ISSN 1870-4069

Steganography in Frequency Domain: Hiding Text through Audio Spectrogram

Luis Enrique Morales-Márquez

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

luise.morales@viep.com.mx

Abstract. Steganography aims to hide information in digital media so that it is imperceptible, among the most used methods is the use of LSB, and hiding bits of a message in spectral peaks at high frequencies can help improve imperceptibility. This study takes some test audios, obtains the spectrogram and hides bits of a message in the peaks at high frequencies, it is concluded that the performance in terms of MSE, SNR, PSNR and SSIM are generally good since they preserve the quality of the carrier audio of the information.

Keywords: Frequency domain, steganography, spectrograms, LSB.

1 Introduction

Steganography aims to hide secret messages or data via digital media such as images, audio clips, or video sequences. In this way, the information that needs to be discreetly transferred is protected from unauthorized individuals, while keeping the appearance of the multimedia file containing it unchanged. In all digital media, the most commonly used technique is embedding in the least significant bits (LSB) [1]. There are some key properties that must be addressed in these procedures:

- Embedding capacity: The amount of data that can be hidden in the cover media in relation to its size. More information cannot be hidden than the information that is contained in the cover itself.
- Undetectability: The data must be inserted in such a way that the secret message cannot be accidentally perceived when playing or observing the file containing it. If the message is detected at first glance, the steganography has failed.
- Robustness: Ability to withstand methods attempting to retrieve the secret message. The recovery of information should not be simple, but neither should it be complicated for the authorized recipient [2].

The success of audio steganography depends on the behavior of the Human Auditory System (HAS), as it is more sensitive to changes than the visual system [3]. Therefore, special care must be taken when using audio media for data hiding.

The hiding method proposed in this article involves obtaining the spectrogram of the audio file and selecting the elements with the highest energy at high frequencies to hide

bits of a text string in the 3 LSB of the high spectral peaks. The audio is then recovered from the modified spectrogram, obtaining the stego audio.

The structure of the article is as follows: Section 2 briefly presents related work on audio steganography; Section 3 details the proposed method and associated theoretical concepts, as well as metrics for evaluating the quality of steganography; Section 4 shows the results obtained, which are analyzed in Section 5; and finally, Section 6 presents the conclusions derived from this study.

2 Related Work

Information hiding in audio files has been widely explored, just as in image or video files. Below are some examples of audio steganography.

The technique of hiding information in the LSB is one of the most widely used in steganography for all types of media. Hussian, J., & Farhan, K. [4] designed a model in 2016 that generates a random sequence of bits, modifying only 2 bits in each audio window. They then generate a HASH of the key and apply an XOR operator with the bits of the data to be hidden. This result is embedded in the bits selected by a random generator.

On the other hand, Chua, T. et al. [1] in 2017 considered that, to maintain the imperceptibility of the embedded messages, it is appropriate to encrypt the secret message using the RC4 algorithm and a password defined by the sender. Subsequently, the bits of the encrypted message are inserted into the selected audio. The recipient must know the encryption password to obtain the secret message back.

In 2021, Zainab, N., & Ban, N. [5] proposed an indirect LSB insertion method, which consists of obtaining the lengths of the audio and the text to be attached. The message must not exceed an eighth part of the length of the audio. After encrypting the message, it is processed with the XOR operator with bits of several prime numbers. The encoded message is compared to the first bit of the position indicated in the audio of a sequence of numbers. If they are equal, the message bit is embedded in the least significant bit of the position indicated in the audio.

Using pre-trained neural networks, Galeta, M. et al. [6] in 2021 employed a residual network to encode an image selected as the secret message and add it to the audio spectrogram. The recovery of the attached image is also the responsibility of a residual neural network. This allows for a larger area for bit insertion than a one-dimensional signal, such as the traditional audio representation. The spectrogram is performed with the cosine transform.

Finally, Abood, E. et al. [7] in 2022 developed a hybrid model. The message is encrypted with a bit-swapping technique based on a key hidden in the audio in the time domain. The insertion of the key is done in LSB at random positions in this domain. Since the sampling rate in the time domain determines the number of points representing the signal, there are a large number of bins in which insertion can be performed.

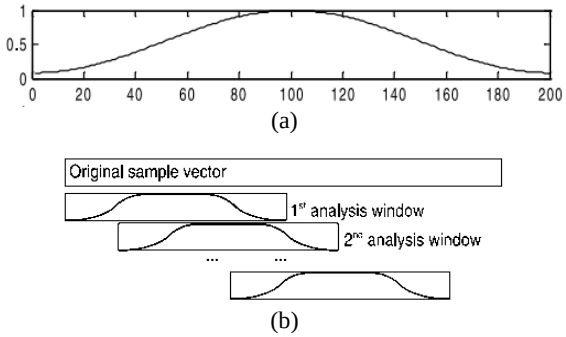


Fig 1. (a) Hamming window, the amplitude is represented on the *y* axis and the number of elements of the window on the *x* axis. (b) Overlapping of the blocks resulting from the windowing [11].

3 Proposed Method

LSB techniques are easy to understand and have been widely developed. The simplest way to apply this method is to sample the signal in the time domain and select the peaks with the highest amplitude. However, this idea may result in a relatively easy extraction of the hidden data. Therefore, in this work, we propose to insert the data inside the audio spectrogram by hiding them in the frequency domain, specifically at high frequencies. It is based on the fact that the human ear has a hearing range that goes from 15Hz to some value between 15kHz and 20kHz, depending on the individual.

In general, we should not be able to perceive audio with alterations at high frequencies, as even though an individual could exceed the 20kHz barrier, their hearing quality at such a high frequency is very poor [8]. This means that, in theory, an audio file with an embedded message should sound almost exactly the same as an unaltered audio and the changes should be detectable only through analysis, to achieve that objective, the following method is proposed.

3.1 Insertion Stage

First, the WAV audio file is selected and sampled at 44.1kHz, a standard sampling frequency that obeys the sampling theorem. This indicates that sampling must be performed at least twice the signal frequency to avoid aliasing phenomenon [9]. In this way, we slightly exceed twice the human audible limit of 20kHz. Then, the signal windowing process is made as follows: the audio is segmented into 256-unit windows using the Hamming-type function, which allows for greater frequency resolution and, therefore, better storage capacity at high frequencies [10]. We choose 256 bins in the window to maintain a balance between the number of blocks to work with and the block size. Additionally, we define a 50% block overlap, which ensures signal continuity during analysis and reconstruction [11] (see Figure 1).

We represent the energy value of a certain frequency at a specific time (see Figure 2), this is called a spectrogram and is obtained through the Short-Time Fourier Transform (STFT) applied to each signal block obtained in the previous step. The STFT

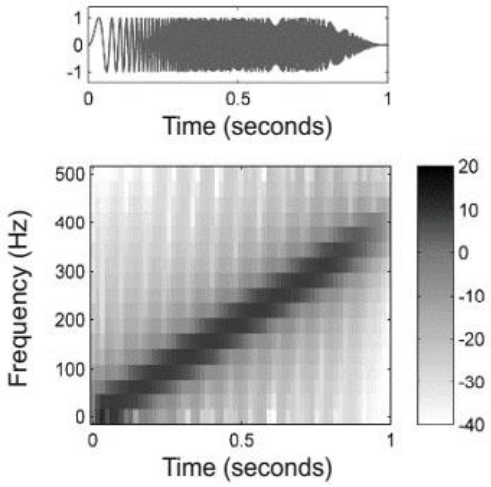


Fig 2. Spectrogram of an audio signal of 1 second duration using a window size of 32 bins and overlap of 16 bins [11].

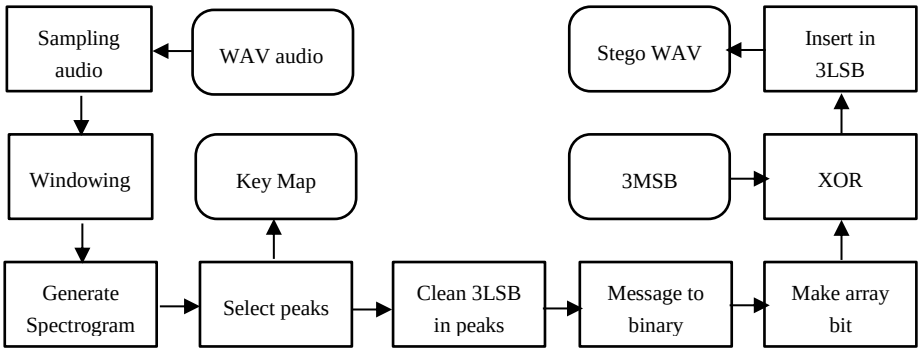


Fig. 3. Steps followed in the insertion stage.

of the nth block with length $K = i_e(n) - i_s(n) + 1$, where i_e and i_s are the final and initial times of the block, k is the frequency of interest and j is the imaginary component. The STFT is defined in Equation (1) [9]:

$$X(k,n) = \sum_{i=i_s(n)}^{i_e(n)} x(i) \exp\left(-jk\left(i - i_s(n)\right)\frac{2\pi}{K}\right). \quad (1)$$

The selection of frequency and initial energy level from which the points of the spectrogram will be chosen to hide the message bits. Starting from the selected frequency, a bit is written as 1 if the point is selected for insertion or a 0 if not, in a key file with which the positions with information are obtained during message recovery.

This is the time to clear the 3 LSBs of the value of selected spectrogram points, that is, the 3 LSB are set to '000'. After clearing, we count the number of bits available to insert information. Then a message that can be hidden in that number of bits is chosen

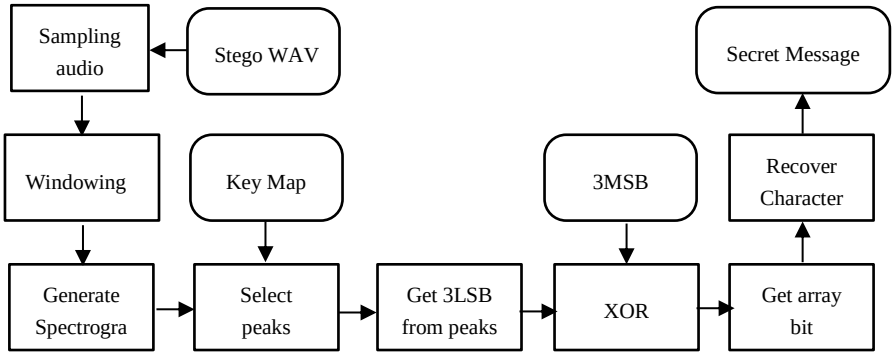


Fig. 4. Process to retrieve secret message from stego audio file.

considering a representation of 8 bits per character, subsequently, every character is converted into 8-bit string.

To reduce the risk of message extraction by unauthorized persons, the message bits are arranged in a vector constructed as follows: we place the first bit of each character, then the second bit of every character, and so on until the least significant bit is reached. For example, considering the string “EL” with ASCII values 69 and 76 for each character respectively, and binary values '01000101' and '01001100', the bit string to be inserted would be '0011000001110010'.

For insertion, the bit string is placed in batches of 3 into the cleaned LSBs, protecting them with an XOR masking with the 3 MSBs of the selected elements of the spectrogram. Finally, the recovery of the modified audio signal is done by applying the inverse STFT and writing it to a WAV file. This procedure can be schematically seen in Figure 3.

3.2 Recovery Stage

The process to retrieve the secret message is similar to the procedure described above: first we read the stego WAV file with a sampling frequency equal to that of the insertion stage, the signal windowing is made with the same parameters as in the insertion stage. Then the spectrogram is obtained using the STFT.

At this point, the key file containing the spectrogram points with information to be extracted is read. For each point, if was selected, we operate the 3 LSBs and the 3 MSBs of the real part using the XOR gate to generate a string. Finally, we extract each character from the string and write it to a text file, considering the order of the character bits from insertion stage. The previous process can be seen in Figure 4.

3.3 Evaluation of Hiding Quality

Traditional metrics are used to compare the original media and the stego media:

– Mean Square Error (MSE): Is the error between the original signal and the stego signal in the form of mean squared error, expressed as an average. Low values indicate insignificant changes in the audio and are given by Equation (2):

$$MSE = \frac{\sum_{i=1}^{M} ||x(i) - y(i)||^2}{M}, \tag{2}$$

where M is the number of points or moments considered in the signal, $x(i)$ is the value that the original signal takes at moment i, and $y(i)$ is the value that the stego signal takes at moment i, both $x(i)$ and $y(i)$ are evaluated in the time domain [7].

– Signal to Noise Ratio (SNR): The ratio between the signal power and the noise power, usually expressed in decibels (dB), is given by Equation (3):

$$SNR = 10 \log_{10} \left(\frac{\sum_{i=1}^{M} x(i)^2}{\sum_{i=1}^{M} [x(i) - y(i)]^2} \right), \tag{3}$$

where $M, x(i)$ and $y(i)$ behave in the same way as in Equation (2) [5].

– Peak Signal to Noise Ratio (PSNR): Used to calculate the quality of steganography, it is a metric that evaluates the distortion of the modified media, measured in dB. Generally, a value greater than 30 indicates that the hidden information will go unnoticed, and it is defined by Equation (4):

$$PSNR = 10 \log_{10} \left(\frac{\max\{x\}}{\sqrt{MSE}} \right), \tag{4}$$

where $max\{x\}$ is the maximum value of signal amplitude values in the time domain and MSE is the Mean Squared Error calculated with Equation (2) [7].

– Structural Similarity Index (SSIM): It is used to evaluate the similarity between the original media and the media with the inserted information. If the result is close to 1, then the altered media maintains good quality and is very similar to the original, and is given by the expression:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{5}$$

where μ_x and μ_y are the mean values of the signal in the time domain of the original media x and the stego media$y$, σ_x and σ_y are the standard deviations of the signals and σ_{xy} is the covariance of the signals, in addition C_1 and C_2 adopt values of 0.01 and 0.03 respectively in order to avoid instability when the mean or standard deviation is close to zero [7].

4 Results

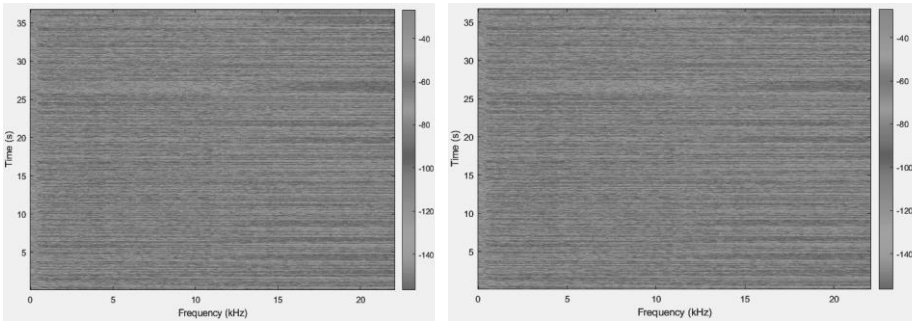
For the tests of the proposed method, 3 audios from the resource list of the Audio Content Analysis website [12] were used, those are "audio_MusicDelta_Britpop_Drum.wav" with a duration of 36 seconds, "audio_pop_excerpt.wav" with 14 seconds, and "audio_speech_excerpt.wav" also with 14 seconds duration. Chat GPT model [13] was asked to generate a random tale in spanish, from which the first necessary characters were extracted, the largest chunk of 141 letters obtained is *"Había una vez un pequeño pueblo ubicado en medio de un*

Table 1. Some parameters selected for insertion, payload capacity and duration.

Cover audio	Minimum Insertion Frequency (kHz)	Minimum Insertion Power (dB/Hz)	Number of Characters	Duration (sec)
audio_MusicDelta_Britpop_Drum	15	-70	33	36
audio_pop_excerpt	10	-70	141	14
audio_speech_excerpt	10	-80	121	14

Table 2. Results of the established metrics.

Cover audio	MSE	SNR (dB)	PSNR (dB)	SSIM
audio_MusicDelta_Britpop_Drum	0.0914	82.1428	94.9246	0.9999
audio_pop_excerpt	0.2824	84.7829	93.0542	0.9998
audio_speech_excerpt	0.1411	76.5768	89.9308	0.9996



(a) (b)

Fig. 5. (a) Spectrogram of the original audio. (b) Spectrogram of the stego audio.

bosque denso y frondoso. El pueblo estaba formado por pequeñas casas de madera, cada u", this text and some subchunks was hidden in audio files.

The audios were sampled at 44.1kHz with a Hamming window of 256 elements and 128 overlaps to ensure signal continuity, and a high frequency and energy level were chosen from which to select the points to hide information in the 3 LSBs, based on this, Table 1 is obtained.

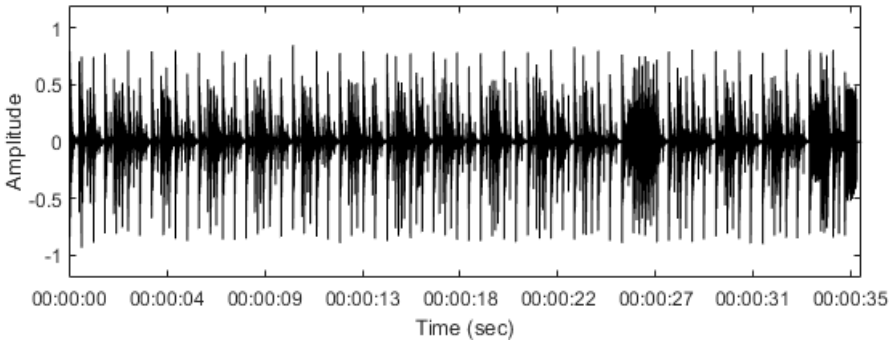
The maximum possible number of bits was hidden according to the method, and the metrics mentioned in section 3 were calculated; the results are reported in Table 2.

The original and stego spectrograms of the file "audio_MusicDelta_Britpop_Drum" are shown below in Figure 5a and 5b, respectively.

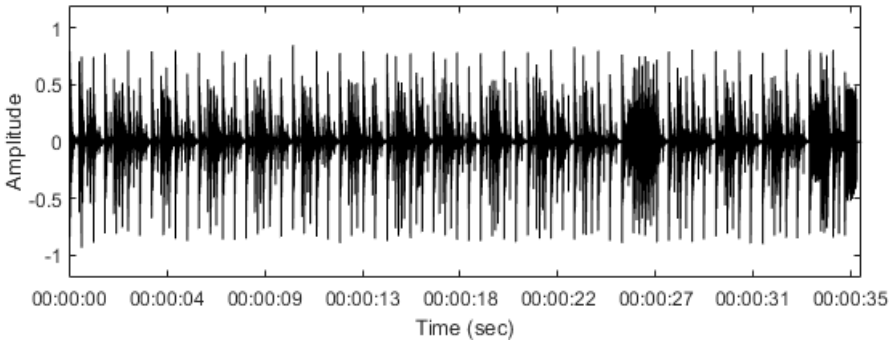
The spectrogram is not usually the common way to represent a signal, the most used way to show them graphically is as waves in the time domain, this representation of the same audio file in Figures 5a and 5b can be seen in Figure 6a and 6b.

Note that both audio files are practically the same when viewed in their full representation, which is to be expected considering the data shown in Table 2, so the difference between the original audio and the stego audio is shown in Figure 7.

A sample of the change can be seen in Figure 8, in (a) the change between the waves is observed in a high-frequency segment where text bits were inserted, while in (b) a low-frequency segment is shown where there is no change.



(a)



(b)

Fig. 6. (a) Original audio signal. (b) Stego signal. The *x* axis represents the index of the bins along the signal.

5 Analysis

The audio "audio_MusicDelta_Britpop_Drum" required an initial frequency of 15kHz, which is higher than the 10kHz of the other two audios. This is because there are higher energy levels at that high frequency, while the other audios had few bins with energy at that frequency, so the range was reduced to 10kHz. Similarly, it was sought to keep the minimum energy level in the region represented in green in the spectrogram. For the 3 audios, this value remains similar. The reason for choosing high energy levels is because these levels, which are the spectral peaks, are particularly useful, as they tend to be resistant to noise, which theoretically helps in preserving the hidden message. Evaluating the resistance to attack or compression of the stego file goes beyond the objective of this document, which is limited to proposing a method of hiding bits in the frequency domain.

Reviewing the data hiding evaluation metrics, it can be seen that the mean squared errors in the 3 files are considerably low, never exceeding 0.3 units with a minimum of 0.0914. The SNR remains of good quality, above 82 and close to each other for the files "audio_MusicDelta_Britpop_Drum" and "audio_pop_excerpt", while it drops a little to

just over 76 for "audio_speech_excerpt". However, it can be considered acceptable. The PSNR is, in general, high, above 89, and the structural similarity is very close to 1 in all cases. Therefore, it can be said that the steganography work has been carried out successfully and preserving the quality of the cover audio, in addition, it was possible to recover 100% of the hidden characters.

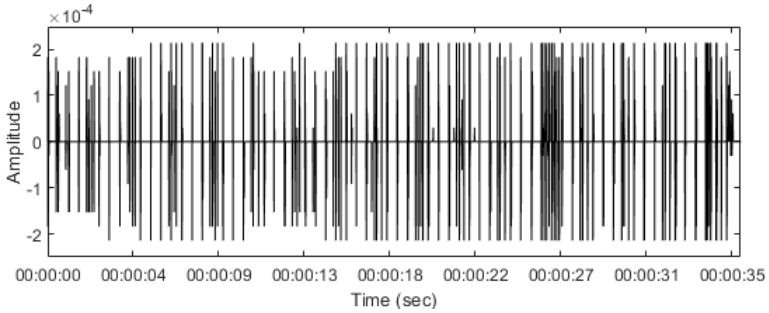


Fig. 7. Audio visible changes in time domain after message insertion.

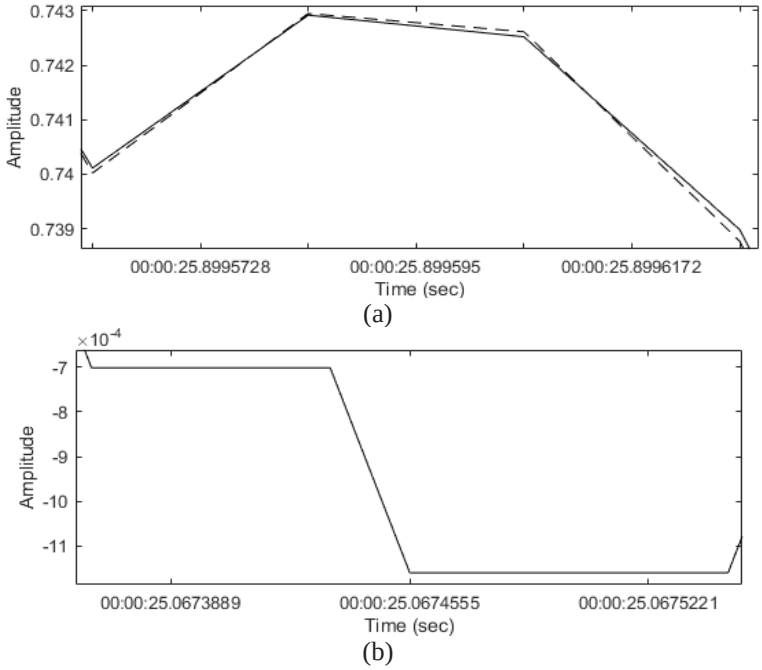


Fig. 8. (a) High frequency segment of the original (continuous line) and stego (dashed line) audios where there is a change. (b) Low frequency segment where there is no insertion.

It is worth noting that relatively small amounts of characters have been hidden. However, this is due to the use of audio files lasting only a few seconds. The use of steganography in longer audio files would allow the transport of much longer texts, supported by a different selection of initial frequencies and energy levels.

Regarding the robustness of the method, the proposed steganography procedure is not robust against lossy compression attacks such as MP3 compression, since a key file is used and due to the large number of parameters that can be used in the compression, it is very difficult to match the right points where information could be stored to recover, in addition to the nature of lossy compression that generates changes in the audio data, also, the alteration of the audio with white noise, due to the a wide range of frequencies that it covers, it alters the audio file making the hidden information impossible to recover, since it even affects high frequencies, which is where the secret message has been hidden.

6 Conclusions

The analysis of the results shows that the proposed method for hiding bits in the frequency domain is effective and manages to preserve the quality of the original audio. The choice of an appropriate initial frequencies and energy levels, as well as the use of resistant spectral peaks, contribute to the effectiveness and likely preservation of the hidden message.

The evaluation metrics, such as MSE, SNR, PSNR, and SSIM, indicate good performance in data hiding for the three audio files analyzed. Despite some variations in the metrics between the files, overall, the results are consistent and satisfactory.

It is worth noting that the main objective of this study was to propose and analyze an audio steganography method in the frequency domain by hiding information in the spectrogram, without addressing attack robustness or compression. Future research could focus on analyzing the robustness of the proposed method against different types of attacks or compressions.

In summary, this study demonstrates that audio steganography in the frequency domain is a viable and effective technique for hiding information in audio files without compromising their quality, opening up new possibilities for information security and communication in digital environments.

References

1. Jian, C., Wen, C., Rahman, N., Hamid, I.: Audio Steganography with Embedded Text. IOP Conference Series, 226, 012084 (2017) doi: 10.1088/1757-899x/226/1/012084.
2. Febryan, A., Purboyo, T. Saputra, R.: Steganography methods on text, audio, image and video: A survey. International Journal of Applied Engineering Research, 12, pp. 10485–10490 (2017)
3. Johri, P., Mishra, A., Das, S. Kumar, A.: Survey on steganography methods (text, image, audio, video, protocol and network steganography). In: 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 2906–2909 (2016)
4. Hussain, M., Rafat, K.: Enhanced Audio LSB Steganography for Secure Communication. International Journal of Advanced Computer Science and Applications, 7(1), pp. 340–347 (2016)
5. Zainab, N., Ban, N.: Image and audio steganography based on indirect LSB. Kuwait Journal of Science, 48(4), pp. 1–12 (2021)

6. Geleta, M., Punti, C., McGuinness, K., Pons, J., Canton, C., Giro-i-Nieto, X.: PixInWav: Residual Steganography for Hiding Pixels in Audio. In: ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2485–2489 (2022)
7. Abood, E., Abdullah, A., Sibahee, M., Abduljabbar, Z., Nyangaresi, V., Kalafy, S. Ghrabta, M.: Audio steganography with enhanced LSB method for securing encrypted text with bit cycling. Bulletin of Electrical Engineering and Informatics, 11(1), pp. 185–194 (2022)
8. Pandey, M., Parmar, G., Patsariya, S.: An Effective Way to Hide the Secret Audio File Using High Frequency Manipulation. In: Communications in computer and information science. Springer Science+Business Media, pp. 125–130 (2011)
9. Lerch, A.: An Introduction to Audio Content Analysis. Wiley (2012)
10. National Instruments: Understanding FFTs and Windowing (2023) https://download.ni.com/evaluation/pxi/Understanding.
11. Müller, M.: Fundamentals of Music Processing. Springer (2015)
12. Lerch, A.: Audio Content Analysis. (2023) https://www.audiocontentanalysis.org/
13. OpenAI: ChatGPT (2023) https://chat.openai.com/.