Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **1**, March **2025**

Compressive Sensing Based Compression Algorithm for the Audio Signal

Electa Alice Jayarani A

Department of Electronics and Communication
K.S. Institute of Technology, Bengaluru, India
electalice@gmail.com

Helen K Joy

Department of Computer Science
Christ University, Bengaluru, India
helenjoy88@gmail.com

Thenmozhi S

Department of Electronics and Communication
Dayananda Sagar College of Engineering, Bengaluru, India
thenmozhi-ece@dayanandasagar.edu

Received November 11, 2024, revised January 25, 2025, accepted February 6, 2025.

Abstract. *Nyquist sampling is used in the conventional digital converter to convert the analog audio signal to digital audio data. Traditional approaches, such as Nyquist sampling, require high sampling rates, leading to large datasets. After sampling, the digital audio data is often compressed using algorithms like Moving Picture Experts Group Layer-3 Audio Coding (MP3), Advanced Audio Coding (AAC), or other codecs to reduce the file size for storage or transmission. Recently, Compressive Sensing (CS) has been used instead of Nyquist Sampling. Instead of sampling at the Nyquist rate, Compressive Sensing samples the signal at a much lower rate. The recovery of the original signal from these fewer samples is possible through minimization techniques. Hence, in the proposed algorithm, we use Compressive Sensing for audio compression. The basis matrix for Compressive Sensing is generated by exploring different transform matrices. The proposed algorithm leverages the principles of Compressive Sensing to enhance audio compression by reducing the number of samples needed and using efficient recovery techniques, resulting in a high compression rate suitable for modern audio applications.*
Keywords: Audio signal, Compressive sensing, Audio Compression, l1 minimization

---

1. **Introduction.** An audio signal is a representation of sound, typically as an electrical voltage or current, that can be transmitted, processed, or recorded. These signals are essential in various applications, including music production, telecommunications, hearing aids, and multimedia systems. Understanding audio signals and their processing is fundamental to a wide array of technologies and applications. Advances in signal processing techniques continue to enhance our ability to manipulate and utilize audio signals effectively, opening new possibilities in communication, entertainment, and beyond. Audio signal compression is critical for efficient storage and transmission, especially in applications such as streaming, telecommunications, and multimedia systems. Traditional compression techniques rely on Nyquist sampling[1], which dictates that the sampling rate must be at least twice the highest frequency present in the signal. This results in a

substantial amount of data, necessitating further compression through codecs like MP3 and AAC[2].

1.1. **Audio Compression Vs Compressive Sensing.** To convert an analog signal to a digital format, it must be sampled at a rate at least twice the highest frequency present in the signal, as stated by the Nyquist-Shannon sampling theorem. This ensures that the signal can be accurately reconstructed from its samples. Figure 1 shows the audio signal acquisition, analog to digital converter which includes Nyquist sampling, encoding, quantization, and Audio Compression. Compressive Sensing (CS) has emerged as a

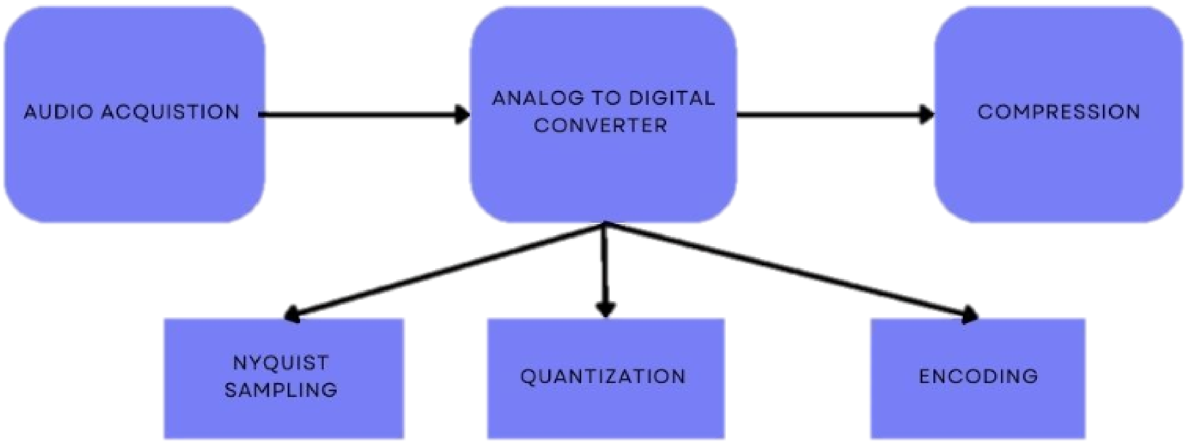


FIGURE 1. Traditional Audio Compression

promising alternative, capable of reconstructing signals from significantly fewer samples[3]. This method capitalizes on the sparsity of signals in a certain domain, allowing for effective compression and accurate reconstruction using optimization techniques such as l1 minimization[4]. Figure 2 shows the audio compression based on compressive sensing.

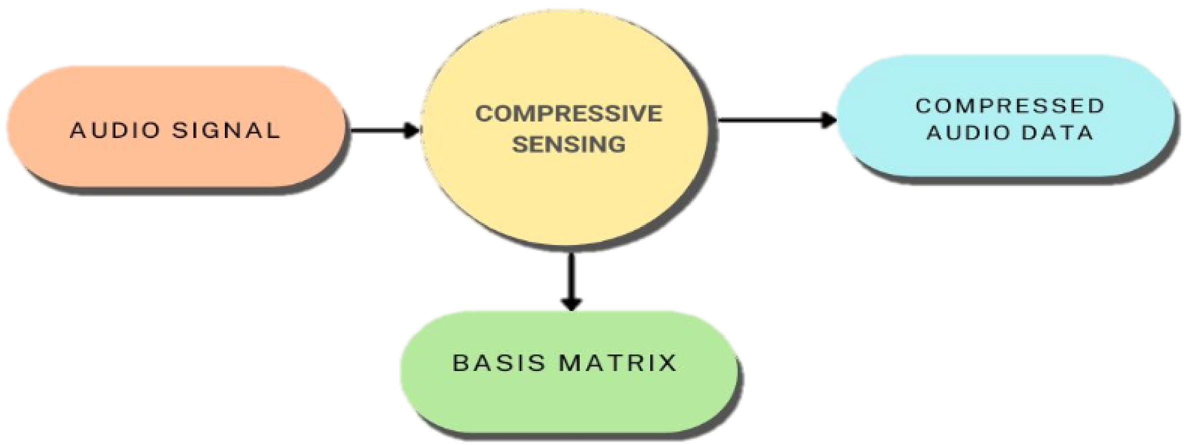


FIGURE 2. Traditional Audio Compression

2. State of Art.

2.1. **Audio Compression algorithms.** Audio compression is a crucial field in signal processing, driven by the increasing demand for efficient storage and transmission of audio data. Over the years, various algorithms have been developed to optimize the trade-off between compression efficiency and signal quality. These algorithms incorporate advanced mathematical models, transformative techniques, and hybrid approaches to address challenges such as maintaining high fidelity, minimizing storage requirements, and ensuring secure communication. The author[5] proposes an audio compression algorithm leveraging compressive sensing. The basis matrix is generated using the Gaussian random numbers. The proposed algorithm uses the Moore-Penrose pseudoinverse to recover the compressed signal. This approach yields correlation coefficients and SSIM values near one and achieves low MSE values around 5.0E-4 while maintaining a PSNR within an acceptable range. The paper [6]proposes a hybrid algorithm for analyzing, compressing, and encrypting speech signals. It compresses the speech by removing low-intensity frequencies while maintaining quality. The compressed signal undergoes a two-level scrambling process: external scrambling rearranges speech segments divided using Fuzzy C-Means, and internal scrambling scatters block values using a Sudoku pattern and quadratic map. Finally, the scrambled speech is encrypted using the Threefish algorithm. The approach is shown to be efficient for both compression and encryption, validated by statistical measures. In [7], the author proposes a statistical method for analyzing four compression algorithms—Huffman coding, LZW, RLE, and LZ77. The algorithms were evaluated using the compression ratio as the comparison parameter. Results showed Huffman and RLE performed best for video and audio files, RLE excelled for images, and LZW outperformed the others for text files. This paper [8] explores audio compression using DWT and RLE techniques in MATLAB, aiming to enhance audio signal clarity and reduce noise at the receiver. The results indicate improvements in compression factor and signal-to-noise ratio (SNR) for all Daubechies wavelet families (DB1, DB6, DB8, DB10) when applying DWT and RLE techniques In [9], the author introduces an alternative method for compressing audio files, utilizing the Discrete Cosine Transform (DCT) and LZW coding techniques. When tested on various uncompressed audio files using the necessary hardware and software, this method produced promising and acceptable results compared to existing audio compression methods. The author [10] introduces a modified audio compression algorithm that utilizes DCT with TAM. The newly developed algorithm achieved a high compression ratio, reducing the output audio signal to one-fourth of its original size. This reduction was facilitated by the recording device's ability to ignore background noise. The results of this research will enhance audio compression systems, particularly for future applications that demand highly compressed audio signals while maintaining high quality. This paper [11] proposes an algorithm based on compressed sensing (CS) for simultaneously compressing and encrypting audio signals. In this method, audio signals are segmented into frames of 1000 samples, which are then transformed into sparse frames using the Discrete Cosine Transform (DCT). Each frame is compressed using a unique sensing matrix, generated by an M-sequence generator and a chaotic mixing scheme, ensuring that the proposed system meets the Enhanced Whitening Security (EWS) criterion. Evaluation results across various classes of audio signals demonstrate that the proposed scheme enables the compressed and secure communication of different types of audio signals. Summary of the literature review is given in table 1.

This review synthesizes key contributions in the field, highlighting their methodologies, strengths, and performance metrics. It provides a comprehensive perspective on the evolution and potential applications of audio compression algorithms. The paper [12] presents a modified vector quantization algorithm for scalable audio coding with multiple layers of audio fidelity. The algorithm enhances the quantization and encoding stages,

TABLE 1. Literature Review

Reference	Methodology	SNR	CR
[5]	CS with Gaussian Random number	20.9	1.81
[6]	Compressing and encryption	12.299	2.0
[8]	DWT and RLE	28.80	1.11097
[9]	DCT and LZW	57.95	4.28
[10]	DCT with TAM	54.31	1.20
[11]	CS with DCT	-	7.81

utilizing psychoacoustic and arithmetic techniques to optimize data handling. It generates a scalable bitstream effective at both low and high bit rates. Subjective evaluations using the MUSHRA test showed improved performance compared to previous algorithms, with mean normalized scores recorded across various bit rates. In [13], the author concludes, the Wavelet compression as the best-performing algorithm among the other compression algorithm such as MP3, LPC, Wavelet, and Sub band algorithms, delivering outstanding accuracy, perceptual quality, and minimal audio distortion. The traditional compression algorithm such as MP3, has some limitation also. The study [14] examined how MP3 compression and reduced sampling rates impact avian bioacoustics applications. MP3 compression lowered detection rates, reduced individual abundance counts, and decreased precision and recall for automated recognition. Sampling rate reduction caused systematic biases in acoustic indices and affected recognizer performance. The authors advise against MP3 compression, recommending lossless compression methods to preserve recording accuracy. Compressive sensing can also be used to protect the audio signal. In [15], the author proposes a method to detect music plagiarism using melodic methods: a supervised learning algorithm, Edit distance, and N-grams. The analysis indicates that the Optimized Edit Distance with the Bipartite Graph Matching approach achieves an accuracy of 87.5% with a smaller dataset. In [16], the author proposes a crypto-watermarking algorithm based on Compressive sensing to protect the audio signals. The proposed algorithm generates the watermark using the HAAR transform. The algorithm is tested on various audio signals, achieving an SNR greater than 30dB, and demonstrates good robustness against attacks such as echo addition, noise addition, and reverberation. Generally, audio compression is implemented on the digitally converted audio data. As the Nyquist sampling is used, the number of data will be more, which increases the algorithm's complexity and the compression time. Our research addresses this issue by proposing an algorithm to compress the audio signal during the analog-to-digital conversion. The compressive sensing is used in the proposed algorithm. By using compressive sensing instead of Nyquist sampling, fewer samples will be acquired, resulting in a higher compression ratio.

2.2. **Compressive Sensing.** To convert the audio signal into a digital signal, the first step involves is acquiring the signal at specific intervals, a process known as sampling. Let's denote the audio signal as A(t), and the traditional sampled data as shown in equation (1),

A(nT) for integer values of n. (1)

Where n is the integer value T=1fs is the sampling rate. To accurately reconstruct the host signal from the sampled data, the sampling process must adhere to the Nyquist rate. In recent days, signal acquisition methods have undergone a paradigm shift, increasingly focusing on sparse coding or compressed sensing techniques [17][18]. Unlike traditional analog-to-digital conversion, which relies on digital filtering based on Shannon's uniform sampling principle, this non-uniform sampling approach generates fewer samples and allows signal recovery through mathematical convex programming rather than interpolation or decimation filters [19]. A critical aspect of Compressive Sensing is the selection of an appropriate basis matrix [20]. This research explores various transform matrices, including Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Haar transform, to identify the optimal basis for audio signal compression. Compressive Sensing exploits the sparsity of signals in a transform domain, enabling the reconstruction of signals from a small number of linear measurements. The audio signal is first transformed into a sparse domain using the selected basis matrix. Sparse sampling is then performed, capturing fewer measurements than traditional methods. The recovery process employs l1 minimization to reconstruct the original signal from these sparse samples. This contrasts with the Nyquist-Shannon sampling theorem, which requires dense sampling to ensure signal recovery.

2.3. **Recovery algorithm.** Sparse recovery involves reconstructing a sparse signal from a limited number of measurements, and one of the common approaches to achieve this is by solving an optimization problem. The most popular method for sparse recovery is l1 minimization, also known as Basis Pursuit. This approach seeks to minimize the l1 -norm of the signal subject to a set of linear constraints. The l1-minimization algorithm is particularly effective because it promotes sparsity in the solution. The l1-minimization is widely used in compressed sensing, image processing, machine learning, and other areas where sparse signal recovery is critical.

3. **Proposed Method.**

3.1. **Algorithm.** The proposed algorithm aims to reduce the sampled data count at the initial stage, rather than during the compression process. The original audio signal is converted into the sparse audio signal using equation (2):

$$\mathbf{a} = \mathbf{B}\mathbf{A} \tag{1}$$

Where:

- $\mathbf{A} \in \mathbb{R}^{n\times 1}$ is the original audio signal
- $\mathbf{B} \in \mathbb{R}^{n\times 1}$ is the transform matrix
- $\mathbf{a} \in \mathbb{R}^{n\times 1}$ is the sparse signal

The compression of the sparse audio signal is given by equation (3):

$$\mathbf{A_{comp}} = \mathbf{\Psi}\mathbf{a} \tag{2}$$

Where:

- $\mathbf{a} \in \mathbb{R}^{n\times 1}$ is the sparse signal
- $\mathbf{\Psi} \in \mathbb{R}^{m\times n}$ is the sensing matrix
- $\mathbf{A_{comp}} \in \mathbb{R}^{m\times 1}$ is the compressed audio signal

Sparse recovery involves finding the sparsest solution to an underdetermined system of equations. Ideally, we'd solve this using ℓ_0-minimization as given in equation (4):

$$\|\mathbf{a}\|_0 \quad \text{subject to} \quad \mathbf{A_{comp}} = \mathbf{\Psi}\mathbf{a} \tag{3}$$

where $\|\mathbf{a}\|_0$ counts the number of non-zero elements in $\mathbf{a}$.

However, ℓ_0-minimization is a non-convex, NP-hard problem, making it computationally intractable for large-scale problems. ℓ_1-minimization is used as a convex relaxation of ℓ_0-minimization. Under the condition of the Restricted Isometry Property of the sensing matrix, the solution to the ℓ_1-minimization problem is exactly the same as the solution to the ℓ_0 problem. Therefore, the ℓ_1-minimization problem[21][22] can be framed as a convex optimization problem, specifically a linear program (LP), given by equation (5):

$$\min \sum_{i=1}^{n} |x_i| \quad \text{subject to} \quad \mathbf{\Psi a} = \mathbf{A_{comp}} \tag{4}$$

3.2. **Methodology.** The proposed methodology for compressing and recovering audio signals using compressive sensing is illustrated in the block diagram shown in Figure 3. It consists of the following steps:

1. The audio signal in the time domain is considered as the primary signal.
2. The primary signal is transformed into a sparse domain using a predefined transform matrix, such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), or Haar transform. This step ensures that the signal exhibits sparsity, a critical requirement for compressive sensing.
3. Using the Haar matrix, the orthogonal basis matrix or sensing matrix is generated, which will be used to perform the compressive sensing.

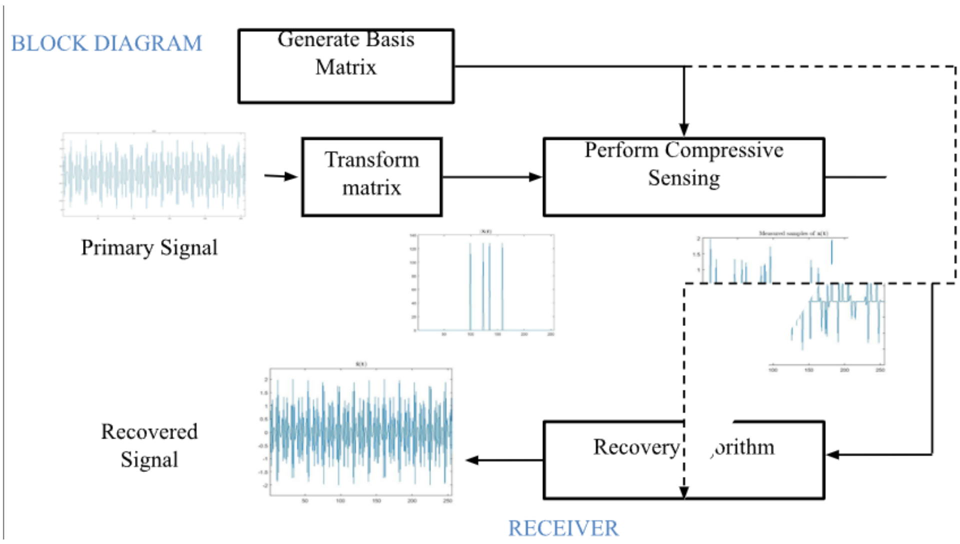


FIGURE 3. Proposed block diagram

4. Compressive sensing is applied to the sparse signal using a sensing matrix, which reduces the dimensionality of the data by capturing only a subset of its components. This step significantly reduces the number of samples required, enabling efficient compression.
5. The compressed signal is stored or transmitted to the receiver for reconstruction.
6. At the receiver end, a recovery algorithm is employed to reconstruct the original signal from the compressed data. The reconstruction process often involves solving an optimization problem, such as using the Moore-Penrose pseudoinverse or other iterative algorithms.
7. The recovered signal is obtained as the output, closely resembling the original primary signal. The quality of reconstruction is evaluated using metrics such as Signal-to-Noise Ratio (SNR) and Compression Ratio.

This methodology leverages the principles of compressive sensing to achieve efficient audio compression while ensuring accurate signal reconstruction at the receiver.

3.3. **Performance Evaluation.** The performance evaluation of the proposed algorithm is measured using the objective measurement parameter, Signal-to-Noise Ratio (SNR) and, it is given in equation (6).

$$\text{SNR} = 10\frac{\sum_{i=1}^{N} A_p^2(i)}{\sum_{i=1}^{N}(A_p(i) - A_r(i))^2}dB \tag{5}$$

Where:

- $A_{p,i}$ is the audio signal before compression
- $A_{r,i}$ is the decompressed audio signal

where Ap is the audio signal before the compression and Ar is the decompressed audio signal. The next performance evaluation is the compression ratio. The compression ratio of an audio file is a measure of how much the original file size has been reduced after compression. It is typically expressed as a ratio or a percentage and can be calculated using the formula given in equation (7):

$$\text{Compression ratio} = \frac{\text{Primary Audio file size}}{\text{Compressed Audio file size}} \tag{6}$$

The compression ratio is closely related to the bitrate of the compressed audio file. Lower bitrates generally result in higher compression ratios but may reduce audio quality.

4. Results And Discussion.

4.1. **Experimental Setup.** The proposed algorithm was implemented using MATLAB 2016 on an Intel Core i5 processor to achieve the research objectives. A set of 5 primary audio clips was selected for the experiment. All audio clips are mono-channel, with a duration of less than 60 seconds, sampled at 44.1 kHz, and have an audio data width of 8 bits. The selected audio clips include solo musical instruments (such as violin, guitar, piano, and flute), songs (such as bass, Handel, track, Mary Song, Backstreet Boys song, Emilie Big World), and various frequency clips. Table 2 provides details of the selected audio clips.

Audio	Length	Duration	Sampling Frequency (Hz)
Handel	73113	8s	8192
Mary	319725	7s	44100
Backstreet Boys	1323000	30s	44100
Emilie Big World	1323000	30s	44100
Irish Whistle	1323000	30s	44100

Table 2. Test audio files

The selected audio clips are considered as the primary signal for the compression and are represented as $A \in \mathbb{R}^{n\times 1}$. Table 3 shows the different primary audio signals in the time domain.

Most of the audio signals are sparse when converted to the transform domain. The transform matrix used is DFT, DCT, and Haar wavelet transform matrix and is represented as B. In the proposed algorithm, different transform matrix is tested to find the optimum sparse domain of the audio signal. The sparse transformation is given in equation (1), a=B A. Figure 4 shows the primary audio signal and the Fast Fourier Transform signal. It can be seen from Figure 4 that the transform signal is sparser, therefore, compressive sensing can be performed on the audio signal.

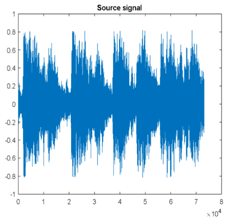
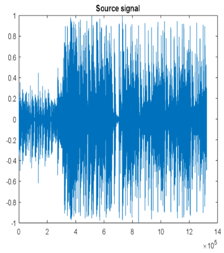
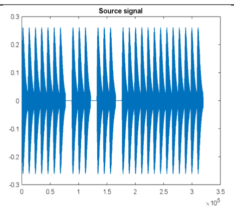
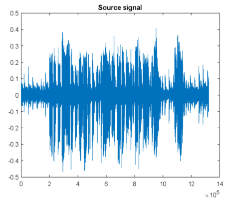
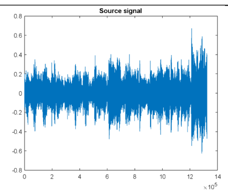
Audio	Audio clips	Audio	Audio clips
Handel		Backstreet Boys	
Mary		Emilie Big World	
Irish Whistle			

TABLE 3. Experimental Audio clips

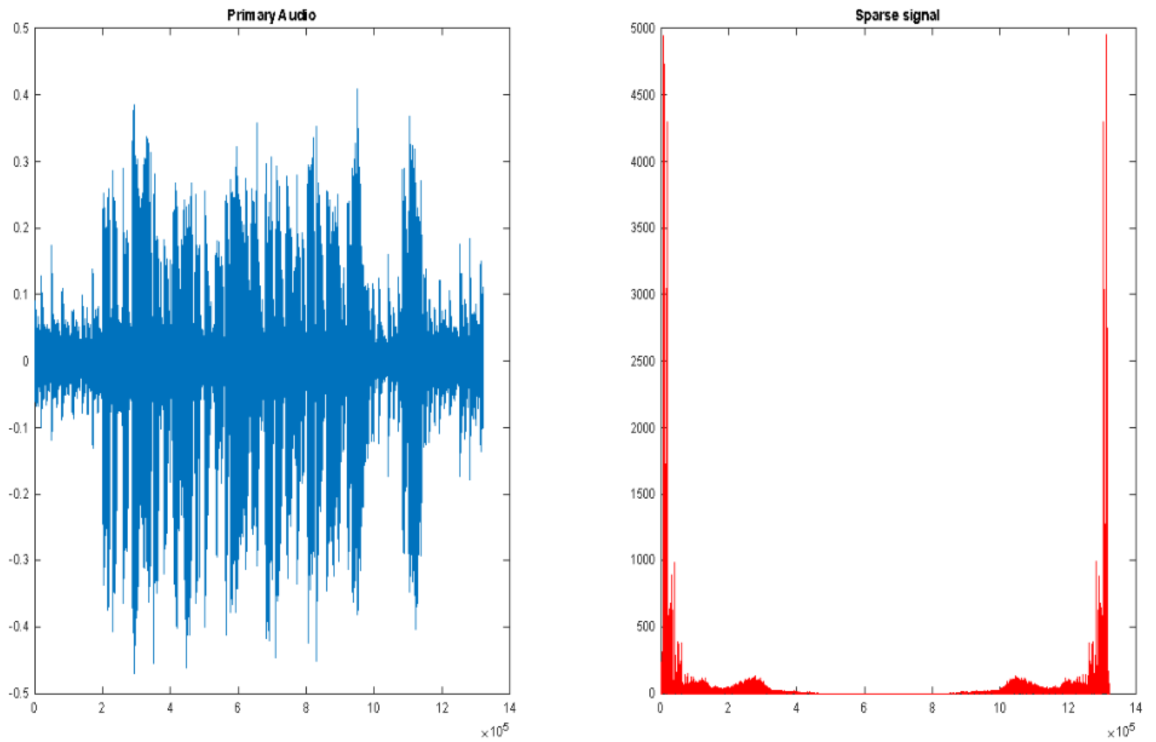


FIGURE 4. Primary and Sparse audio signal

Consider the transform signal is K-sparse. The next criteria to consider is the length of the measurement m. The condition for the recovery of all K-sparse signals [4, 14] for any recovery algorithm is given in equation (8):

$$m \geq CK \ln \frac{N}{K} \tag{7}$$

Where $C > 0$ is the universal constant and is independent of K, m, and N. It can be observed from equation (8) that m is directly proportional to K. Therefore, if the sparsity is low, the measurement m is also chosen to be small relative to N, ensuring that the solution to the underdetermined system of linear equations remains reasonable.

After the selection of suitable $m \ll N$ and K, the sensing or basis matrix $\psi \in \mathbb{R}^{m \times n}$ is generated using the Haar matrix. The Haar matrix is used to generate the basis matrix because of its orthogonal nature. As the sparse audio signal $a \in \mathbb{R}^{n \times 1}$ and the basis matrix $\psi \in \mathbb{R}^{m \times n}$ are generated, the compression of the primary audio signal is performed using compressive sensing. The compressed audio signal is represented as $A_{comp} \in \mathbb{R}^{m \times 1}$. The compressed signal can be stored with less space or can be transmitted. At the receiver, using ℓ_1 minimization, as given in equation (5), the primary signal can be recovered without disturbing the audio quality.

After the selection of suitable $m \ll N$ and K, the sensing or basis matrix $\psi \in \mathbb{R}^{m \times n}$ is generated using the Haar matrix. The Haar matrix is used to generate the basis matrix because of its orthogonal nature. As the sparse audio signal $a \in \mathbb{R}^{n \times 1}$ and the basis matrix $\psi \in \mathbb{R}^{m \times n}$ are generated, the compression of the primary audio signal is performed using compressive sensing. The compressed audio signal is represented as $A_{comp} \in \mathbb{R}^{m \times 1}$. The compressed signal can be stored with less space or can be transmitted. At the receiver, using ℓ_1 minimization, as given in equation (5), the primary signal can be recovered without disturbing the audio quality.

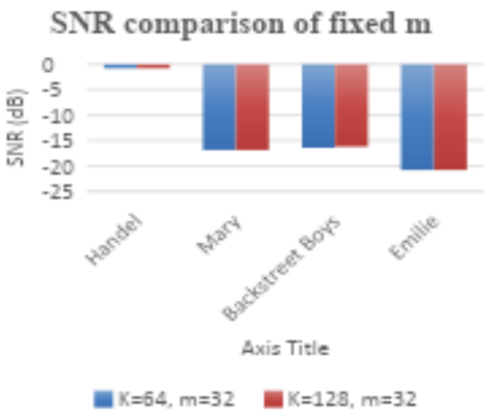
4.2. **Results.** The proposed algorithm is implemented on the selected audio clips. The selected audio signal serves as the primary signal and is divided into multiple frames for processing. Various frame lengths were tested for the proposed algorithm, but as the results showed minimal variation, a frame length of 256 is chosen and discussed in this paper. As discussed in the previous section, different transform matrices, such as DFT, DCT, and Haar, are applied to the primary signal to convert it into a sparse signal. The Discrete Fourier Transform (DFT) matrix achieved the best SNR performance. Assuming the audio signal is K-sparse in the frequency domain, various K values were tested, and the results are discussed in this section.

The next step is to determine m, the measurement of the compressive sensing matrix. The proposed algorithm is implemented by varying both K and m. Table 4 shows the SNR of the fixed $m = 32$ and variable K. At the receiver, for each compressed segment, the primary signal is recovered using the ℓ_1 optimization algorithm. The measurement parameter used is SNR and is given in equation (6). Finally, the signal in the frequency domain is converted back to the time domain by applying the inverse transform.

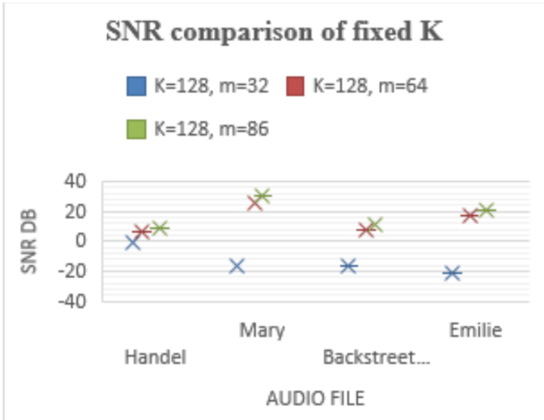
Measurement	**Handel**	**Mary**	**Backstreet Boys**	**Emilie**
K=64, m=32	-0.7618	-16.86	-16.43	-20.79
K=128, m=32	-0.7618	-16.86	-16.2	-20.79

TABLE 4. SNR comparison with fixed m.

From Table 4, it is observed that for $m < 32$, the SNR is less than zero. Figure 5a shows the SNR comparison of fixed $m = 32$ and varying $K = 64$ to 128. The SNR can be improved by increasing m from 32 to 64 and 86. Table 5 shows the SNR of the fixed K and variable m.



(A) SNR comparison with fixed m



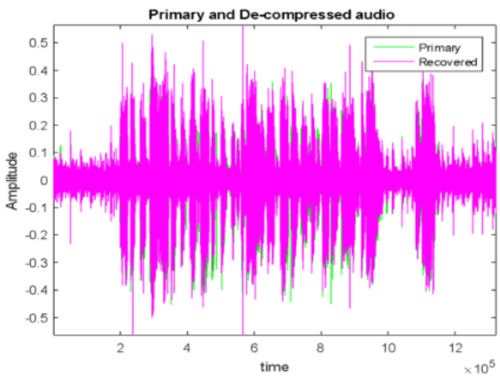
(B) SNR comparison with fixed K

FIGURE 5. (a) SNR comparison with fixed m (b) SNR comparison with fixed K

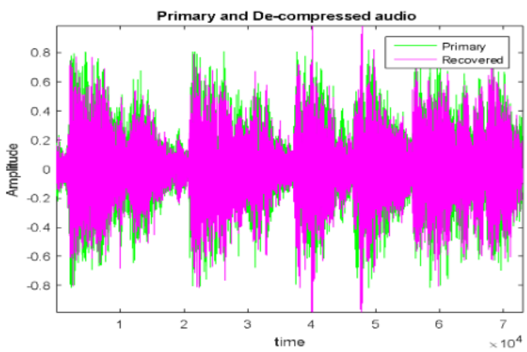
Measurement	Handel	Mary	Backstreet Boys	Emilie
K=128, m=32	-0.7618	-16.86	-16.2	-20.79
K=128, m=64	6.0461	25.52	8.0134	17.456
K=128, m=86	8.7678	30.59	10.55	21.2504

TABLE 5. SNR comparison with fixed K.

Figure 5b shows the SNR comparison of fixed $K = 128$ and varying $m = 64$ to 86. It is observed from the table and the graph that the SNR is improved by increasing the measurement parameter m. The proposed algorithm is tested with various sparse parameters K varying from $K = 32$ to 128 and compressive sensing measurement value m varying from $m = 32$ to 86 for a frame length of 256. The best results are discussed in this article. The primary audio signal and the decompressed signal are shown in Figure 6.



(A) Primary and Decompressed Emilie Audio



(B) Primary and Decompressed Handel Audio

FIGURE 6. Primary and Decompressed Audio signals

The table 6 shows the compression ratio and SNR for the K=128. The table compares the compression ratio and Signal-to-Noise Ratio (SNR) for four different audio samples (Handel, Mary, Backstreet Boys, Emilie Big World) at two different values of M (64 and 86). For all audio samples, the compression ratio is 4:1 when M=64. The compression ratio decreases to 3:1 when M=86.

Audio	**Compression Ratio**	**SNR (M=64)**	**SNR (M=86)**	
Handel	4:1	3:1	6.0461	8.7678
Mary	4:1	3:1	25.5222	30.5951
Backstreet Boys	4:1	3:1	8.0134	10.5518
Emilie Big World	4:1	3:1	17.4562	21.2504

Table 6. Compression Ratio and SNR

In summary, increasing M from 64 to 86 reduces the compression ratio but results in a higher SNR for all audio samples. The proposed algorithm is implemented on different transform matrices such as DFT, DCT, and Haar transform. The table 7 shows the SNR comparison of all the transform matrices. The comparison is made with the assumption of a compression ratio of 4:1.

Audio	**DFT**	**DCT**	**Haar**
Handel	6.0461	6.7139	-1.4513
Mary	25.52	19.17	-47.5822
Backstreet Boys	8.0134	6.8539	-15.8943
Emilie Big World	17.456	12.5379	-27.1608

Table 7. SNR of different transforms

The table compares the performance of three different transforms—Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Haar Transform on the four audio clips. From the table, we can observe, that the DFT generally performs the best among the three transforms, whereas, the DCT performs well but slightly worse than DFT. The Haar Transform performs poorly across all audio files, with particularly low values for Mary and Emilie Big World. Hence, we can conclude, that the best compression ratio with the acceptable SNR is achieved using the DFT transform matrix when the audio signal is K=128 and the measurement parameter of the sensing matrix m=64.

5. **Comparison And Conclusion.**

5.1. **Comparison.** In recent days, many compression algorithms have been developed to compress audio signals. All the existing algorithms perform compression on the digitally converted data of the audio signal. In the proposed algorithm, compression is performed during the analog-to-digital conversion. The proposed compression algorithm uses compressive sensing rather than traditional Nyquist sampling for digital conversion. As compressive sensing results in fewer samples, the compression ratio is increased. The proposed algorithm is evaluated using parameters such as SNR and Compression Ratio. The obtained results are compared with those of existing algorithms. The comparison is shown in the table 8.

Signal to Noise Ratio measures the clarity or quality of the signal. Higher values indicate better quality. Compression Ratio represents the percentage of data compression, with higher values indicating more compression. The proposed algorithm achieves a good

Reference	Methodology	SNR	CR
[5]	CS with Gaussian Random number	20.9	1.81
[6]	Compressing and encryption	12.299	2.0
[8]	DWT and RLE	28.80	1.11097
[9]	DCT and LZW	57.95	4.28
[10]	DCT with TAM	54.31	1.20
Proposed algorithm	Compressive sensing with DFT and Haar based basis matrix	30.59	4.02

Table 8. Comparison with the existing algorithms

balance between SNR and CR, offering better compression performance (CR = 4.02) than most existing methods, while maintaining a competitive SNR (30.59 dB). The [9] results show significant better SNR and CR. But the complexity of the system is more, as the analog audio signal is converted to digital signal and then the compression is performed. Whereas in the proposed algorithm the compression is done during the analog to digital conversion only, hence it reduces the system complexity. Therefore, the proposed algorithm is particularly suitable for applications where a moderate-to-high compression ratio is desired without a significant loss in signal quality. In conclusion, the proposed algorithm effectively bridges the gap between compression efficiency and signal fidelity, making it a competitive alternative to existing techniques.

5.2. **Conclusion.** In conclusion, the proposed algorithm effectively harnesses Compressive Sensing to achieve significant audio compression by reducing the required sampling rate and employing efficient signal recovery methods. This approach not only minimizes the data size but also maintains the quality of the audio signal, offering a promising alternative to traditional Nyquist sampling methods for modern audio applications. The proposed algorithm offers an effective balance between SNR and CR, achieving significant compression while preserving acceptable signal quality. The proposed algorithm was tested using various transform matrices, including DFT, DCT, and Haar transforms, with the best performance achieved using the DFT transform matrix. In future, the research can be carried out to perform the hardware implementation of the proposed algorithm.

REFERENCES

[1] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Pearson Education, 2010.

[2] I. M. Pu, *Fundamental Data Compression*, Butterworth-Heinemann, 2005, 246 pages, ISBN 978-0-7506-6310-6.

[3] I. Selenick, "Introduction to sparsity in signal processing", NYU-poly, Nov. 2012.

[4] E. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements", *Communication on Pure and Applied Mathematics*, pp. 1207-1223.

[5] E. W. Abood et al., "Provably Secure and Efficient Audio Compression Based on Compressive Sensing", *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 335-346, Feb. 2023.

[6] I. Q. Abduljaleel and A. H. Khaleel, "Speech Signal Compression and Encryption Based on Sudoku, Fuzzy C-Means, and Threefish Cipher", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 5049-5059, Dec. 2021. ISSN: 2088-8708.

[7] A. A. R. Beserra, L. C. Souza, and D. F. L. Souza, "Bootstrap Analysis of Compression Algorithms," *IEEE Latin America Transactions*, vol. 18, no. 9, pp. 1639-1645, Sept. 2020.

[8] A. Tsegaye and G. Tariku, "Audio Compression Using DWT and RLE Techniques", *American Journal of Electrical and Electronic Engineering*, vol. 7, pp. 14-17, 2019. doi: 10.12691/ajeee-7-1-3.

[9] S. Shukla, M. Ahirwar, R. Gupta, S. Jain, and D. S. Rajput, "Audio Compression Algorithm Using Discrete Cosine Transform (DCT) and Lempel-Ziv-Welch (LZW) Encoding Method," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, pp. 476-480, 2019.

[10] A. O. Salau, I. Oluwafemi, K. F. Faleye, and S. Jain, "Audio Compression Using a Modified Discrete Cosine Transform with Temporal Auditory Masking," *2019 International Conference on Signal Processing and Communication (ICSC)*, NOIDA, India, pp. 135-142, 2019.

[11] R. Moreno-Alvarado, E. Rivera-Jaramillo, M. Nakano, and H. Perez-Meana, "Joint Encryption and Compression of Audio Based on Compressive Sensing," *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Hungary, pp. 58-61, 2019.

[12] S. Prince, D. Bini, A. A. Kirubaraj, S. J. Immanuel, and M. Surya, "Audio Compression using a Modified Vector Quantization Algorithm for Mastering Applications," *International Journal of Electronics and Telecommunications*, vol. 69, no. 2, pp. 287-292, 2023.

[13] U. Ijaz, F. Gillani, A. Iqbal, M. S. Sharif, M. F. Anwar, and A. Ijaz, "Fine-Tuning Audio Compression: Algorithmic Implementation and Performance Metrics", *IJIST*, vol. 6, no. 1, pp. 220-236, March 2024.

[14] A. G. MacPhail, D. A. Yip, E. C. Knight, R. Hedley, M. Knaggs, J. Shonfield, and E. M. Bayne, "Audio Data Compression Affects Acoustic Indices and Reduces Detections of Birds by Human Listening and Automated Recognisers", *Bioacoustics*, vol. 33, no. 1, pp. 74-90, 2023.

[15] P. Nguyen, H. Le, V. Bui, T. Chau, V. Tran, and T. Bui, "Detecting Music Plagiarism Based on Melodic Analysis", *Journal of Information Hiding and Multimedia Signal Processing*, vol. 14, no. 2, pp. 52-62, June 2023.

[16] E. A. Appadurai, M. R. Bhatt, and G. D. D., "Semi Fragile Audio Crypto-Watermarking Based on Sparse Sampling with Partially Decomposed Haar Matrix Structure", *Acta Cybernetica*, vol. 24, no. 4, pp. 679-697, Nov. 2020.

[17] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, 2013.

[18] I. Ahmed, A. Khalil, and J. Frnda, "Sparse Signal Representation, Sampling, and Recovery in Compressive Sensing Frameworks," *IEEE Access*, vol. 10, pp. 85002-85018, 2022.

[19] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: Analog to Digital at Sub-Nyquist Rates", *IET Circuits, Devices and Systems*, pp. 8-20, 2010.

[20] A. Electa Alice Jayarani, M. R. Bhatt, and D. D. Geetha, "An Introduction to Sparse Sampling on Audio Signal by Exploring Different Basis Matrices," in *Advances in Artificial Intelligence and Data Engineering*, Springer, 2021.

[21] J. Zhang, Y. Li, Z. L. Yu, Z. Gu, Y. Cheng, and H. Gong, "Deep Unfolding with Weighted ℓ_2 Minimization for Compressive Sensing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 3027-3041, Feb. 2021.

[22] J. Hu, Z. Xie, and H. Li, "A L1-Minimization for Image Reconstruction from Compressed Sensing", *Sensors and Transducers*, 2014.