. The loss metrics indicate that the improved model has good convergence performance. Furthermore, extensive experiments show that this steganography method has high security and undetectability. Some previous works have utilized GAN for audio steganography; however, those works either modified an existing audio or perform certain changes on a generated one. As far as we know, our method is the first directly generated stego-audio.
(2) As an essential part of a steganography method, an extractor is carefully designed to reconstruct the secret audio from the stego-audio. This consists of five resolution blocks, which are composed with a residual network structure to learn the acoustic features of different frequency bands on the audio spectrum, and this can effectively reduce feature loss. The experimental results show that this extractor can guarantee the complete transmission of a secret audio in both auditory and semantic aspects.
(3) The proposed method does not perform any embedding operation on the existing or generated audio cover, and the distributions of the sample value of the stego-audio and real audio are the same, which can fundamentally resist detection from steganalysis tools.

The rest of this paper is organized as follows. The relevant literature is summarized and analyzed in Section 2. Section 3 describes the details of the proposed method. Our experiments and analyses are presented in Section 4 to verify the performance. Finally, our conclusions are given in Section 5.

2. Related Work

Traditional audio steganography can be implemented in three domains: the time domain, transform domain, and compression domain [20]. In the time domain, the most common is Least Significant Bit (LSB) steganography [21], whose basic idea is to replace the least significant bit in audio sample values with a secret bit. Other typical audio steganography methods in this domain include echo steganography [22], which utilizes the masking effect of the human auditory system; spread spectrum steganography [23], in which, a narrow band information signal is expanded over a wide frequency range; and Quantized Index Modulation steganography (QIM), which treats a secret message as a quantization index [24].

The transform domains used for audio steganography methods include discrete Fourier transform (DFT) [25], discrete cosine transform (DCT) [26] and discrete wavelet transform (DWT) [27]. With the development of audio compression technologies, audio encoded in a compressed way has become popular and, thus, has become a suitable cover option. Furthermore, this kind of steganography can be categorized into three approaches [28]: one is to embed a secret into an audio cover to obtain a stego-audio and then compress it, another is to directly embed a secret into a compressed audio cover, and the last is to decompress the compressed cover and embed a secret and then recompress it to obtain a stego-audio.

GANs [6] were, first, applied to natural language processing [29], computer vision [30], and other fields and are gradually being employed in the information hiding field. Until now, most steganography methods based on GANs took images as covers. Furthermore, coverless image steganography based on GANs is becoming a research hotspot and has also achieved some fruitful research findings. In 2017, Volkhonskiy et al. [31] opened up a

new space for a GAN in information hiding fields and proposed a DCGAN based image steganography model, which consists of three parts: a generator (G), a discriminator (D), and a steganalyzer (S).

Ref. [32] proposed an adaptive steganography method based on a GAN by learning the embedding cost for image steganography, and this outperformed hand-crafted steganography algorithms for the first time in all kinds of steganographic performance. Furthermore, ref. [18] encoded a secret into a cover image using GAN to generate a stego-image. The stego-image is visually indistinguishable from its corresponding cover image. Although a stego-image is generated in this method, the cover is modified. Furthermore, the steganalysis methods can still detect the existence of a secret in the generated stego-image by analyzing the changes to the statistical features.

Ref. [33] first realized coverless information hiding based on a GAN, where the stego-image was directly generated by replacing the class labels of the input data with a secret as a driver, and then the secret was extracted from the stego-image by the discriminator. Herein, as mentioned above, 'coverless' does not mean that there is no media to carry a secret; instead, it means that it does not perform any embedding operation on an existing cover.

Ref. [34] also proposed a coverless image information hiding method, which employed the Wasserstein Generative Adversarial Network (WGAN) and was driven by a secret image to generate a stego-image directly without performing any modification to either the existing cover or a generated cover. In 2021, a cryptographic coverless information hiding method [35] was proposed, which utilized a generative model to transmit a secret image between two different image domains. Aiming at the problem of face privacy leakage in social robots, ref. [36] proposed a visual face privacy protection method. Based on the above knowledge, it can be seen that GANs have been well applied in image covers.

Due to the great achievements of GANs regarding images, they have been gradually applied to audio media, such as audio synthesis [37], speech enhancement [38], and speech emotion [39]. In 2019, a general audio synthesis framework based on DCGAN, named WaveGAN [19], was proposed. WaveGAN is the first attempt to employ a GAN to synthesize raw waveform audio in an unsupervised way. Its main work is to flatten a two-dimensional DCGAN into a one-dimensional model due to the different structures of images and audios.

This contribution provided a new idea for generative audio steganography. In addition, the existing generative audio steganography still needs a cover, which might be an existing cover or a generated cover, and this still relies on the security and robustness of the algorithm as in the traditional methods. Therefore, in order to fundamentally resist the detection of the steganography analysis tools, this paper proposes audio steganography based on WaveGAN that directly generates a stego-audio driven by a secret audio without any modification.

3. The Coverless Audio Steganography Framework

In this section, a coverless audio steganography method is proposed based on WaveGAN, which directly generates a stego-audio when a secret audio is input. In addition, the reconstruction module is designed to reconstruct the secret message from the stego-audio. In order to describe the proposed method more clearly, the relevant symbols in this paper are summarized in Table 2.

Table 2. The meanings of the main symbols used in this paper.

Symbol	The Meaning
R, S	Real dataset, Secret dataset
θ	Network parameters
r, s	Real audio, Secret audio
D, G, E	Discriminator, Generator, Extractor
B	Batch size
M, N	The number of rows and columns in an audio 2-D array
MD	The matrix of the MFCC distance
C	The number of correctly recognized audios
T	Total number of tested audios
X_{spec}, Y_{spec}	The matrix obtained by short-time Fourier transform of the secret audio and the reconstructed secret audio
X, Y	The norm matrixes, representing the norms of each elements in X_{spec} and Y_{spec}, respectively.
D_{fake}	The probability of misattribution that the discriminator regards the input stego-audio from the generator as a real audio
D_{real}	The probability that the discriminator regards the input real audio from the real dataset as a real audio
D_{fake}^{valid}	The probability of misattribution that the discriminator regards the input stego-audio from the generator as a real audio in the validation procedure
D_{real}^{valid}	The probability that the discriminator regards the input real audio from the validation dataset as a real audio
gp	The clipped gradient norm in the training dataset
gp^{valid}	The clipped gradient norm in the validation dataset

3.1. The Proposed Method

A complete steganography algorithm includes the process of hiding a secret and extracting a secret. Similarly, a complete coverless steganography algorithm also includes the generation module of the stego-audio and the extraction module of the secret messages. Therefore, the proposed model is divided into two subsections: the generation module and the reconstruction module. The entire model is illustrated in Figure 1. Furthermore, each module is introduced in detail from two perspectives: the function perspective and its network architecture perspective.

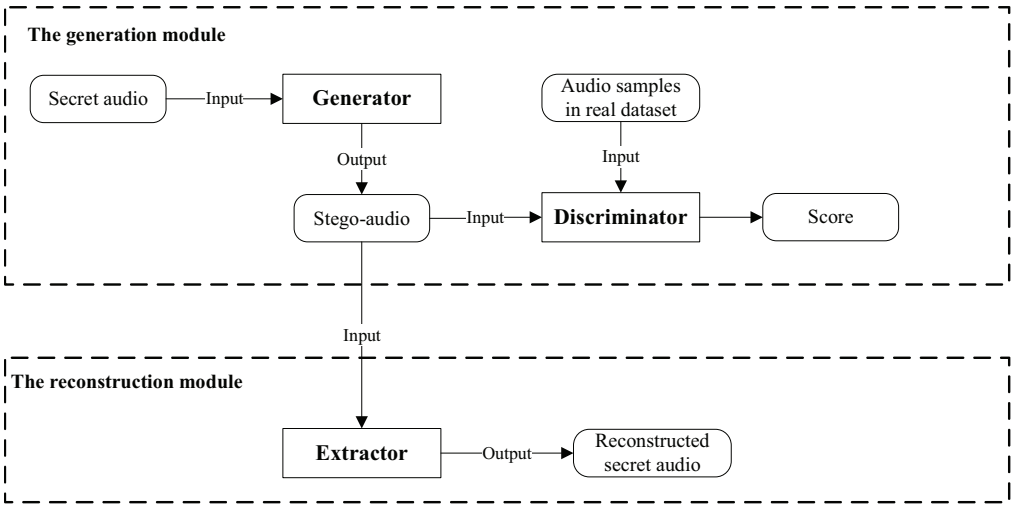

Figure 1. The model of the proposed steganography.

3.1.1. The Generation Module

In this paper, the stego-audios are expected to be generated directly. Therefore, the audio synthesis model WaveGAN is applied to the generation module, which consists of a generator and a discriminator, to generate stego-audios. However, it was improved as follows: First, instead of a noise being fed into the WaveGAN, a secret audio is input into the generator. Second, a post-processing layer is added after the generator. Consequently, the secret audio will be transformed into a stego-audio in our model.

At the same time, for the sake of security, an essential demand is that the feature distribution and the auditory characteristics of the stego-audio should be the same as the audios in the real dataset. In order to meet the above security requirements, the discriminator is used to guide the training of the generator to generate a more authentic stego-audio. When the stego-audio or real audio is input to the discriminator, the discriminant score will be obtained. Specifically, the stego-audio is input to the discriminator to output the probability D_{fake}, and the real audio is input to output the probability D_{real}. These two probabilities will be used to calculate the loss functions to optimize this model.

The design principle of the generation module is essentially audio generation, and the audio synthesis model WaveGAN is employed as the basis of the generation module to realize the generative steganography. The generator is composed of two parts: the core part and the post-processing part. The core part and the discriminator are based on the WaveGAN as shown in Figure 2. Since the audio signal is sequential, the WaveGAN model employs 1D convolution (Conv1d) to extract sequential features. The core part of the generator consists of a linear layer, five groups of transposed convolutions and their activation functions.

After the secret audio is input to the generator, the secret audio will be input to the post-processing part after passing through the core part to obtain the final stego-audio. The structure of the discriminator is in the opposite state to the core part of the generator. The discriminator is composed of five groups of Conv1d, a linear layer and the activation function between them. The discriminator introduces the phase shuffle operation after each activation function. The reason why phase shuffle is introduced is that the transposed convolution in the generator gives the generated stego-audio strong periodic information. Furthermore, the kind of periodic information makes the discriminator judge the authenticity of the stego-audio only through periodicity.

Consequently, the discriminator will not work well, and the generator cannot generate high-quality stego-audio. In order to solve the above problem, a phase-shuffling operation is added between each Conv1d to randomly change the phase of the audio, which can remove the periodic noise effect. Consequently, the discriminator can judge more accurately, and the audio generated by the generator sounds more realistic.

In this work, WaveGAN is used to directly generate stego-audio and realize coverless audio steganography. However, for the security of the steganography algorithm, it is necessary to guarantee the audio quality of the stego-audio first. Therefore, on the basis of using WaveGAN as the generation module, a post-processing layer is replenished to reduce the noise and improve the quality of the generated stego-audio. The post-processing part is supplied after the core part of the network structure in the generation module. Furthermore, it is composed of a Conv1d layer. Subsequent ablation experiment in Section 4.4 show that the post-processing layer effectively improves the audio quality of the generated stego-audio.

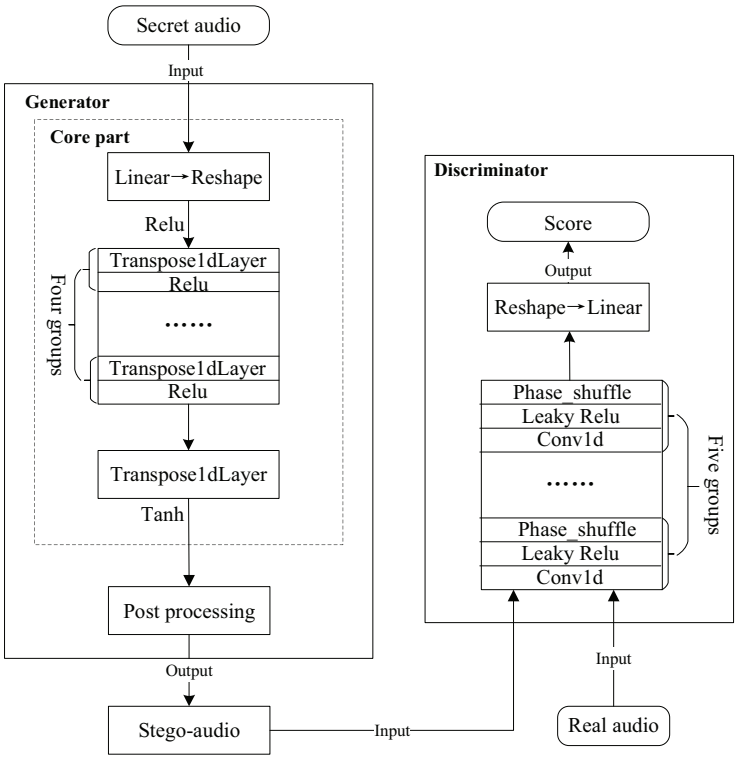

Figure 2. The generator and discriminator structure.

3.1.2. The Reconstruction Module

The reconstruction module is carefully designed as an extractor denoted as E. Its design principle is that the receiver directly inputs the stego-audios into the extractor to reconstruct the secret audios. In the scenario considered in this work, the receiver does not have prior knowledge of the original secret audio. Therefore, in the beginning, the reconstruction module should be trained until the reconstruction module can effectively reconstruct the complete secret audio. The sender and receiver share the trained model. After receiving the stego-audio, the receiver inputs it into the trained reconstruction module to obtain the secret audio. Therefore, unlike the traditional steganography, the proposed method does not need an extra secret key. In other words, for the steganography framework proposed in this paper, the received stego-audio itself is the key.

Its network structure is illustrated in Figure 3. The generated stego-audios are input into the extractor. This network includes five groups of convolution neural networks and a full connection layer. Each group consists of a Conv1d operation and a resolution block. The resolution block is a residual network structure consisting of four Dilated Conv1d layers and four Conv1d layers. The Dilated Conv1d in the resolution block can provide the different receptive fields. In addition, a residual network structure is used in the resolution block to learn the acoustic features of different frequency bands on the audio spectrum, which effectively reduces feature loss during training. Finally, a feature transformation is performed through the fully connected layer to obtain the final reconstructed secret audio.

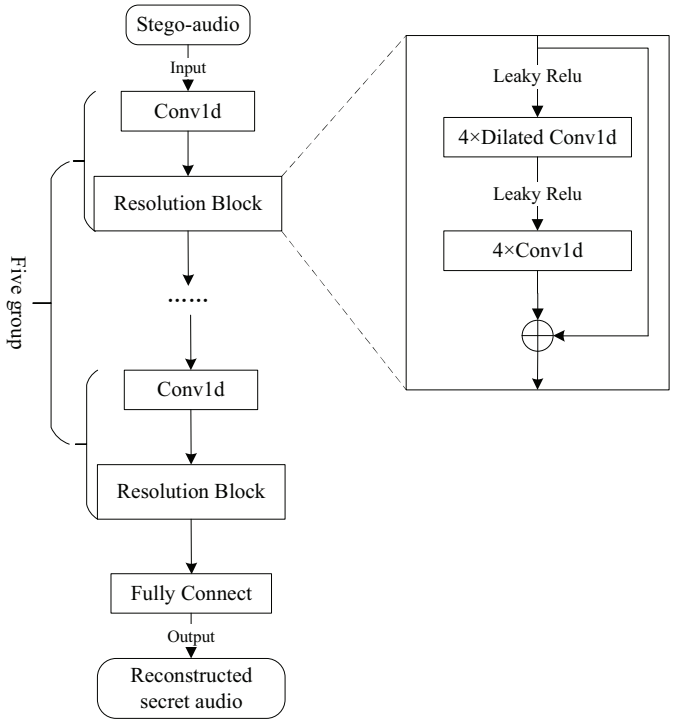

Figure 3. The extractor structure.

3.2. The Loss Function

In this paper, two types of loss functions are defined. One is only involved in backpropagation to guide the training; and the other is used to verify the results without participating in the backpropagation. Among them, four loss functions are defined to guide the model training, including L_D^{train}, D_{wass}^{train}, L_G, and L_{array}.

The first two loss functions (namely, L_D^{train} and D_{wass}^{train}) improve the discriminator to judge accurately; L_G guides the generator to generate more realistic stego-audios; and L_{array} ensures that the reconstructed secret audios by the extractor are less loss and more sound. At the same time, four loss functions are defined to verify the results—namely, L_D^{valid}, D_{wass}^{valid}, L_{mag}, and L_{sc}. L_D^{valid} and D_{wass}^{valid} are used to verify the distinctive ability of the discriminator, and L_{mag} and L_{sc} are employed to verify the reconstruction ability of the extractor. The above loss functions are described as follows in the module that they are located in.

In the generation module, the generator and the discriminator are trained iteratively. The generator is optimized by minimizing D_{fake} to make the generated stego-audio increasingly realistic, and its loss function is shown below:

$$L_G = -\left(D_{fake}\right) \tag{1}$$

For the discriminator, the generated stego-audio and the audio samples from the real dataset are fed at the same time. The same loss indicators as used in the WaveGAN are employed to optimize the discriminator, including D_{wass}^{train} and L_D^{train}, and they are calculated as follows:

$$L_D^{train} = D_{fake} - D_{real} + gp \tag{2}$$

$$D_{wass}^{train} = D_{real} - D_{fake} \tag{3}$$

The former minimizes the distribution difference between the generated stego-audio and the samples from the real dataset by minimizing the Wasserstein distance; the latter

replaces the weight clipping in [40] and adds the gradient penalty [41] to strengthen the constraints to successfully train the model.

At the same time, the discriminator, as a steganalyzer, determines whether the input data is real audio or stego-audio, that is to say, the discriminator is used to determine whether the input data contains the secret or not. After the continuous optimization on the generator and the discriminator, the statistic features of the generated stego-audio are near to the real ones to such an extent that the steganalyzer cannot determine whether the input audio contains a secret.

In order to prevent the overfitting phenomenon, the following loss function is given to verify the distinctive ability of the discriminator

$$L_D^{valid} = D_{fake}^{valid} - D_{real}^{valid} + gp^{valid} \tag{4}$$

$$D_{wass}^{valid} = D_{real}^{valid} - D_{fake}^{valid} \tag{5}$$

During the training, after a batch of secret audios are fed into the generator, stego-audios are synthesized. In the reconstruction module, the stego-audios are fed into the extractor to reconstruct the secret audios. Every original secret audio has one corresponding reconstructed secret audio. Herein, the first extractor loss L_{array} is defined as the function (6), and it is used to optimize the extractor in the gradient back propagation.

$$L_{array} = \frac{1}{B \times M \times N} \sum_{b=1}^{B} \sum_{i=1}^{M} \sum_{j=1}^{N} |s_{ij} - E_{ij}(G(s))| \tag{6}$$

Herein, the secret audio and the reconstructed secret audio are represented in the form of multi-dimension arrays, and M and N represent the number of rows and columns of the array, respectively. M is the frame of the audio, N is the channel of the audio, and B is the batch size. s_{ij} represents the sampling value of the jth channel in the ith frame in the original secret audio; $E_{ij}(G(s))$ represents the element in the reconstructed secret audio extracted by the extractor E.

To further verify the integrity of the reconstructed secret audio, the following two loss indicators are defined, namely, L_{mag} and L_{sc}. The former is to evaluate the differences in amplitude, and the latter is to evaluate the differences in the frequency domain. In addition, they are only verification indicators and do not participate in the gradient back propagation.

$$X = ||X_{spec}|| \tag{7}$$

$$Y = ||Y_{spec}|| \tag{8}$$

$$L_{mag} = \frac{1}{B} \sum_{b=1}^{B} \left| \log_e^{Y_b} - \log_e^{X_b} \right| \tag{9}$$

where X_{spec} and Y_{spec} represent the matrix obtained by short-time Fourier transform of the secret audio and the reconstructed secret audio, respectively. X and Y are the norm matrix, whose values are the norms of the corresponding elements in X_{spec} and Y_{spec}, respectively.

$$L_{sc} = \sum_{b=1}^{B} \frac{\sqrt{\sum_i \sum_j \left(Y_{b,i,j} - X_{b,i,j} \right)^2}}{\sqrt{\sum_i \sum_j \left(Y_{b,i,j} \right)^2}} \tag{10}$$

where $X_{b,i,j}$ is the element in X. Concretely, $X_{b,i,j}$ represents the value of the element whose time is i and frequency is j in the bth audio. Similarly, $Y_{b,i,j}$ represents an element in Y.

The whole training process is shown in the following pseudocode marked as Algorithm 1.

Algorithm 1 The training process of the proposed model.

Initialize the real dataset R and the secret dataset S, Batch size B;
Initialize the generator net G with random θ_g;
Initialize the discriminator net D with random θ_d;
Initialize the extractor net E with random θ_e;
for each training iteration **do**
    Sample a batch of real data (denoted as r) from R;
    Sample a batch of secret data (denoted as s^d) from S;
    Obtain fake audio $G(s^d)$ by inputting s^d to *Generator*;
    Obtain D_{fake} by inputting fake audio $G(s^d)$ to *Discriminator*;
    Update *Discriminator* parameters θ_d by minimizing:
        $\tilde{V} = \frac{1}{B}\sum_{i=1}^{B}\log D(r_i) + \frac{1}{B}\sum_{i=1}^{B}\log(1 - D(fake_i))$,
        $\theta_d \leftarrow \theta_d + \eta\nabla\tilde{V}(\theta_d)$
    Sample another batch of secret data (denoted as s^g) from S;
    Update *Generator* parameters θ_g by minimizing:
        $\tilde{V} = \frac{1}{B}\sum_{i=1}^{B}\log\left(1 - D\left(G\left(s_i^g\right)\right)\right)$,
        $\theta_g \leftarrow \theta_g + \eta\nabla\tilde{V}(\theta_g)$
    Freeze *Generator* parameters θ_g and *Discriminator* parameters θ_d;
    Sample another batch of secret data (denoted as s^c) from S;
    Update *extractor* parameters θ_e by minimizing:
        $L_{array} = \frac{1}{B\times M\times N}\sum_{b=1}^{B}\sum_{i=1}^{M}\sum_{j=1}^{N}|s_{ij}^e - E_{ij}(G(s^e))|$
        $\theta_e \leftarrow \theta_e + \eta\nabla L_{array}(\theta_e)$
end for

4. The Experiments and Analysis

In this section, extensive experiments are presented to verify the effectiveness of our method. The proposed method was implemented using PyTorch1.18.0 and trained on NVIDIA RTX2080 Ti GPUs, with a total of 500 training epochs. The parameter settings of the proposed method is shown in Table 3.

Table 3. Parameter settings.

Generator	Value	Discriminator	Value	Extractor	Value
TransposeConv1d Layer	5	Conv1d Layer	5	Conv1d Layer	5
Upsample Factor	4	PhaseShuffle Layer	4	Resolution Block	5
Stride	1	PhaseShuffle Factor	0.2	Dilated Conv1d Layer	4
Learning Rate	1×10^{-4}	Learning Rate	1×10^{-4}	Learning Rate	1×10^{-4}

Two kinds of datasets are required to train our model: the secret audios and the real dataset. The SC09 dataset was taken as the secret audios. The SC09 dataset is a subset of the Speech Commands Dataset [42], which contains 0∼9 monophonic voice commands of various male and female voices, and the duration of each voice command is one second. The sampling rate of the SC09 dataset was degraded from 16,000 to 8000.

A subset [43] of the Xeno Canto [44] bird sound dataset was modified as the real dataset, which contains 88 kinds of bird songs. In order to save computing resources, the modifications were performed as follows: (1) convert the original flac files into wav files; (2) crop them into two-second bird sound audios and filter out the cropped blank audios; and (3) modify the original sampling rate from 44,100 to 8000.

A total of 800 secret audios and 1600 real audios were randomly chosen for training in the whole experiment. Additionally, the training dataset, test dataset, and validation dataset are split into an 80:10:10 ratio.

4.1. Model Performance

The loss function used in the training process of the model is mentioned above, and the experimental results are shown in Figures 4 and 5. Figure 4 shows the changes in various loss indicators for the generator and the discriminator. Figure 5 displays the changes in various loss indicators for the extractor. It can be seen from the above two figures that, with the increase of the training epochs, the values of each loss indicator tend to be stable. This shows that the model can converge well. On the premise of the convergence performance, analysis was performed on the steganography performance and the reconstructed secret audios.

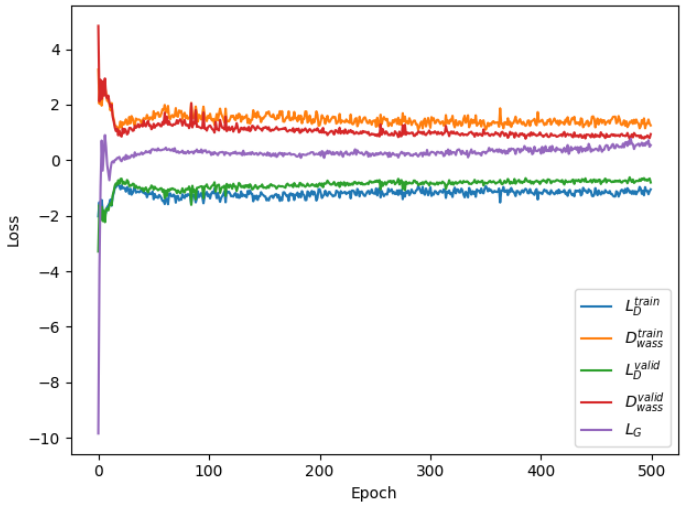

Figure 4. The changes of the loss indicators for the generator and discriminator.

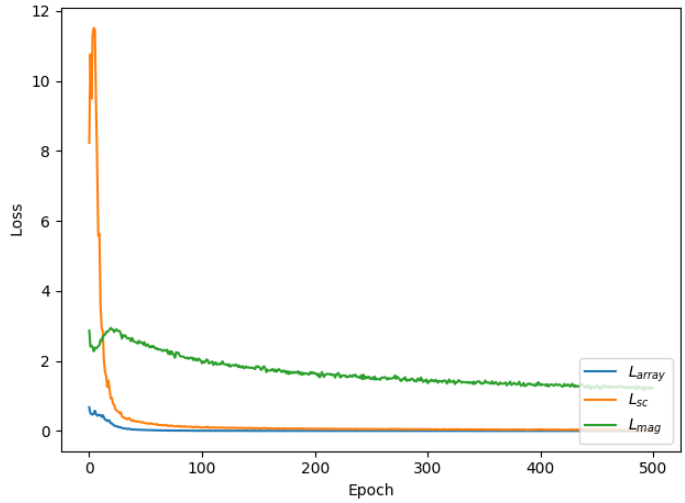

Figure 5. The changes of the loss indicators for the extractor.

4.2. Analysis on the Steganography Performance

4.2.1. The Steganographic Capacity

Steganographic capacity refers to the maximum number of bits that can hide secret messages in a digital cover under the precondition that it cannot be detected. In the proposed coverless steganography method, a two-second stego-audio is generated by the generator driven by a one-second secret audio. Furthermore, different from the previous steganography, the form of the secret in the proposed method is audio. In this section, the steganographic capacity of this method is calculated from the following two perspec-

tives: one is a general method from the size perspective; the other is calculated from the semantic perspective.

From the first perspective, two-second stego-audio is generated directly driven by one-second secret audio, and the ratio of their time lengths between the secret audio and the generated stego-audio can be used to calculate the steganographic capacity. In the proposed method, the read secret audio is fed into the generator in the form of an array, and the stego-audio is also generated in the form of an array.

The ratio between the length of the secret audio array and the length of the generated stego-audio array can also be used to indirectly measure its steganographic capacity. After experimental measurement, the length of the secret audio array to be hidden is 8192, and the length of the generated stego-audio array is 16,384. Consequently, from this perspective, the ratio of the above two methods is 50%. As a result, the proposed method has a high capacity according to this measure indicator.

Although the secret message is transmitted in the form of audio, what is transmitted in essence is the semantic content in the secret audio. From the second perspective, the semantic content in the secret audio should be extracted, and its bit-length is calculated as its steganographic capacity. In this experiment, the duration of each secret audio is one-second, and the semantic content is the speech command of 0∼9. The semantic content that needs to be transmitted is uncertain. Thus, we need to calculate the semantic content that can be generally transmitted in one second. This is expressed in letters in English and similar languages or in alternate characters or digits in other languages.

The following takes secret messages in English as an example. The number of letters can be an indicator of the steganography capacity. It is found that the average speaking speed of British English reaches 198 words per minute [45]. Average word lengths were computed in the range of 6.665∼11.14 [46]. According to ASCII or UTF-8 encoding rules, an English letter is equal to 8 bits. Therefore, it can be concluded that the words that can be speech per minute occupy an average of about 1320∼2206 bits. As a result, it can be calculated that 22∼37 bits of semantic content can be transmitted in a one-second audio.

4.2.2. The Audio Quality

In the proposed method, the stego-audio is generated directly. It must be ensured that the generated stego-audios are acoustically indistinguishable from the natural audios. Therefore, a subjective indicator, MOS (Mean Opinion Score) [47], was chosen to measure the quality of the generated stego-audio. It adopts five levels to evaluate the quality of the tested speech. Thirty listeners were selected to rate the generated stego-audio. Before that, each listener was required to listen to a series of audio samples from the real dataset to ensure that they have the same standard as much as possible.

Ten stego-audios were selected randomly for the experiment. Each listener listened to the ten audios and gave a score. Thus, each stego-audio received 30 scores from 30 different listeners. Finally, the average score of each stego-audio was calculated in turn, and these are listed in Table 4. This shows that the generated stego-audio had high quality and low distortion. The average MOS of the above stego-audios was further calculated as 4.15, indicating that it is not easy for third parties to detect the existence of the secret message.

Table 4. The average MOS of ten randomly selected stego-audios generated by this model with a post-processing layer.

The Stego-Audios	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
The average MOS	4.13	4.09	4.17	4.09	4.21	4.12	4.25	4.14	4.01	4.30

4.2.3. The Authenticity in Statistical Characteristics

In order to prevent third parties from detecting the presence of the secret message, it is necessary to ensure not only the speech quality of the generated stego-audio but also its authenticity compared with the real audio. Therefore, in this section, two different methods

were meticulously designed to verify the authenticity of the generated stego-audio in terms of the statistical characteristics.

Kernel Density Estimation

The GAN makes it impossible to find the real audio in the real dataset corresponding to a given stego-audio. That is to say, there is no one-to-one relationship between a stego-audio and a real audio. Kernel Density Estimation (KDE) is usually used to estimate the unknown density function in the probability theory, which is one nonparametric test method. The distribution of the sample data itself can be seen intuitively through the kernel density estimation diagram. Therefore, KDE is employed to calculate the distribution of the generated stego-audios and real audios.

To calculate their sample value distribution, ten audios were randomly selected from the generated stego-audios and real audios, respectively. The statistical results are shown in Figures 6 and 7. It can be seen in the figures that the generated stego-audios and real audios have very similar distributions. Therefore, the conclusion can be drawn that the generated stego-audios and the real audios have the same statistical distribution.

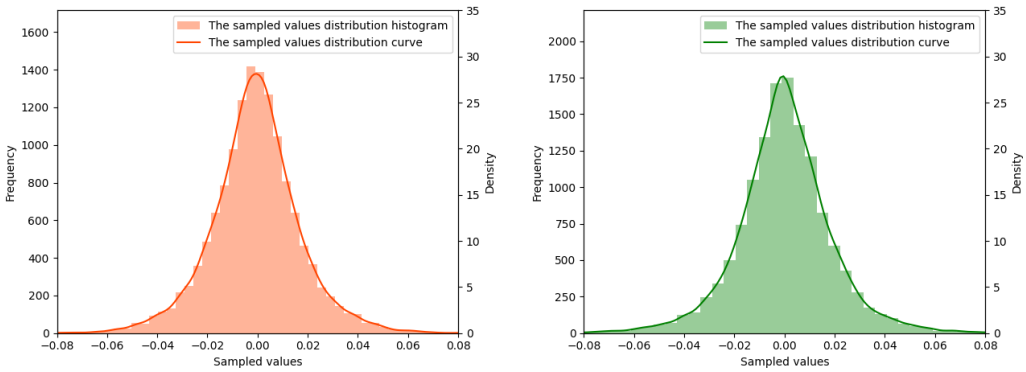

Figure 6. The sample value distribution comparison. On the left is the distribution curve and histogram of the sample values in the real audios, where the x-axis represents random variables, the left y-axis represents the frequency, and the right y-axis represents the density which is calculated as multiplying the frequency by the group distance. The same is true on the right for the stego-audios.

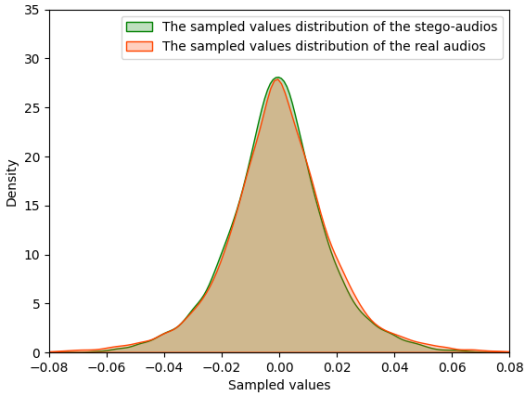

Figure 7. The distributions of the stego-audios and real audio samples.

The Euclidean Distances between the MFCCs

The human auditory system is a special nonlinear system, and its sensitivity to different frequency signals is different from these. The nonlinear representation of the MFCC (Mel Frequency Cepstrum Coefficient) can well reflect the specific features of audio. The Euclidean distance between the MFCCs of the two audios can reflect the difference between them.

Furthermore, the Euclidean distance between the MFCCs of the generated stego-audio and its most similar audio in the real dataset can reflect the authenticity of the generated stego-audio. Therefore, the nearest neighbor in the training dataset for each generated stego-audio was found, and the Euclidean distances between the MFCCs were calculated. In order to verify the effectiveness of the nearest neighbor in the training dataset, the nearest neighbor in the test dataset for each generated stego-audio was found again, and the Euclidean distances between the MFCCs were calculated.

Then, we compared the Euclidean distances between the MFCCs of each generated stego-audio with its nearest neighbors in the training dataset and test dataset to verify the authenticity of the generated stego-audios. The experimental steps are briefly described as follows:

(1) Calculate the Euclidean distances between the MFCCs of each stego-audio and all real audio samples in the training dataset in turn, denoted as MD^{train}, which is a matrix of size $m * n$. m and n denote the number of stego-audios and real audio samples in the training dataset, respectively. Each element $MD_{ij}^{train}(i \in \{1, ..., m\}, j \in \{1, ..., n\})$ of the matrix represents the Euclidean distance between the MFCCs of the ith stego-audio and the jth real audio sample in the training dataset. Find the minimum value of each row in the matrix MD^{train}, denoted as MD_{min}^{train}, which is a matrix of size $m * 1$. This step is illustrated in Figure 8. Consequently, we find the most similar audio samples for each stego-audio from the training dataset. In other words, the nearest neighbors from the training dataset for each stego-audio are found.

(2) Calculate the Euclidean distances between the MFCCs of each stego-audio and all real audio samples in the test dataset, denoted as MD^{test}, which is a matrix of size $m * t$. t is the number of audio samples in the test dataset. Each element $MD_{ij}^{test}(i \in \{1, \ldots, m\}, j \in \{1, \ldots, t\})$ of the matrix represents the Euclidean distance between the MFCCs of the ith stego-audio and the jth real audio sample in the test dataset. Find the minimum value of each row in the matrix MD^{test}, marked as MD_{min}^{test}, which is a matrix of size $m * 1$. This step is illustrated in Figure 9. Therefore, we find the most similar audio samples for each stego-audio from the test dataset. In other words, the nearest neighbors from the test dataset for each stego-audio are found.

(3) Compare the values of the corresponding elements in the MD_{min}^{train} and MD_{min}^{test}.

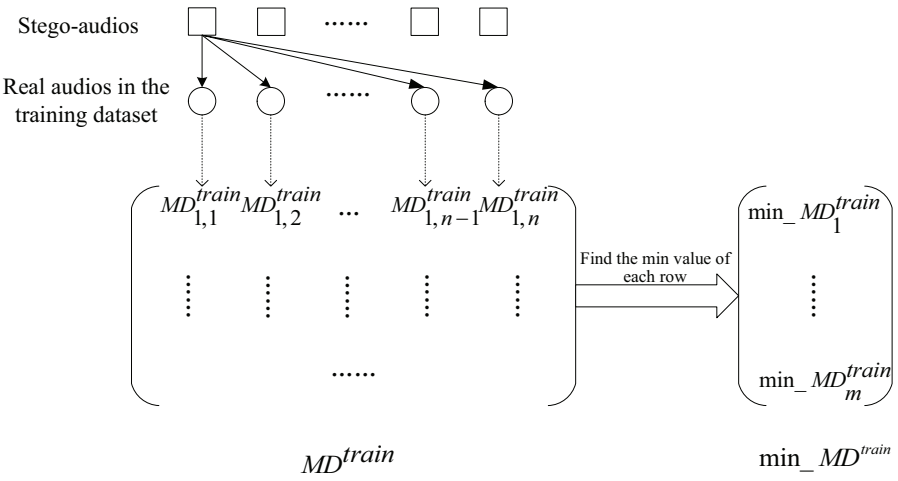

Figure 8. Illustration of Step (1).

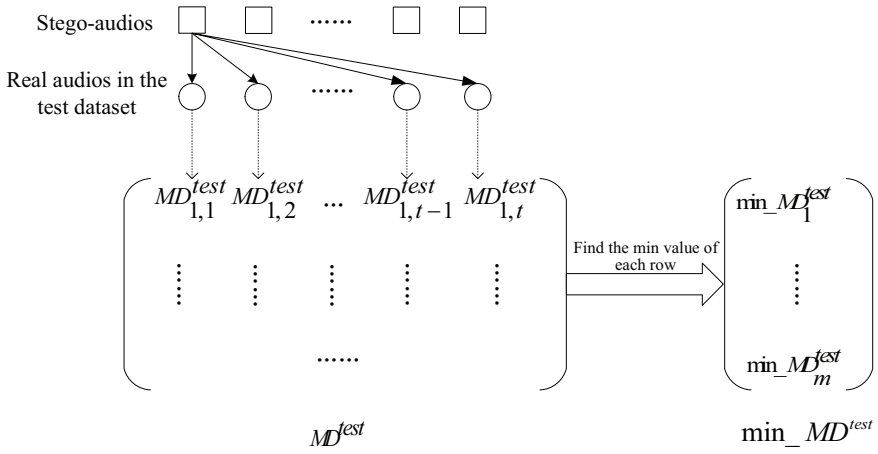

Figure 9. Illustration of Step (2).

As shown in Figure 10, the green and red curves represent the MD_{min}^{train} and MD_{min}^{test}, respectively. Furthermore, the different values between their corresponding elements in MD_{min}^{train} and MD_{min}^{test} are shown in the blue curve of Figure 10. It can be seen from Figure 10 that the Euclidean distances between the MFCCs of each stego-audio and its nearest neighbor in the training dataset are basically the same as the Euclidean distances between the MFCCs of its nearest neighbor in the test dataset. The different values of their Euclidean distances between the MFCCs are very small.

This shows that the generated stego-audios have great similarity with the real audios. Consequently, the generated stego-audio is very real, and it is not easy for a third party to detect the existence of a secret by detecting the particularity of the generated stego-audio. In addition, the similarity of the distances between the MFCCs of the stego-audio and the nearest neighbors in the training dataset and test dataset can also indicate that there is no overfitting in this model.

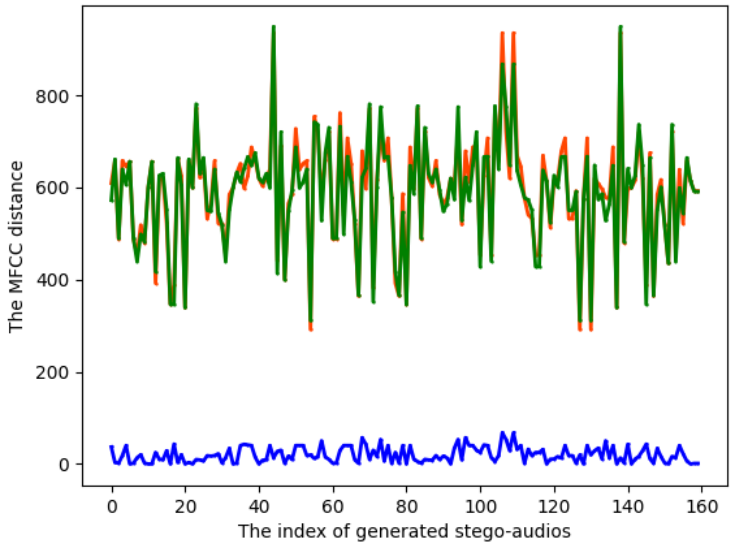

Figure 10. The Euclidean distances between the MFCCs of each stego-audio and the nearest neighbor in the training dataset as well as its nearest neighbor in the test dataset.

4.2.4. Steganalysis

To verify that the proposed method is fundamentally resistant to detection by existing steganalysis methods, we used the trained steganalysis methods Chen-Net [48] and Lin-Net [49] to directly detect the generated stego-audios. Furthermore, we compared

the accuracy with three different audio steganographies: The traditional steganography method [50] refers to a modified audio steganography that embeds a secret in the cover.

The method based on generating a cover [10] refers to audio steganography that embeds a secret in the generated cover. The proposed method refers to audio steganography that directly generates the stego-audio. As shown in Table 5, the accuracy of the method based on generating the cover is significantly lower than the traditional method, and the trained steganalysis methods fail to detect this proposed method. Therefore, our method is fundamentally resistant to detection by steganalysis methods.

Table 5. The accuracy of steganalysis detection with the different steganography algorithms under different embedding rates.

Steganography Methods	Steganalysis Methods	Embedding Rates			
		0.1	**0.2**	**0.3**	**0.5**
Traditional method	Chen-Net	50.00	56.95	63.28	69.77
	Lin-Net	57.34	64.06	67.93	74.13
Based on generating a cover	Chen-Net	48.14	51.29	57.34	61.25
	Lin-Net	49.03	52.33	59.43	64.39
The proposed method	Chen-Net	50.00	50.00	50.00	50.00
	Lin-Net	50.00	50.00	50.00	50.00

4.3. Analysis on the Reconstructed Secret Audio

4.3.1. On the Auditory

In the proposed algorithm, the type of secret messages is audio, and the secret is reconstructed by neural networks, which might make the reconstructed secret audios distorted. This distortion may affect the transmission of secret audios on the audio. Therefore, this distortion can be measured by comparing the difference between the secret audios to be hidden and the reconstructed secret audios. In this section, two methods are used to compare the differences between them. One is to employ the various loss indicators of the extractor. The other is to use various spectrograms to visually compare differences.

In the process of model training, the secret audio to be hidden and the reconstructed secret audio were guaranteed to have a one-to-one correspondence. Therefore, the differences between them are compared by calculating the L_{array}, which can reflect the distortion degree of the reconstructed secret audios in the time domain. In addition, based on the above information, L_{mag} can reflect the differences between the secret audio and the reconstructed secret audio in amplitude, and L_{sc} can reflect their differences in the frequency spectrum. Figure 5 shows that, with the increase of training epochs, these three loss values are all close to 0. This indicates that the difference between them becomes very little. It can be stated that the extractor successfully reconstructs the secret audio with little distortion. Due to the large redundancy of the audio, it is verified that the secret audios can be transmitted completely through audio.

In order to further observe the differences between them, different forms of spectrograms are used for comparison. Figures 11–13 show the comparison results of two groups of secret audio and reconstructed secret audio, respectively. Furthermore, in Figures 11–13, the upper and lower lines are a group of secret audios and the reconstructed secret audios, respectively. It can be seen from Figures 11–13 that there is a small gap between them. Furthermore, this gap is caused by the subtle noise of the reconstructed secret audios. As audios have some redundancy, the secret audios can still be transmitted integrally on audio.

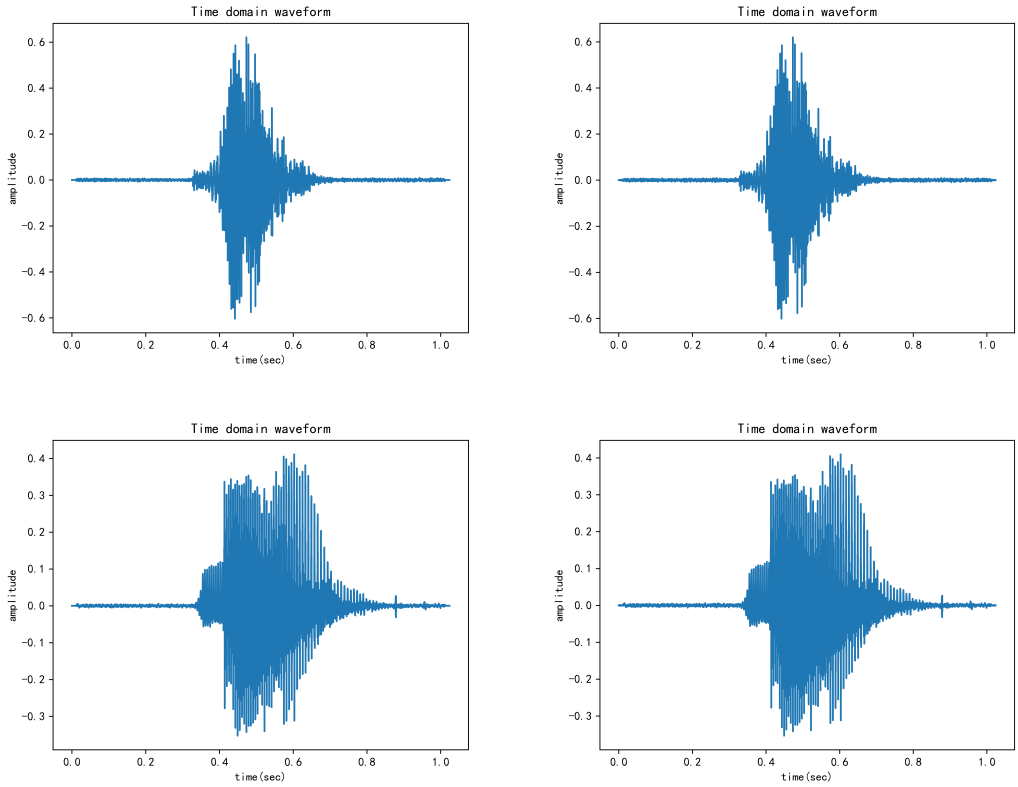

Figure 11. The time-domain waveform. The left column is the time-domain waveform of the secret audios, and the right column is the time-domain waveform of the reconstructed secret audios.

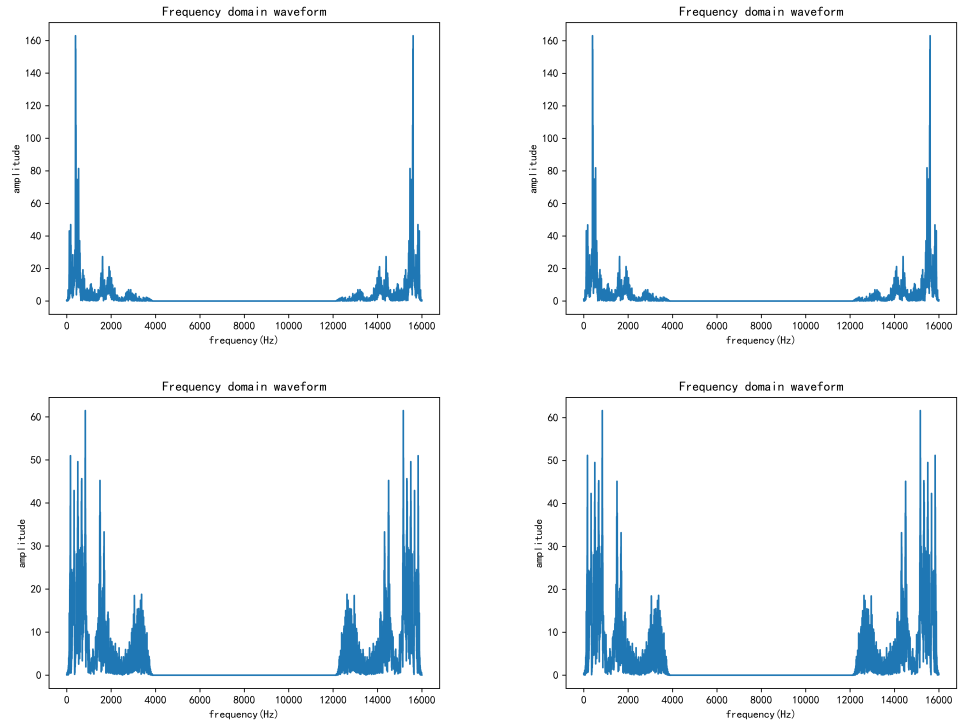

Figure 12. The frequency-domain waveform. The left column is the frequency-domain waveform of the secret audios, and the right column is the frequency-domain waveform of the reconstructed secret audios.

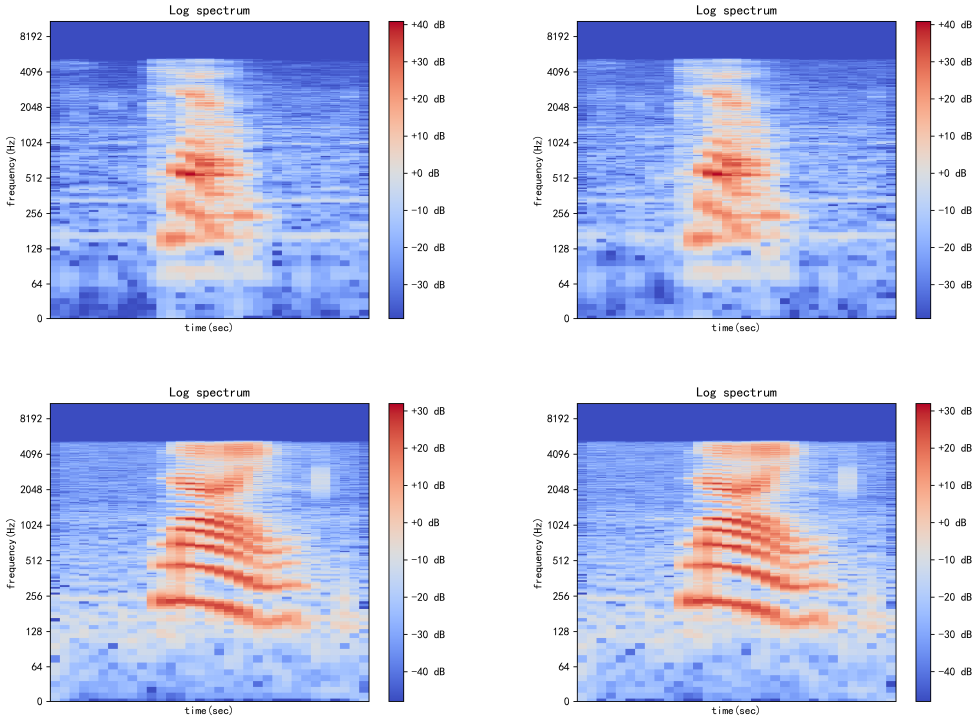

Figure 13. The log spectrogram. The left column is the log spectrogram of the secret audios, and the right column is log spectrogram of the reconstructed secret audios.

4.3.2. On the Semantics

In essence, the ultimate goal of reconstructing the secret audios is to recover their complete secret semantics. As mentioned above, the reconstructed secret audios have some slight distortion, which likely affects the complete semantic transmission of a secret. In this section, we verify that this method guarantees the semantic transmission of the secret audios. These experiments are conducted from the following two perspectives: One is to use the STOI (Short-Time Objective Intelligibility) indicator to test the degree that the reconstructed secret audios can be fully understood. Another is to use speech recognition methods to test the probability that the semantics in the reconstructed secret audios can be correctly recognized.

STOI is one of the important indicators to measure speech intelligibility. A word in a speech signal can only be understood or not. From this perspective, it can be considered that the intelligibility is binary; thus, the value range of STOI is quantified between 0 and 1. STOI represents whether a word can be understood correctly, and a value of 1 indicates that the speech can be fully understood [51].

Since the audios have some redundancy, the reconstructed secret audios are allowed to be distorted; however, it is necessary to ensure that the receivers can understand them. Accordingly, the STOI indicator was chosen to verify the integrity of the reconstructed secret audios. In this experiment, twenty-four reconstructed secret audios were randomly selected to detect their STOI values, and the results are shown in Table 6. The calculations show that the average STOI was 0.9887, indicating that the reconstructed secret audios can be understood well and completely transmitted with semantics.

Table 6. The STOI values of the reconstructed secret audios.

Number	1	2	3	4	5	6	7	8
STOI	0.9993	0.9495	0.9990	0.9885	0.9860	0.9948	0.9972	0.9554
Number	9	10	11	12	13	14	15	16
STOI	0.9642	0.9745	0.9990	0.9981	0.9999	0.9991	0.9994	0.9983
Number	17	18	19	20	21	22	23	24
STOI	0.9998	0.9841	0.9457	0.9996	0.9995	0.9992	0.9999	0.9979

From the second perspective, the speech recognition method was used to detect the specific semantic information of the reconstructed secret audios. In this experiment, sixty-four reconstructed secret audios were randomly selected, and speech recognition was performed on them. Two speech recognition tools were chosen: Speech Recognition-PocketSphinx [52] and Google Cloud Speech API [53]. The speech recognition accuracy rate is defined as the following formula. Table 7 shows the accuracy rate of the secret audio using the above speech recognition tools. Table 7 explains that the semantic information of all the reconstructed secret audios was correctly identified. As a result, the proposed method realized the complete semantic transmission of the secret messages.

$$Accuracy = \frac{C}{T} \times 100\% \tag{11}$$

where C is the number of correctly recognized audios and T is the total number of detected audios.

Table 7. The speech recognition accuracy under different speech recognition APIs.

Speech Recognition Tools	**Accuracy**
Speech Recognition-PocketSphinx	100%
Google Cloud Speech API	100%

4.4. Ablation Experiment

A post-processing layer was added to solve the noise problem in the generated stego-audios in this model. Therefore, it was necessary that ablation experiments were conducted to demonstrate the effectiveness. The speech quality of the generated stego-audio was tested using the same method as in Section 4.2.2 when the post-processing layer was not added. The ten stego-audios generated by this model without the post-processing layer were selected randomly for the experiment. The average MOS values of these ten stego-audios are listed in Table 8.

Compared with Table 4, it can be seen that the average MOS of the generated stego-audios with a post-processing layer had better voice quality. In addition, the average MOS of all the stego-audios without a post-processing layer was calculated as 3.52, and the average MOS of all the stego-audios with a post-processing layer was calculated as 4.15 as shown in Table 9. Apparently, the post-processing layer effectively solved the noise problem and improved the speech quality.

Table 8. The average MOS of ten randomly selected stego-audios generated by this model without a post-processing layer.

The Stego-Audios	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Their average MOS	3.70	3.51	3.65	3.54	3.38	3.49	3.47	3.42	3.43	3.56

Table 9. The MOS ablation experiment for a post-processing layer.

The Post-Processing Layer	Yes	No
MOS	4.15	3.52

4.5. Robustness Experiment

The stego-audio may be jammed during transmission in the public channel; thus, the received audio may contain some extra noises, which might lead to the reconstruction failure of a secret audio. Therefore, it is necessary to evaluate the robustness of the proposed model. The experimental settings were as follows. Thirty-two stego-audios were randomly selected for the robustness testing. Every stego-audio with superposed noises of 5, 10, 15, and 20 db was sent to the extractor, respectively, and 128 audios were reconstructed.

We performed speech recognition on these 128 reconstructed secret audios using the same speech recognition methods as in Section 4.3.2. The speech recognition accuracy rate defined in Equation (11) was also employed here. The experimental results are shown in Table 10. It can be seen in the table that additional noises added to stego-audios did not affect the reconstruction of the secret audios. In other words, even if a stego-audio is jammed to some extent, the receiver can still completely recover the meaning of the secret audio. Thus, the result states that the proposed method has good robustness.

Table 10. The speech recognition accuracy.

Speech Recognition Method	Accuracy				
	0 db	5 db	10 db	15 db	20 db
Speech Recognition-PocketSphinx	100%	100%	100%	100%	100%
Google Cloud Speech API	100%	100%	100%	100%	100%

5. Summary

In this paper, we proposed a new coverless audio steganography method. This is the first work that directly generates stego-audio in audio steganography. The covert communication of the proposed method demonstrated good reliability and security. Experimental and theoretical analysis shows that this proposed method not only has high security and undetectability but also guarantees the complete semantic transmission of secret audio even in the case of distortion.

However, there are still some shortcomings. First, this work considered an idealized scenario in which the sender shares the network model with the receiver, and this premise is a great challenge in reality. In addition, the generation module and the reconstruction module need prior knowledge of the secret audio. If a new secret audio is given, the proposed method needs be further trained on the original basis. Therefore, in future work, we plan to propose a model-free audio steganography framework that can achieve audio steganography from both the time and frequency domains.

Author Contributions: Conceptualization, J.L. and K.W.; methodology, J.L. and K.W.; software, J.L.; validation, J.L. and K.W.; investigation, J.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, K.W.; supervision, K.W. and X.J.; project administration, K.W. and X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Public Sector Support Project of Science and Technology Plan of Shinan District, Qingdao City (Grant No. 2022-2-025-XX).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiao, B.; Huang, Y.; Tang, S. An approach to information hiding in low bit-rate speech stream. In Proceedings of the IEEE GLOBECOM 2008—2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November–4 December 2008; IEEE: New York, NY, USA, 2008; pp. 1–5.
2. Wang, T.; Wang, K. Information hiding method based on short video classification and duration. *J. Qingdao Univ. Nat. Sci. Ed.* **2021**, *34*, 6.
3. Hu, Y.; Huang, Y.; Yang, Z.; Huang, Y. Detection of heterogeneous parallel steganography for low bit-rate VoIP speech streams. *Neurocomputing* **2021**, *419*, 70–79. [CrossRef]
4. Wang, Y.; Guo, L.; Wei, Y.; Wang, C. A steganography method for aac audio based on escape sequences. In Proceedings of the 2010 International Conference on Multimedia Information Networking and Security, Nanjing, China, 4–6 November 2010; IEEE: New York, NY, USA, 2010; pp. 841–845.
5. Wei, Z.; Wang, K. Lightweight AAC Audio Steganalysis Model Based on ResNeXt. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9074771. [CrossRef]
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661.
7. Wu, J.; Chen, B.; Luo, W.; Fang, Y. Audio steganography based on iterative adversarial attacks against convolutional neural networks. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2282–2294. [CrossRef]
8. Ye, D.; Jiang, S.; Huang, J. Heard more than heard: An audio steganography method based on gan. *arXiv* **2019**, arXiv:1907.04986.
9. Yang, J.; Zheng, H.; Kang, X.; Shi, Y.Q. Approaching optimal embedding in audio steganography with GAN. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: New York, NY, USA, 2020; pp. 2827–2831.
10. Chen, L.; Wang, R.; Yan, D.; Wang, J. Learning to generate steganographic cover for audio steganography using gan. *IEEE Access* **2021**, *9*, 88098–88107. [CrossRef]
11. Yue, F.; Zhu, H.; Su, Z.; Zhang, G. An Adaptive Audio Steganography Using BN Optimizing SNGAN. *Chin. J. Comput.* **2022**, *45*, 427–440.
12. Wang, J.; Wang, R.; Dong, L.; Yan, D. Robust, Imperceptible and End-to-End Audio Steganography Based on CNN. In Proceedings of the Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, 30 October–1 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 427–442.
13. Ren, Y.; Liu, D.; Xiong, Q.; Fu, J.; Wang, L. Spec-resnet: A general audio steganalysis scheme based on deep residual network of spectrogram. *arXiv* **2019**, arXiv:1901.06838.
14. Zhu, J.; Wang, R.; Yan, D. The sign bits of huffman codeword-based steganography for aac audio. In Proceedings of the 2010 International Conference on Multimedia Technology, Ningbo, China, 29–31 October 2010; IEEE: New York, NY, USA, 2010; pp. 1–4.
15. Wang, Y.; Yang, K.; Yi, X.; Zhao, X.; Xu, Z. CNN-based steganalysis of MP3 steganography in the entropy code domain. In Proceedings of the Proceedings of the sixth ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 55–65.
16. Wang, Y.J.; Guo, L.; Wang, C.P. Steganography method for advanced audio coding. *J. Chin. Comput. Syst.* **2011**, *32*, 1465–1468.
17. Ren, Y.; Liu, D.; Liu, C.; Xiong, Q.; Fu, J.; Wang, L. A Universal Audio Steganalysis Scheme based on Multiscale Spectrograms and DeepResNet. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 665–679. [CrossRef]
18. Qin, J.; Wang, J.; Tan, Y.; Huang, H.; Xiang, X.; He, Z. Coverless Image Steganography Based on Generative Adversarial Network. *Mathematics* **2020**, *8*, 1394. [CrossRef]
19. Donahue, C.; McAuley, J.; Puckette, M. Adversarial audio synthesis. *arXiv* **2018**, arXiv:1802.04208.
20. Wang, Y. Research on the Mechanism and Key Technology of Audio Steganalysis. Ph.D. Thesis, University of Science and Technology of China, Hefei, Anhui, 2011.
21. Balgurgi, P.P.; Jagtap, S.K. Audio steganography used for secure data transmission. In *Proceedings of the International Conference on Advances in Computing*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 699–706.
22. Erfani, Y.; Siahpoush, S. Robust audio watermarking using improved TS echo hiding. *Digit. Signal Process.* **2009**, *19*, 809–814. [CrossRef]
23. Dutta, H.; Das, R.K.; Nandi, S.; Prasanna, S.M. An overview of digital audio steganography. *IETE Tech. Rev.* **2020**, *37*, 632–650. [CrossRef]
24. Sun, X.; Wang, K.; Li, S. Audio steganography with less modification to the optimal matching CNV-QIM path with the minimal hamming distance expected value to a secret. *Multimed. Syst.* **2021**, *27*, 341–352. [CrossRef]
25. Gang, L.; Akansu, A.N.; Ramkumar, M. MP3 resistant oblivious steganography. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, USA, 7–11 May 2001; Proceedings (Cat. No. 01CH37221); IEEE: New York, NY, USA, 2001; Volume 3, pp. 1365–1368.
26. Ma, Y.; Han, J. Audio watermarking in the DCT domain: Embedding strategies and algorithms. *Acta Electron. Sin.* **2006**, *34*, 1260–1264.
27. Sheikhan, M.; Asadollahi, K.; Shahnazi, R. Improvement of embedding capacity and quality of DWT-based audio steganography systems. *World Appl. Sci. J.* **2011**, *13*, 507–516.

28. Ru, X. Research on Audio Steganography and Analysis Technology. PhD Thesis, Zhejiang University, Hangzhou, China, 2006.
29. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA 4–9 February 2017; Volume 31.
30. Chen, J.; Zhang, Z.; Xie, X.; Li, Y.; Xu, T.; Ma, K.; Zheng, Y. Beyond Mutual Information: Generative Adversarial Network for Domain Adaptation Using Information Bottleneck Constraint. *IEEE Trans. Med. Imaging* **2021**, *41*, 595–607. [CrossRef]
31. Volkhonskiy, D.; Nazarov, I.; Burnaev, E. Steganographic generative adversarial networks. In Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019), Amsterdam, Netherlands, 16–18 November 2019; SPIE: Washington, DC, USA, 2020; Volume 11433, pp. 991–1005.
32. Yang, J.; Ruan, D.; Huang, J.; Kang, X.; Shi, Y.Q. An embedding cost learning framework using GAN. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 839–851. [CrossRef]
33. Liu, M.m.; Zhang, M.q.; Liu, J.; Zhang, Y.n.; Ke, Y. Coverless information hiding based on generative adversarial networks. *arXiv* **2017**, arXiv:1712.06951.
34. Duan, X.; Song, H. Coverless information hiding based on generative model. *arXiv* **2018**, arXiv:1802.03528.
35. Li, Q.; Wang, X.; Wang, X.; Ma, B.; Wang, C.; Shi, Y. An encrypted coverless information hiding method based on generative models. *Inf. Sci.* **2021**, *553*, 19–30. [CrossRef]
36. Lin, J.; Li, Y.; Yang, G. FPGAN: Face de-identification method with generative adversarial networks for social robots. *Neural Netw.* **2021**, *133*, 132–147. [CrossRef]
37. Kim, J.H.; Lee, S.H.; Lee, J.H.; Lee, S.W. Fre-GAN: Adversarial frequency-consistent audio synthesis. *arXiv* **2021**, arXiv:2106.02297.
38. Li, Y.; Sun, M.; Zhang, X. Perception-guided generative adversarial network for end-to-end speech enhancement. *Appl. Soft Comput.* **2022**, *128*, 109446. [CrossRef]
39. Sahu, S.; Gupta, R.; Espy-Wilson, C. Modeling Feature Representations for Affective Speech Using Generative Adversarial Networks. *IEEE Trans. Affect. Comput.* **2022**, *13*, 1098–1110. [CrossRef]
40. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875v3.
41. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. *arXiv* **2017**, arXiv:1704.00028.
42. Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv* **2018**, arXiv:1804.03209.
43. Tatman, R. British Birdsong Dataset. Available online: https://www.kaggle.com/rtatman (accessed on 25 November 2022).
44. Xeno, C. Speech Commands Zero Through Nine (SC09) Dataset. Available online: https://xeno-canto.org/ (accessed on 23 November 2022).
45. Li, W. British English-Speaking Speed 2020. *Acad. J. Humanit. Soc. Sci.* **2021**, *4*, 93–100. [CrossRef]
46. Rajput, N.K.; Ahuja, B.; Riyal, M.K. Alphabet usage pattern, word lengths, and sparsity in seven Indo-European languages. *Digit. Scholarsh. Humanit.* **2020**, *35*, 727–736. [CrossRef]
47. Viswanathan, M.; Viswanathan, M. Measuring speech quality for text-to-speech systems: Development and assessment of a modified mean opinion score (MOS) scale. *Comput. Speech Lang.* **2005**, *19*, 55–83. [CrossRef]
48. Chen, B.; Luo, W.; Li, H. Audio steganalysis with convolutional neural network. In Proceedings of the fifth ACM Workshop on Information Hiding and Multimedia Security, Philadelphia, PA, USA, 20–22 June 2017; pp. 85–90.
49. Lin, Y.; Wang, R.; Yan, D.; Dong, L.; Zhang, X. Audio steganalysis with improved convolutional neural network. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, Paris, France, 3–5 July 2019; pp. 210–215.
50. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
51. Taal, C.H.; Hendriks, R.C.; Heusdens, R.; Jensen, J. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 2125–2136. [CrossRef]
52. CMU. Speech Recognition-PocketSphinx. Available online: https://github.com/cmusphinx/pocketsphinx (accessed on 13 July 2022).
53. Google. Google Cloud Speech API. Available online: https://cloud.google.com/speech-to-text/docs/ (accessed on 13 July 2022).