

# A comparative analysis of audio steganography methods and tools

PIETER MATTHIJS REYERS, University of Twente, The Netherlands

Steganography is the process of hiding data within other data. Unlike cryptography which aims to secure data through obscuring its meaning, steganography aims to secure data by obscuring its existence altogether. Although many different types of cover data, such as text or images, can be used to embed secret data in, this paper focuses on audio steganography because it is uniquely positioned to enable real time secure communication. A problem within currently available audio steganography research is that there are no thorough comparative analyses available of modern audio steganography methods, such as machine learning based methods. While such comparative analyses are available for many other types of steganography, such as image steganography. The origin of this problem can be partially attributed to the fact that there is a wide variety of evaluation features and audio datasets in use among audio steganography researchers. This makes it difficult to directly compare the results of two different papers. To address this problem, this paper provides a comparative analysis of two methods proposed in the last 5 years (An Audio Steganography Method Based on generative adversarial networks (GAN), and Logistic Tan Map Based Audio Steganography) and two popular steganography tools (Hide4PGP, Steghide) that have existed for over a decade. This analysis was performed using three evaluation features (Bit Error Rate (BER), Signal to Noise Ratio (SNR), Embedding Percentage (EP)) and three datasets (TIMIT, GZTAN, ESC50). These were selected to be most suitable for the evaluation of steganography research based on criteria formed after thorough review of the ones most commonly found in audio steganography research. This comparative analysis and the datasets/evaluation features recommendation is important because it will help future searchers in more consistently evaluating their audio steganography methods and understand those evaluation results in the context of the performance of other methods. It also helps shed light on the performance of new methods in comparison to the old existing tools.

Additional Key Words and Phrases: Audio steganography, Data security, Machine learning, Generative adversarial network, Logistic Tan Map

## 1 INTRODUCTION

Steganography is the process of hiding data within other data [33]. Unlike the related field of cryptography, which aims to secure data by obscuring its meaning from any unauthorized parties [4], steganography seeks to secure data by obscuring its existence all together. The data which the secret data is embedded within is called the cover data (or cover audio) while the resulting data with the secrets embedded in it is called the stego data (or stego audio) [3, 4]. Although many different types of cover media can be used, such as text [26] or images [34], this research focuses on audio steganography because it is uniquely positioned for enabling real-time communication [4, 33]. This is in part because of the prevalence of reliable audio channels in VoIP (Voice over Internet Protocol) services such as Discord or Microsoft Teams [19], but also

because many audio steganography methods are computationally inexpensive enough that they could be run in real time [2, 33].

Data security is only becoming more relevant in our increasingly digital world and audio steganography could play an important role in facilitating secure real time communication [36], but despite this potential it is unclear how audio steganography methods proposed in the last few years compare against each other as there are no extensive comparative analyses of recent audio steganography methods available [4]. This problem is rooted primarily in the fact that there is a wide variety of datasets and evaluation features in use [4], which means two papers often cannot be directly compared since different evaluation features have different units or numerical ranges and methods may perform differently on different types of audio [10] so a papers results may depend strongly on the dataset used.

To address this problem, this paper provides a comparative analysis of two methods proposed in the last 5 years ([44] and [12]) and two popular steganography tools that have existed for over a decade (Hide4PGP and Steghide). The results of which can be found in section 7. The analysis was performed using three evaluation features (Bit Error Rate (BER), Signal to Noise Ratio (SNR), Embedding Percentage (EP)) and three datasets (TIMIT[13], GZTAN[38], ESC50[30]). These were selected as most suitable using criteria based on a review of recent audio steganography papers as described in sections 3.4 and 3.5.

Because the terms *tools* and *method* are used frequently in this paper it is important to define exactly what they mean. When speaking about *steganography tools*, this paper refers to computer programs that implement a steganography method in a user friendly way and that are usually distributed as a compiled binary. So programs such as Steghide or MP3Stego are considered a tool, but a Python notebook included with a paper is not. When speaking about *new methods*, this paper refers to steganography methods/algorithms introduced in a paper during the last 5 years (2018-2023). To keep things concise, when referring to both the tools and the new steganography methods, this research paper simply refers to the *techniques* (plural). Colloquially a steganography technique or method (singular) are both understood to mean an algorithm or system for performing steganography, but to avoid confusion with the *techniques* this paper always uses *method* when speaking about such an algorithms.

The comparative analysis and the datasets/evaluation features recommendation provided in this paper are important because it will help future searchers to more consistently evaluate their newly developed audio steganography methods, as well as help with understanding those evaluation results by placing them in the context of the performance of other methods. This paper also helps shed light on the performance of new methods in comparison to the existing tools by providing a novel comparative analysis.

---

TScIT 39, July 7, 2023, Enschede, The Netherlands

© 2023 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## 1.1 Research questions

The main research question of this paper is as follows: "How do audio steganography methods proposed in the last 5 years and popular audio steganography tools compare against each other in the 3 key metrics: capacity, robustness, and perceptual transparency?" In supporting this research question, the following sub research questions were identified and answered first:

- (1) Which existing dataset(s) is most suited for the evaluation of steganography techniques?
- (2) Which evaluation features are most appropriate for measuring each key metric?
- (3) What are the limitations and challenges of using these techniques in practice?

## 2 AUDIO STEGANOGRAPHY BACKGROUND

This section provides a brief theoretical background to help understand the rest of the contents of this paper. It touches on how different types of audio steganography methods are categorized and evaluated. As mentioned in section 1.1, there are three key metrics [4, 5] by which the performance of an audio steganography method is generally evaluated:

- *Perceptual transparency*: The ability of a given steganography method to evade detection[4], often measured as the amount of noise added by the method. Although some papers [6, 25] also include a subjective score of how perceivable the noise is because the human auditory system (HAS) does not perceive all types of noise equally.
- *Robustness*: The ability of the embedded data to withstand signal degradation (e.g. compression, resampling, filtering) of the stego file or deliberate attacks such as LSB dropping [5].
- *Hiding capacity*: The density with which secret data can be embedded in the cover signal [24].

These key metrics naturally oppose each other as by their nature increasing the performance in one will usually hurt the performance in another [4] (e.g. increasing hiding capacity will cause more signal distortion and thus reduce perceptual transparency). There exist many different evaluation features that attempt to quantify the performance in a given key metric, such as BER[41] (Bit error rate) for robustness or PSNR[25] (Peak Signal to Noise Ratio) for transparency. A detailed explanation of the features used in this report can be found in section 6, while a detailed overview of commonly used evaluation features is included in the Appendix in Tables 4, 5, and 6. Unfortunately the evaluation features used in audio steganography research can differ greatly per paper [4, 16, 42, 46] and not all papers include a feature for each of the key metrics [12, 46].

There are many different methods for embedding secret data inside audio cover files. Some of these methods, like LSB coding [28], borrow ideas from other steganography fields like image steganography [34], while other methods, like those targeting the MP3 codec [17], are unique to the audio medium. Previous reviews [3, 10, 27] have identified three main domains into which all these audio steganography methods can be categorised, namely: *the spacial domain*, *the transform domain*, and *the coded domain*. One review [27] identified a fourth domain, *the compressed domain*, which covers methods that operate directly on compressed cover data, but

this domain has been purposefully left out here as no papers whose method belonged to the compressed domain were cited.

Although each domain encompasses many different methods (some examples of which are given below) this paper only discusses the LSB coding and DWT methods as these are immediately relevant to understanding the techniques included in the comparative analysis.

### 2.1 Spatial domain - LSB coding

The spatial domain, sometimes also referred to as the temporal domain [10] or the substitution techniques [27], encompasses relatively simple methods, like Echo hiding [21] and Parity coding[37], in which the secret data is embedded directly in the audio bit stream.

Most relevant for this paper, it also encompasses LSB coding (or low-bit encoding [10]) methods. LSB methods are some of the simplest [15] and most common [33] steganography methods which work by replacing the least significant bit of samples in the cover audio with a bit from the secret data.

In its simplest form the bits of the secret data are simply distributed evenly amongst all the LSB's of the cover audio samples [10, 15], though this creates a very predictable type of distortion that is easily detected by steganalysis methods. Many improvements have been proposed to address this by adaptively selecting which samples to modify, such as only using samples with an absolute value higher than some threshold [28]. There also exist variants that seek to improve hiding capacity by replacing not just the lowest bit but the lowest two or three bits, sometimes called LSB-2 and LSB-3 respectively [25].

### 2.2 Transform domain - DWT

The transform domain, which includes methods like spread spectrum[9] and phase coding [11], encompasses methods in which the audio data is actually interpreted as audio and then transformed in some way to hide the secret data inside the audio signal [35]. Usually with the goal of only altering the audio signal in ways the HAS is insensitive to [10].

The transform domain also includes DWT (Discrete wavelet transform) methods, which are methods involving embedding the secret data within the frequency sub bands of the cover audio signal's wavelet coefficients [24, 27, 46].

### 2.3 Coded domain

Unlike the previous methods which embed in the audio signal itself (pre-encoder embedding), methods in the coded domain seek to use the properties of a specific audio codec (like MP3 [17]) to embed the secret data [18]. No coded domain methods are included in this comparative analysis.

## 3 RESEARCH METHODOLOGY

The research process for this paper was divided into separate stages, which are listed and explained in chronological order below.

### 3.1 Stage 1: Literature review

In the first stage existing literature in the form of both conference papers and journal papers were searched and categorized, both

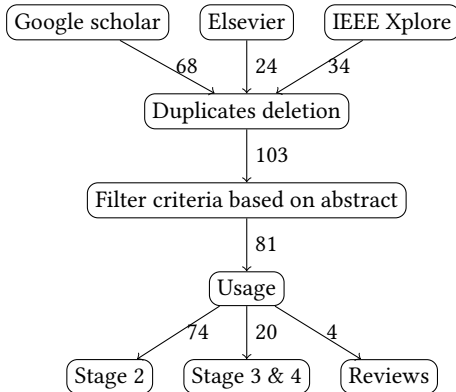
to provide the theoretical background found in section 2 and to determine the available state of the art methods. Specifically, the databases of Google scholar, Elsevier, and IEEE Xplore were searched with the terms "audio steganography", "audio steganography review", "audio steganography method", "audio steganography machine learning". (Other databases were considered but not searched due to time constraints).

The criteria listed below was taken into account during the literature search. The last 7 years (after 2016) was chosen as a threshold rather than the last 5 years (after 2018) so more papers could be taken into account for determining common datasets and evaluation features. The stage in which the new methods for the comparison were chosen applies further criteria which does use 2018 as threshold (see section 3.2).

- The paper must be published after 2016
- The paper is written in English.
- The paper is about **audio** steganography.
- For papers proposing a new method; it must actually be a new method, not a novel combination of existing cryptography and steganography methods.

Aside from those papers which were deemed not to fit the criteria based on the title alone, the searched initially yielded 68 papers from Google scholar, 24 from Elsevier, and 34 from IEEE Xplore. These papers were then loaded into a research assistant (Zotero) to delete duplicates and filter according to the criteria based on the papers' abstracts. Leaving 81 papers in total, as can be seen in Figure 1.

Fig. 1. High level overview of the literature review process.



A subset of papers proposing a steganography method was then further analyzed and documented in a spreadsheet<sup>1</sup> to record their used datasets and evaluation features, which was then used in stage 3 and 4. The complete list of 74 papers proposing a new steganography method was further filtered in stage 2 according to the criteria listed in section 3.2. While the 4 identified reviews [4, 11, 27, 36] were used in writing the audio steganography background found in section 2.

<sup>1</sup><https://docs.google.com/spreadsheets/d/e/2PACX-1vRZCvdOHhNPun-Zil2f77CBKt169S8ZKmkcIDNkWPssh4TxP1PgumFvNaUysHM9bf-FSnUy5EO6v6-c/pubhtml>

### 3.2 Stage 2: Techniques selection

This stage served to select which audio steganography tools and new methods would be included in the comparison. The results of which can be found in section 4. To represent the state of the art of the audio steganography field, two recent methods were selected based on the following inclusion criteria:

- The paper must be published after 2016.
- The method proposed in the paper must be compatible with or easily adaptable to the evaluation dataset chosen in stage 3.
- The source code or an executable version of the method must be available or made available by authors.

To represent the established steganography methods in the comparison, existing audio steganography tools were reviewed and two were selected. The tools were discovered through their appearances in research papers discovered in stage 1 and through a simple Google search for "audio steganography tools". The chosen tools can be found in section 4.2.

### 3.3 Stage 3: Data set selection

This stage addressed research question 1, finding the dataset(s) most suited for the evaluation of steganography methods. To achieve this the criteria for a suitable evaluation dataset were first established based on the information from the papers and reviews identified in stage 3.1. These criteria can be found in section 5. Then a list of potential datasets was formed based on datasets used by existing papers in the spreadsheet<sup>1</sup> and extended with additional datasets (Speaker Recognition Dataset, ESC-50[30]) discovered on Kaggle<sup>2</sup> and Google Research<sup>3</sup>. After which all datasets were evaluated with the previously established criteria (the result of which can be found in the Appendix in Table 3) and a combination of suitable datasets was selected (see section 5).

### 3.4 Stage 4: Evaluation features selection

This stage addressed research question 2, finding the evaluation feature most suited for measuring the performance for each one of the three key metrics (perceptual transparency, robustness, hiding capacity). To determine which features were the most suitable, all of the features found in existing literature and documented in the spreadsheet<sup>1</sup> were reviewed (see Appendix for Tables 4, 5, and 6) and the most suitable feature for each metric was determined. The results of which can be seen in section 6.

### 3.5 Stage 5: Analysis

To answer the last research question all the selected techniques were applied to the selected datasets and the evaluation features were computed in the manner described in the subsections below. The resulting data is visualized and discussed in section 7 while the derived conclusions and answer to the last research question can be found in section 8.

**3.5.1 Audio files selection methodology.** Because each audio file used for evaluation had to be embedded and extracted multiple

<sup>2</sup><https://www.kaggle.com/>

<sup>3</sup><https://research.google.com/>

times in a process that took about 2 minutes per file, it was simply not feasible to use all 7300 audio files in the datasets. Instead a random subset of audio files was chosen for each music genre in GTZAN (55 files in total; 5 per genre) and a random subset of audio files for some speakers in TIMIT (16 files in total) were selected to keep the data analysis computation times within a few hours while still providing statistically significant results.

In order for the data selection process to be reproducible for any future researchers the random number generator was seeded with a constant value. The source code for this process can be found in the `get_data.py` file in the code repository <sup>1</sup>.

**3.5.2 Hiding capacity methodology.** To determine the hiding capacity for the tools, a script (`compute_hiding_capacity.py`, see code repository <sup>1</sup>) was written to apply the built-in commands `steghide --info cover.wav` and `hide4pgp -i cover.wav` to all audio files selected in section 3.5.1. The resulting hiding capacity for each file/tool was then saved in a CSV file (`results/hiding_capacity.csv`).

However, due to an issue with StegHide certain files from the TIMIT [13] dataset, such as `SI1629.wav`, could not actually have the reported amount of bytes embedded in them. To fix this a "back off" system was implemented where the secret data would be reduced in size by one byte for that file each time StegHide returned an error.

To determine the hiding capacity for the new methods (GAN and TAN; see section 4), an automated script (`compute_gan_tan.py`) was produced to systematically increase the secret message size until the message could no longer be successfully extracted, which was then considered the maximum hiding capacity. To keep the comparison fair any message compression features were disabled for those methods and tools that supported them.

**3.5.3 Perceptual transparency methodology.** The amount of secret data that is embedded influences how much distortion is introduced to the stego audio. Because of this several stego files were generated for each technique and cover audio file combination with secret message sizes ranging from 1B to 100kB. The SNR of all these stego files was then computed (`compute_transparency.py`) as described in section 6.2. And this data was then used so that the relationship between secret message size and SNR for each of the techniques could be graphed, as can be seen in Figure 3 in section 7.2

For the perceptual transparency evaluation, the SNR of the stego file relative to the cover file was computed as described in section 6.

Because the amount of secret data that is embedded influences how much distortion is introduced to the stego audio, the SNR was computed at various secret message sizes to determine the relationship between the secret message size and SNR, as can be seen in Figure 3.

**3.5.4 Robustness methodology.** To measure the robustness of the techniques, stego files with the maximum size secret message were generated for each technique. Each of these files was then mixed with various levels of background noise from the ESC-50[30] dataset, ranging from 0dB (full volume) to -65536dB (imperceptible to HAS). After which the secret message was extracted from the noisy stego

audio and compared against the real secret message to compute the BER (Bit error rate) as described in section 6.

## 4 TECHNIQUES SELECTION

This section describes how the techniques (i.e. tools and new methods) to be included in the comparative analysis were selected.

### 4.1 Audio steganography methods

Recall that one of criteria for the new methods established in section 3.2 was the availability of source code or an executable version of the method. This is needed to be able to apply the method to the evaluation dataset chosen in section 5, but unfortunately this criteria proved especially difficult. Although many papers claim that their source code is available on request, these requests were often left unanswered, and for those papers which had their source code openly available this often did not include trained model weights or code for training the model.

The following methods were ultimately chosen over the other candidates [7, 8, 23, 40, 46] because an implementation could be found or the algorithm was trivial to implement:

#### GAN **Heard More Than Heard: An Audio Steganography Method**

**Based on GAN** [44] This method operates in the transform domain in a DWT like fashion. The cover audio is converted into a spectrogram which is then fed into an adversarial GAN network to embed a secret message, after which the resulting spectrogram is turned back into stego audio.

**TAN Logistic Tan Map Based Audio Steganography** [12] This method is an improvement on regular LSB, where the secret bits are pseudo randomly distributed along the audio signal using a logistic tan map with the goal of providing better security.

### 4.2 Audio steganography tools

The following two tools were chosen to be included in the comparative analysis because of their prevalence in audio steganography research and because each tool represents a different audio steganography domain:

- **Hide4PGP 2.0** [1]: This freeware developed by Heinz Repp operates in the temporal domain using a variation of LSB.
- **Steghide 0.5.1** [32]: Developed by Stefan Hetzl, this open source software operates in the transform domain.

Previous attempts to compare the performance of these tools [10] exists but was insufficient because the tools were only compared in one of the three key metrics, making it impossible to tell which tool performed better than the others. Although *S-Tools*, *MP3Stego* [29], and *Steganos* were initially taken into consideration for the comparison analysis, they were ultimately excluded since their source code or generated executable was not readily available. Except for *MP3Stego*[29], whose source code could be found but was discovered to have a flaw which prevented it from supporting the bit rate of the audio files in the GZTAN [38] dataset.

## 5 DATASET SELECTION

In order to have a fair comparison of the steganography methods and tools selected in section 4, they must naturally all be applied to

<sup>1</sup><https://github.com/MatthijsReyers/steganography-analysis>

the same dataset. To determine what constitutes a good dataset for steganography research, the existing literature identified in section 3.1 was reviewed again to determine the following criteria.

- (1) **Audio recordings must be diverse both in types (ambient sound, speech, music) and origin.**  
Different steganography methods may perform differently on different types of audio [10], or even different genres of music [12], so to ensure a fair and useful comparison, different kinds of audio must be taken into consideration.
- (2) **Audio must be provided in a lossless format (e.g. WAV, AIFF, FLAC).**  
Lossy audio codecs like MP3 alter the audio signal [17], and thus make it more difficult to attribute what signal distortion/alterations were introduced by the method vs by the encoder. This can affect the evaluation metrics.
- (3) **Audio must be provided in sufficiently high bit rate (at least 256Kbps).**  
Certain steganography methods are designed to operate on high bit rate audio [45], so to ensure that these can be included in the comparison the dataset must at least match their required bit rate.
- (4) **Audio recordings must be at least 2 seconds in length.**  
Certain adaptive steganography methods such as [41, 45] will change their embedding rate depending on the audio signal, which means that to measure an embedding rate representative of the real world performance of the method, the average of a long period of time must be taken. 2 seconds was chosen somewhat arbitrarily to ensure the recordings would be more than long enough.
- (5) **Dataset must be in the public domain or free to use.**  
The most commonly used datasets are all free to use [4], which displays the importance researchers place on the accessibility of datasets for easy sharing and reproduction of the research.

### 5.1 Datasets review

The full list of reviewed papers and occurrences of commonly used datasets is not included in this paper but can be found in the spreadsheet<sup>1</sup> described in section 3.1.

A systematic review[4] performed in 2020 identified the following commonly used datasets: NOIZEUS [43], TIMIT [13, 45, 46], GTZAN [25, 38], CORPORA [14]. Additionally, it was also found that many papers create their own custom datasets [4] from sources such as the FMA (freemusicarchive.org) database [16], which might indicate that the researchers were unable to find a suitable dataset. The CMU\_ARCTIC [22] dataset was also found in existing papers during the literature review process [16] and has been included for review as well. Aside from previously used datasets, the databases on Kaggle<sup>2</sup> and Google Research<sup>3</sup>. were also searched for suitable audio datasets. For a comprehensive list of all datasets reviewed, see Table 3 in the appendix.

<sup>1</sup><https://docs.google.com/spreadsheets/d/e/2PACX-1vRZCvdOHhNPun-Zil2f77CBKt169S8ZKmkcIDNkWPssh4TxP1PgucmFvNaUysHM9bf-FSnUy5EO6v6-c/pubhtml>

<sup>2</sup><https://www.kaggle.com/>

<sup>3</sup><https://research.google.com/>

As can be seen in Table 3, it is clear that no single dataset currently matches all the criteria. However multiple datasets can be combined to fill in each others deficiencies, which is the approach this paper has taken. Ultimately TIMIT [13] and GTZAN [38] were chosen to provide speech and music recordings respectively because of their prevalence in existing audio steganography research [4]. Their prevalence is relevant because it increases the chance that the results of this paper can be compared to a previously published paper. Additive background noise samples from the ESC-50 [30] dataset were chosen to provide realistic background noise for the robustness evaluation phase. ESC-50 [30] was chosen over AURORA-5 because it is freely available on GitHub while AURORA-5 requires researchers/their universities to buy a licence.

## 6 EVALUATION FEATURES SELECTION

As mentioned previously in section 2, there are many different evaluation features that attempt to measure a steganography method's performance in one of the three key metrics. Unfortunately there is no consensus among steganography researchers which features are most suitable and many papers thus include different features or even no features for a given key metric [4]. This means two given papers often cannot be directly compared.

To address this problem this paper determined the best feature for evaluating each of the key metrics by first making a list of which features were used in the papers found during the literature review, and then determining the pros and cons of each feature.

A full review of all the discovered evaluation features for *Robustness*, *Perceptual transparency*, and *Hiding capacity* can be found in the appendix in Tables 4, 5, and 6 respectively. Where the evaluation feature selected for use in the analysis phase is marked with a green check mark (✓). The results of the review were as follows:

### 6.1 Robustness - BER

This paper measures robustness with BER (Bit Error Rate) in percentage (%), since for the papers that do provide some kind of robustness testing, BER is by far the most commonly used feature [17, 41, 42]. It makes sense to follow this convention as it could allow for the results of this paper to be compared with other papers.

BER is computed using the following formula [20] where  $b_{err}$  is the number of incorrect bits in the extracted secret data and  $b_{total}$  is the total number of secret bits that were embedded:

$$BER = \frac{b_{err}}{b_{total}} \times 100\% \quad (1)$$

It is important to be aware that some papers do not feature any robustness analysis and instead use BER to measure how many bits were changed in the cover audio after the embedding process in order to measure perceptual transparency. That is *not* what this paper uses BER for, in this paper BER always means the percentage of erroneously extracted bits in the extracted secret text.

### 6.2 Perceptual transparency - SNR

This paper measures perceptual transparency using SNR (Signal to Noise Ratio). Although it would probably be wise to also include a subjective evaluation feature like PESQ to measure directly how

detectable the introduced noise is to the HAS (human auditory system), and many papers do indeed use both [8, 16], the work required for performing such a survey is not feasible for the scope of this paper.

SNR was chosen because it was the only one for which an acceptable value was found (30db or higher [17]). The possible range of SNR is linked directly to the bit depth of the audio in use since this determines the possible values  $x_i$  and  $y_i$  can take. In this case the audio datasets chosen in section 5 both have a bit depth of 16 bits and all SNR values in this report are thus be between  $-\infty db$ , and 96.32db. SNR is computed using the following formulas:

$$SNR = 10 \log_{10} \left( \frac{XS}{MSE} \right), \quad (2)$$

$$\text{where } MSE = \frac{1}{N} \sum_{i=1}^n (x_i - y_i)^2$$

$$XS = \frac{1}{N} \sum_{i=1}^n x_i^2$$

Where  $x$  and  $y$  are the cover and stego audio signals respectively,  $N$  is the number of samples in a signal, and  $x_i$  and  $y_i$  are the  $i^{\text{th}}$  sample of  $x$  and  $y$  respectively.

### 6.3 Hiding capacity - EP

This paper measures hiding capacity in EP (Embedding Percentage), because unlike ER (Embedding Rate,  $kb/s$ ) or BC (Bit Count,  $bits$ ), it is independent of the length and bit rate of the carrier audio.

This means it gives a reasonably accurate indication of the relative performance between the hiding capacity of other methods and research papers even when these used a different dataset. Additionally it also has a clearly defined range (0% tot 100%) making it easy to understand where a result lies relative to the theoretical best performance (100%). (Although this theoretical best performance would of course have terrible perceptual transparency as there would effectively be nothing left of the cover audio). EP is computed using the following formula:

$$EP = \frac{b_{\text{changed}}}{b_{\text{total}}} \times 100\% \quad (3)$$

Where  $b_{\text{changed}}$  is the number of bits that have changed between the cover and stego audio, and  $b_{\text{total}}$  is the total number of bits in the stego audio.

## 7 RESULTS

The following section contains visualizations and explanations of the evaluation data generated for the comparison. For a detailed description on how this evaluation data was generated please see section 3.5 and for the conclusions derived from these results, please see section 8.

### 7.1 Hiding capacity results

Out of concern that the character distribution of the secret data might affect the performance of the machine learning based GAN method, a small test with three different types of secret text was performed. 100 paragraphs of Lorem ipsum text, randomly generated

ascii characters, and the Sherlock Homes story *A study in scarlet* were used, the results of which are shown in Table 1.

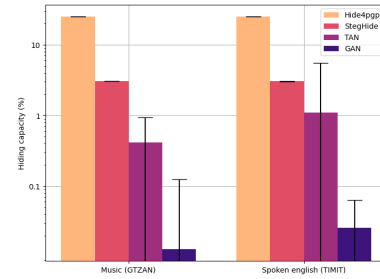
Table 1. Effect of different secret text types on hiding capacity of GAN method, in %, formatted as mean $\pm$ std.

Secret text type	Spoken English (TIMIT)	Music (GTZAN)
Lorem ipsum	0.010 $\pm$ 0.005	0.006 $\pm$ 0.004
Random ascii	0.018 $\pm$ 0.009	0.007 $\pm$ 0.006
Sherlock Homes	0.026 $\pm$ 0.014	0.013 $\pm$ 0.017

Because the Sherlock Homes story gave the best performance and arguably has the most realistic character distribution since it is real English text, the rest of the evaluations were performed using this as the secret text.

A full comparison of the performance of the maximum hiding capacity of the different tools and methods can be seen in Figure 2. A logarithmic scale was used on the y-axis so the smaller values can be seen in more detail, while the exact numbers can be found in Table 2.

Fig. 2. Maximum hiding capacity of different methods/tools, please note the use of a logarithmic y-axis.



As is clear from Figure 2 and Table 2, Hide4PGP has by far the biggest embedding percentage of all the methods, averaging almost exactly 25%. To put this number in context, Hide4PGP was able to fit the text of *A study in scarlet* ( $\approx 236$  kiB) multiple times inside a 30 second wav file (mono, 353 kbit/s). This might be useful if you want to embed large files like images, but if the goal is only to embed a short text message all of the methods will suffice.

Table 2. Maximum hiding capacity of different methods in % formatted as mean $\pm$ std.

	Spoken English (TIMIT)	Music (GTZAN)
<b>Hide4PGP</b>	24.999 $\pm$ 0.000	24.989 $\pm$ 0.003
<b>StegHide</b>	3.053 $\pm$ 0.001	3.044 $\pm$ 0.042
<b>TAN</b>	0.013 $\pm$ 0.112	0.026 $\pm$ 0.037
<b>GAN</b>	0.415 $\pm$ 0.519	1.091 $\pm$ 4.392

Neither of the tools are able to vary the embedding rate depending on the audio and they thus achieved the same hiding capacity within regardless of the genre of music. Only StegHide has a slight amount



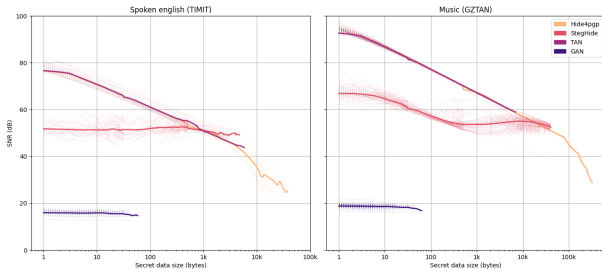
of variance for the TIMIT dataset because it failed to embed all the secret bytes it claimed to be able to embed, as described in section 3.5.2.

### 7.2 Perceptual transparency results

The amount of secret data that is embedded in the cover audio influences how much distortion is introduced to the stego audio. Because of this the SNR of stego files with different sizes of secret messages was computed as described in section 3.5.3. This data has been visualized in Figure 3 which shows the relationship between the secret message size (x-axis) and the SNR value for the different techniques. Previous research [17] has established that the recommended SNR for an audio signal is 30db or higher.

And while only anecdotal evidence, all values above 30db were indeed not perceptible to the researcher.

Fig. 3. Average SNR at different secret message sizes, dotted lines represent individual audio files to illustrate the variance in results.



All methods performed slightly better on the GZTAN dataset, this makes sense since the TIMIT dataset contains lots of silences in between the speech and adding noise to silence is more noticeable than adding noise to a loud signal like music.

Hide4PGP and TAN performed the best at most message sizes. Their overlap in the graph can likely be attributed to the fact that they are both LSB-1 methods. This means that the pseudo random distribution of the secret bits used by TAN did not create any noticeable SNR improvement over the strategy used by Hide4PGP.

The GAN paper [44] notably also used SNR as a metric for perceptual transparency, but unfortunately did not describe how SNR was computed or if it was in decibels. When using the assumption that they used pure ratio, their results can be converted to decibels to find they achieved an SNR of about  $10\log_{10}(7.24) = 8.6\text{dB}$  on the TIMIT dataset. This is significantly worse than the results in this paper, which might be due to the fact that our model was trained on a different dataset or because they used a formula for SNR that cannot be converted to decibels this way. The relationship between secret message size and SNR for the GAN method is fairly constant, since the GAN network always seems to modify the spectrogram roughly the same amount regardless of message size. The line for GAN stops relatively early because this paper used only 2 seconds of audio for the spectrogram, but it is easy to imagine a sliding window system which would allow for larger message sizes to be reached.

### 7.3 Robustness results

The robustness of the different techniques against various levels of background noise was computed as described in section 3.5.4. This data was then visualized in Figure 4, which shows the BER (on y-axis) at various levels of noise (on the x-axis).

Fig. 4. BER (bit error rate) of the methods at different levels of noise, lower is better.

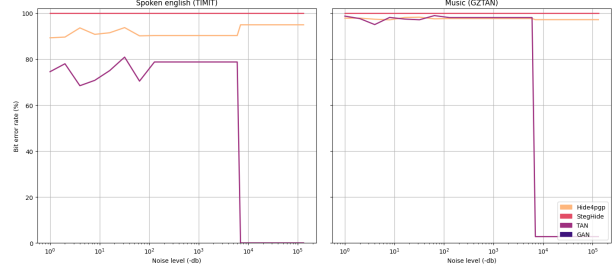
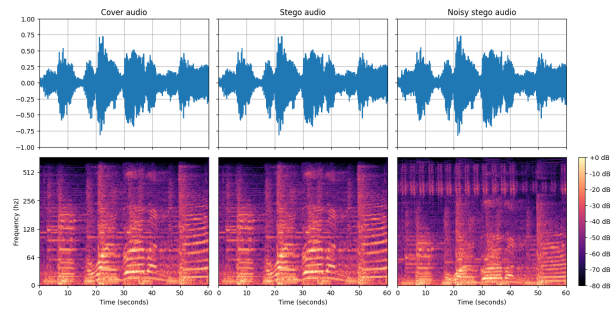


Fig. 5. Waveform and spectrogram of the cover, stego, and noisy stego audio for StegHide. File: blues.00000.wav, noise at -8dB.



As can be seen Figure 4, essentially all the tools and methods were unable to deal with this distortion. Only TAN was eventually able to decode the secret message after the level of the noise became so low that the least significant bits of the audio file were left unchanged.

It is unknown why the other techniques were still unable to extract the secret message at this point as the stego file should have been essentially left unchanged. This might point to a possible flaw in the testing methodology, however such a flaw could not be identified.

Hide4PGP appears to perform better than the others, however this is only the case because Hide4PGP does not include a checksum or error correcting code, meaning that it cannot detect whether it is extracting a secret message or random bits. And since some of those random bit will happen to be the right bits by pure chance the BER is not 100%, but at around 70% BER the extracted output is essentially unrecognizable.

The other methods and tools do include some kind of checksum or error correcting codes which allows them to throw an error rather than output random data, but since they were unable produce any output, their performance was counted as 100% BER. Depending on the use case this may actually preferable since the integrity of the extracted message is ensured this way.

## 8 CONCLUSIONS

In regards to research question 1, this paper reviewed the datasets most commonly used in steganography research and determined in section 5 that the TIMIT [13], GZTAN [38], and ESC-50 [30] datasets are most suitable for the evaluation of steganography techniques. Where TIMIT[13] is used to represent speech recordings, GZTAN [38] represents music recordings, and ESC-50 [30] is used for sourcing realistic background noise for robustness testing.

For research question 2 it was determined in section 6 that the evaluation features most suitable for measuring each of the key metrics are EP, SNR, and BER. With the addition of a subjective feature like PESQ when possible.

Future researches can use these recommendations for datasets and evaluation features to evaluate their new steganography methods in such a way that the obtained results are compatible with this paper. Meaning that the performance of their method can be directly compared against the performance of the method evaluated in this paper.

For research question 3; the limitation and challenges of new audio steganography methods, it was found in the previous section that the new methods in this paper (GAN and TAN) did not manage to outperform the older tools (StegHide and Hide4PGP) in any of the key metrics. This means that people seeking to apply these new steganography methods are better off using one of the existing tools instead or investigating other new steganography methods. For researchers working on their own new method or comparative analysis it means that older tools such as StegHide and Hide4PGP are still worth including in the comparison as despite their age they are still relevant.

## 9 DISCUSSION

The fact that non of the new method managed to outperform the existing tools is perhaps rooted in the method selection. There are other papers built around GAN which claim to achieve even better performance, but these could not be included in this research due to issues in obtaining the source code. Additionally the model used for the GAN method was not trained on the GZTAN dataset so it is possible that the GAN method might have performed better if it had been trained on this dataset as well.

The logistic tan map variation of LSB coding that was analyzed in this paper was designed to improve the security of the message by making it more difficult to extract. The order of the secret bits is essentially shuffled after all. However, this shuffling does not necessarily improve perceptual transparency, There exist other variations on LSB coding that aim to reduce perceptual transparency, it is possible that these might get a better SNR score and manage to beat StegHide and TAN.

Because of the above mentioned reasons the results of analysis should not be taken as proof that no improvements have been made in audio steganography in the nearly two decades since StegHide and Hide4PGP were first developed. Rather, it simply means that more research need to be done and more methods need to be compared to determine which method is the current state of the art.

The fact that none of the methods proved resilient against even the slightest amount of additive noise points to a possible fault

with the robustness testing methodology used. Additionally other types of signal corruption such as high pass filters or phone line emulation were considered but could not be included in the paper because of time constraints, as such the robustness analysis may not be representative of the real world.

### 9.1 Limitations and future work

Some limitations of this research and other possible avenues for future research are summarized in a list below:

- **Computational intensity metric:** As alluded to in the introduction, audio steganography is uniquely suited for real time communication, a fourth key metric to measure the computational intensity of a given method could be considered to evaluate if a method is suitable for real time communication and how much CPU power such a system would require.
- **Steganalysis tools:** Due to time constraints this paper did not investigate the ability of the methods to evade detection by the state of the art steganalysis tools, it is possible that the adversarial GAN based method would have outperformed the other methods in this kind of testing. Future research should include this kind of testing.
- **Secret text types:** Modern machine learning based steganography methods likely perform differently based on the type/distribution of the secret text used, future research could investigate how large these differences are and what the implications are for the training data selection.
- **More methods:** This comparative analysis included only two methods due to time constraints and source code availability, future work could reuse the automated scripts developed for this paper to save time and focus on comparing many different methods instead.
- **Robustness testing:** There is no standard way to evaluate robustness, this research used additive background noise, but other signal degradation scenarios such as high pass filters, phone line simulation, and re-encoding could also be tested. Ideally a whole literature review could be dedicated to researching all the different types of signal degradation and which ones are the most relevant to different usage scenarios.

## ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Dipti K. Sarmah, for her guidance and support throughout this project.

## REFERENCES

- [1] 2023. Hide4PGP tool. *Visited on: 2023-06-01 (2023)*. <http://www.heinz-repp.onlinehome.de/Hide4PGP.html>
- [2] Abdulaleem Z Al-Othmani, Azizah Abdul Manaf, and Akram M Zeki. 2012. A survey on steganography techniques in real time audio signals and evaluation. *International Journal of Computer Science Issues (IJCSI)* 9, 1 (2012), 30.
- [3] A. H. Ali, L. E. George, A. A. Zaidan, and M. R. Mokhtar. 2018. High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain. *Multimedia Tools and Applications* 77, 23 (2018), 31487–31516. <https://doi.org/10.1007/s11042-018-6213-0>
- [4] A. A. AlSabhan, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar. 2020. Digital audio steganography: Systematic review, classification, and analysis of the current state of the art. *Computer Science Review* 38, 100316 (2020). <https://doi.org/10.1016/j.cosrev.2020.100316>
- [5] M. Asad, J. Gilani, and A. Khalid. 2011. An enhanced least significant bit modification technique for audio steganography. *International Conference on Computer*



- Networks and Information Technology* (2011), 143–147. <https://doi.org/10.1109/ICCNIT.2011.6020921>
- [6] S. S. Bharti, M. Gupta, and S. Agarwal. 2019. A novel approach for audio steganography by processing of amplitudes and signs of secret audio separately. *Multimedia Tools and Applications* 78, 16 (2019), 23179–23201. <https://doi.org/10.1007/s11042-019-7630-4>
- [7] L. Chen, R. Wang, L. Dong, and D. Yan. 2023. Imperceptible adversarial audio steganography based on psychoacoustic model. *Multimedia Tools and Applications* (2023). <https://doi.org/10.1007/s11042-023-14772-9>
- [8] L. Chen, R. Wang, D. Yan, and J. Wang. 2021. Learning to Generate Steganographic Cover for Audio Steganography Using GAN. *IEEE Access* 9 (2021), 88098–88107. <https://doi.org/10.1109/ACCESS.2021.3090445>
- [9] A. Das. 2012. Steganography: Secret Data Hiding in Multimedia. In *A. Das (Ed.), Signal Conditioning: An Introduction to Continuous Wave Communication and Signal Processing* Springer (2012), 275–295. [https://doi.org/10.1007/978-3-642-28818-0\\_11](https://doi.org/10.1007/978-3-642-28818-0_11)
- [10] F. Djebbar, B. Ayad, K. A. Meraim, and H. Hamam. 2012. Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing* 2012, 25 (2012). <https://doi.org/10.1186/1687-4722-2012-25>
- [11] H. Dutta, R. K. Das, S. Nandi, and S. R. M. Prasanna. 2020. An Overview of Digital Audio Steganography. *IETE Technical Review* 37, 6 (2020), 632–650. <https://doi.org/10.1080/02564602.2019.1699454>
- [12] M. T. Elkandoz and W. Alexan. 2019. Logistic Tan Map Based Audio Steganography. *International Conference on Electrical and Computing Technologies and Applications (ICECTA)* 2019 (2019), 1–5. <https://doi.org/10.1109/ICECTA48151.2019.8959683>
- [13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, D. S. Pallett, N. L. Dahlgren, V. Zue, and J. G. Fiscus. 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus. *Linguistic Data Consortium* (1993). <https://doi.org/10.35111/17gk-bn40>
- [14] S. Grocholewski. 1997. CORPORA - speech database for Polish diphones. *5th European Conference on Speech Communication and Technology (Eurospeech 1997)* (1997), 1735–1738. <https://doi.org/10.21437/Eurospeech.1997-492>
- [15] S. Gupta, A. Goyal, and B. Bhushan. 2012. Information hiding using least significant bit steganography and cryptography. *International Journal of Modern Education and Computer Science* 4, 6 (2012), 27. <https://doi.org/10.5815/ijmecs.2012.06.04>
- [16] A. S. Hameed. 2021. A High Secure Speech Transmission Using Audio Steganography and Duffing Oscillator. *Wireless Personal Communications* 120, 1 (2021), 499–513. <https://doi.org/10.1007/s11277-021-08470-8>
- [17] S. Hemalatha and Ramathika. 2020. A Robust MP3 Audio Steganography with Improved Capacity. *IEEE 5th International Conference on Computing Communication and Automation (ICCCA)* 2020 (2020), 640–645. <https://doi.org/10.1109/ICCCA49541.2020.9250894>
- [18] F. Hemeida, W. Alexan, and S. Mamdouh. 2021. A Comparative Study of Audio Steganography Schemes. *International Journal of Computing and Digital Systems* 10 (2021), 555–562. <https://doi.org/10.12785/ijcds/100153>
- [19] M. Hilbert. 2014. What Is the Content of the World’s Technologically Mediated Information and Communication Capacity: How Much Text, Image, Audio, and Video? *The Information Society* 30, 2 (2014), 127–143. <https://doi.org/10.1080/01972243.2013.873748>
- [20] R. Indrayani. 2020. Modified LSB on Audio Steganography using WAV Format. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)* (2020), 466–470. <https://doi.org/10.1109/ICOIACT50329.2020.9332132>
- [21] H. J. Kim and Y. H. Choi. 2003. A novel echo-hiding scheme with backward and forward kernels. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 8 (2003), 885–889. <https://doi.org/10.1109/TCSVT.2003.815950>
- [22] J. Kominek and A. Black. 2004. The CMU Arctic speech databases. *SSW5-2004* (2004).
- [23] Y. Lin, R. Wang, D. Yan, L. Dong, and X. Zhang. 2019. Audio Steganalysis with Improved Convolutional Neural Network. *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security* (2019), 210–215. <https://doi.org/10.1145/3335203.3335736>
- [24] H. Liu, J. Liu, R. Hu, X. Yan, and S. Wan. 2017. Adaptive Audio Steganography Scheme Based on Wavelet Packet Energy. *Ieee 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity)* (2017), 26–31. <https://doi.org/10.1109/BigDataSecurity.2017.20>
- [25] M. M. Mahmoud and H. T. Elshoush. 2022. Enhancing LSB Using Binary Message Size Encoding for High Capacity, Transparent and Secure Audio Steganography—An Innovative Approach. *IEEE Access* 10 (2022), 29954–29971. <https://doi.org/10.1109/ACCESS.2022.3155146>
- [26] M. A. Majeed, R. Sulaiman, Z. Shukur, and M. K. Hasan. 2021. A Review on Text Steganography Techniques. *Mathematics* 9, 21 (2021), Article 21. <https://doi.org/10.3390/math9212829>
- [27] S. Mishra, V. K. Yadav, M. C. Trivedi, and T. Shirmali. 2018. Audio Steganography Techniques: A Survey. *Advances in Computer and Computational Sciences* (2018), 581–589. [https://doi.org/10.1007/978-981-10-3773-3\\_56](https://doi.org/10.1007/978-981-10-3773-3_56)
- [28] Hussein A Nassrullah, Wameedh Nazar Flayyih, and Mohammed A Nasrullah. 2020. Enhancement of LSB Audio Steganography Based on Carrier and Message Characteristics. *J. Inf. Hiding Multim. Signal Process.* 11, 3 (2020), 126–137.
- [29] Fabien Peticolas. 1998. mp3stego. <http://www.cl.cam.ac.uk/~fapp2/steganography/mp3stego/index.html> (1998).
- [30] K. J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. *Proceedings of the 23rd ACM International Conference on Multimedia* (2015), 1015–1018. <https://doi.org/10.1145/2733373.2806390>
- [31] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. 2001. Perceptual evaluation of speech quality (PESQ)—a new method for speech quality assessment of telephone networks and codecs. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)* 2 (2001), 749–752. <https://doi.org/10.1109/ICASSP.2001.941023>
- [32] Hetzl S. 2023. Steghide tool. *Visited on: 2023-06-01* (2023). <https://steghide.sourceforge.net/index.php>
- [33] S. B. Sadkhan, A. A. Mahdi, and R. S. Mohammed. 2019. Recent Audio Steganography Trails and its Quality Measures. *2019 First International Conference of Computer and Applied Sciences (CAS)* (2019), 238–243. <https://doi.org/10.1109/CAS47993.2019.9075778>
- [34] A. K. Sahu and M. Sahu. 2020. Digital image steganography and steganalysis: A journey of the past three decades. *Open Computer Science* 10, 1 (2020), 296–342. <https://doi.org/10.1515/comp-2020-0136>
- [35] H. Shivaram, D. Acharya, R. Adige, S. Deepthi, and K. Upadhy. 2015. Audio steganography in discrete wavelet transform domain. *International Journal of Applied Engineering Research* 10 (2015), 37544–37549.
- [36] D. Tan, Y. Lu, X. Yan, and X. Wang. 2019. A Simple Review of Audio Steganography. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (2019), 1409–1413. <https://doi.org/10.1109/ITNEC.2019.8729476>
- [37] G. P. TVS and S. Varadarajan. 2015. A novel hybrid audio steganography for imperceptible data hiding. *International Conference on Communications and Signal Processing (ICCSPP)* Springer (2015), 0634–0638. <https://doi.org/10.1109/ICCSPP.2015.7322565>
- [38] George Tzanetakis, Georg Essl, and Perry Cook. 2001. Automatic Musical Genre Classification Of Audio Signals. <http://ismir2001.ismir.net/pdf/tzanetakis.pdf>
- [39] J. Wang, R. Wang, L. Dong, and D. Yan. 2020. Robust, Imperceptible and End-to-End Audio Steganography Based on CNN. In *S. Yu, P. Mueller, J. Qian (Eds.), Security and Privacy in Digital Economy* Springer (2020), 427–442. [https://doi.org/10.1007/978-981-15-9129-7\\_30](https://doi.org/10.1007/978-981-15-9129-7_30)
- [40] Y. Wang, K. Yang, X. Yi, X. Zhao, and Z. Xu. 2018. CNN-based Steganalysis of MP3 Steganography in the Entropy Code Domain. *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security* (2018), 55–65. <https://doi.org/10.1145/3206004.3206011>
- [41] G. Xin, Y. Liu, T. Yang, and Y. Cao. 2018. An Adaptive Audio Steganography for Covert Wireless Communication. *Security and Communication Networks* 10 (2018), e7096271. <https://doi.org/10.1155/2018/7096271>
- [42] P. Xue, H. Liu, J. Hu, and R. Hu. 2018. A Multi-Layer Steganographic method Based on Audio Time Domain Segmented and Network Steganography. *AIP Conference Proceedings* 1967, 1 (2018). <https://doi.org/10.1109/ICCCA49541.2020.9250894>
- [43] Hu Y and Loizou P. 2007. Subjective evaluation and comparison of speech enhancement algorithms. *Speech communication* 49 (2007), 588–601.
- [44] D. Ye, S. Jiang, and J. Huang. 2019. Heard More Than Heard: An Audio Steganography Method Based on GAN. *arXiv arXiv:1907.04986* (2019). <https://doi.org/10.48550/arXiv.1907.04986>
- [45] K. Ying, R. Wang, Y. Lin, and D. Yan. 2021. Adaptive Audio Steganography Based on Improved Syndrome-Trellis Codes. *IEEE Access* 9 (2021), 11705–11715. <https://doi.org/10.1109/ACCESS.2021.3050004>
- [46] R. Zhang, H. Dong, Z. Yang, W. Ying, and J. Liu. 2022. A CNN Based Visual Audio Steganography Model. *Artificial Intelligence and Security* (2022), 431–442. [https://doi.org/10.1007/978-3-031-06794-5\\_35](https://doi.org/10.1007/978-3-031-06794-5_35)

## A COMPLETE TABLES OF ALL REVIEWED DATASETS AND EVALUATION FEATURES.

Table 3. All reviewed datasets and their respective deficiencies based on the numbered criteria established in section 5.

Name	Description	Deficiencies
<i>NOIZEUS</i> [43]	Consists of 30 spoken sentences corrupted with eight different real-world noise signals taken from the AURORA dataset, which includes noise recordings such as suburban vehicles and ambient street noise.	- (1) Includes only spoken English and it thus not representative of other languages or audio types like music. - (3) Audio is provided at 128 kb/s.
<i>TIMIT</i> [7, 13, 46]	Consists of recordings of 630 different speakers recorded at 256 kb/s. This dataset is free for non-commercial use and was designed for the development and evaluation of automatic speech recognition systems.	- (1) Includes only spoken English and it thus not representative of other languages or audio types like music.
<i>GTZAN</i> [38]	Consists of music files in 10 genres with 100 audio files each, all at a length of 30 seconds and bitrate of 352 kb/s. This dataset was originally created for the development and evaluation of genre detection machine learning systems.	- (1) Genres in this dataset include primarily western styles (classical, blues, hiphop, etc.) and is thus not representative of traditional music from other cultures or indigenous groups.
<i>CORPORA</i> [14]	Consists of polish speech recordings of 45 different speakers saying 365 different utterances (letters, digits, 200 names, and 114 sentences). This dataset was originally designed for the development and evaluation of automatic speech recognition systems.	- (1) Includes only spoken Polish and it thus not representative of other languages or audio types like music. - (5) Unknown copyright status.
<i>Speaker Recognition Dataset</i>	Found on Kaggle, this dataset contains recordings of speeches of five prominent leaders. Each recording is 1 second long and has a bitrate of 256 kb/s.	- (1) Includes only spoken English and it thus not representative of other languages or audio types like music.
<i>CMU_ARCTIC</i> [22]	Consists of around 1150 utterances selected from out-of-copyright texts recorded at a bitrate of 256 kb/s.	- (1) Includes only spoken English and it thus not representative of other languages or audio types like music.
<i>AURORA-5</i>	An extension of the TI-Digits speech dataset where the samples have been distorted in ways representative of the real world, such as simulated cellular network transmission and different types of additive background noise.	- (1) Includes only spoken English and background noise and it thus not representative of other languages or audio types like music. - (5) Although technically free to use, still requires a paid academic subscription to access.
<i>ESC-50</i> [30]	An collection of 2000 environmental audio recordings, all 5 seconds long and recorded at 44.1 kHz.	- (1) Includes only environmental noise and no speech or music.

Table 4. Discovered robustness evaluation features. The feature selected for use in this research is marked with green check mark.


Name	Description	Range	Evaluation
<i>ACC</i> [39]	<b>Accuracy of message extraction.</b> Ranging from where 100% means all secret data was extracted without error, (essentially the inverse of BER).	(0%, 100%)	+ Easily understood metric, commonly found in machine learning papers. - Not commonly used in steganography papers, found in only one reviewed paper.
<i>BER</i> [17, 41, 42]	 <b>Bit Error Rate.</b> Percentage of bits erroneously transmitted or received, essentially the inverse of accuracy, where 0% BER means 100% ACC.	(0%, 100%)	+ Easily understood metric. + Very commonly used in steganography papers, included in around half of all reviewed papers.

Table 5. Discovered perceptual transparency evaluation features. The feature selected for use in this research is marked with green check mark.

Name	Description	Range	Evaluation
MSE [12, 16]	<b>Mean Squared Error.</b> The error level between the original cover audio signal and the produced stego audio signal. Where lower values mean less noise.	$(0, 65535^2)$ for 16-bit audio	+ Objective and easily calculated. + Easily understood metric. - Unlimited range, making it difficult to interpret when noise becomes audible.
PSNR [46]	<b>Peak signal-to-noise ratio.</b> Represents the ratio between the maximum power of the cover audio and the noise/distortion added during the steganography process in decibels. (Note that PSNR is essentially MSE adjusted for signal power).	$(-\infty db, 96.32 db)$ for 16-bit audio	+ Objective and easily calculated. + Uses the logarithmic decibel unit, more representative of how the HAS perceives sound.
SNR [16, 17]	<input checked="" type="checkbox"/> <b>Signal to Noise Ratio.</b> Represents the ratio between the maximum power of the cover audio and the noise/distortion added during the steganography process in decibels. Where $\infty db$ is no noise and $0 db$ is pure noise, with the recommended SNR for an audio signal being around $30 db$ or higher [17].	$(-\infty db, 96.32 db)$ for 16-bit audio	+ Objective and easily calculated. + Uses the logarithmic decibel unit, more representative of how the HAS perceives sound.
PESQ [16, 25, 31]	<b>Perceptual Evaluation of Speech Quality.</b> Subjective quality assessment based on opinion scores. Part of an international standard, which prescribes that PESQ scores should be $> 3.5$ .	(1.0, 4.5)	- Based on opinion scores that requires many participants/much time to be accurate + Commonly used, included 50% of reviewed papers. + International standard and ITU-T recommendation.
SPCC [17]	<b>Squared Pearson Correlation Coefficient.</b> Used for measuring the quality of an audio signal based on the correlation of samples. Higher values indicate higher quality.	(0, 1)	+ Objective and easily calculated. - Uncommon, found in only one reviewed paper.
CCf [16]	<b>Correlation coefficient factor.</b> Statistical quantity for measuring the similarity between two signals, where 1 means the signals are identical.	(-1, 1)	+ Objective and easily calculated. - Uncommon, found in only one reviewed paper.

Table 6. Discovered capacity evaluation evaluation features. The feature selected for use in this research is marked with green check mark.

Name	Description	Review
EP, % [16, 25]	<input checked="" type="checkbox"/> <b>Embedding Percentage.</b> What percentage of bits in the stego audio file are used for embedding the secret data. Where 0% means none of the bits are used for embedding the secret data, i.e. the cover audio is left unchanged.	+ Commonly found in machine learning papers. + Easily compared and understood because of the well defined range.
BC, bits [20]	<b>Bit count.</b> The total/absolute number of bits in the stego audio file that are used for embedding the secret data.	- Highly depended on the length of the used cover audio. - Not easily compared because of large values.
ER, kb/s [8]	<b>Embedding Rate.</b> The average number of secret bits hidden in the stego file per second, often defined in Kilo-bits/second.	- Intrinsically linked to the bit rate of the cover audio. + Also used for denoting audio bit rates. + Commonly used in steganography papers.