https://doi.org/10.31449/inf.v48i4.5496 Informatica **48** (2024) 641–652 **641**

Wavelet Decompositions, Hierarchical Encoding and Convolutional Neural Network Integrated Lossless Audio Codec

Asish Debnath, Uttam Kr. Mondal
Dept. of Computer Science, Vidyasagar University, Midnapore, 721102, West Bengal, India
E-mail: debnathasish@gmail.com, uttam_ku_82@yahoo.co.in

Keywords: CNN, MSE, MAE, lossless audio compression, approximation coefficients, detail coefficients, hierarchical encoding

Received: November 30, 2023

In this paper, a lossless audio codec is proposed by leveraging Wavelet transformation, Hierarchical encoding with Convolutional Neural Network architecture. In the first phase, three level 1D wavelet decomposition is applied on the input audio for generating approximation and detail coefficients. In the next phase, the approximation and detail coefficients are transformed into binary streams by utilizing the proposed dynamic hierarchical encoding algorithm. In this encoding technique, coefficients are converted to binary by dynamically accumulating the binary path values. In the subsequent phase, the binary stream is transformed into image patterns and further compressed by reducing the dimensionality by the proposed convolutional neural network(CNN) model. The model's effectiveness is evaluated against current conventional lossless audio benchmarks and machine learning-based methods. Experiment results demonstrate that the method shows better performance than existing lossless audio techniques.

Povzetek: Razvit je avdio kodek, ki združuje valovne transformacije, hierarhično kodiranje in konvolucijske nevronske mreže za izboljšanje kompresije.

1 Introduction

In today's world, immense amount of audio data is being generated at every moment. Therefore, using the network bandwidth and storage space efficiently, audio data compression is one of the paramount important. The advent of deep neural network opened the possibilities of achieving excellent result along with the conventional techniques in this area. Lossless audio compression [1] is the audio compression technique utilized whenever the requirement is to preserve the quality of the original input audio and reconstructed audio signal. It also reduces the file size without losing the audio information. On the contrary, lossy audio compression losses some audio data permanently to achieve higher compression. Lossless audio data [2] compression is used where data loss is not expected at all. A graph based [3] and cluster quantization [4] based audio encoding techniques introduced recently. Deep learning [5] based approaches are applied recently in audio compression. Lossless compression is required in medical industies for compression and sending various bio-signals [6].
In this present work, we proposed an lossless audio codec (WLCLAC) by sequentially integrating three layer 1D wavelet [7] decomposition, adaptive hierarchical binary encoding, and CNN [8] compression architecture. The proposed model works in three stages. In the first step, the input audio sampled values are transformed using a three-level one-dimensional discrete wavelet decomposition approach. Using wavelet transformations, detail coefficients(cD) and approximation coefficients(cA) are generated at each level of decomposition. For the signal S, the structure of the wavelet decomposition at i level is as follows: $[cA_i, cD_i, .., cD_1]$ For i = 3, this structure comprises the terminal nodes of the following tree shown in figure 1.
In the second step, hierarchical binary encoding technique is applied on the input wavelet coefficients. Coefficients are segregated into integers first, then search in the proposed hierarchy. If the node is matched with the input digit, then binary path values are accumulated and translated as an encoded binary stream of the input digit.
In the third step, CNN encoder decoder model is used to further compress the binary stream.

Figure 2 depicts the encoding and decoding model. Utilizing the encoder, size of the input audio compressed in the encoding step. Compressed signal stored as the latent space with reduced and compressed form. Decoder section reconstructs the signal which is very much similar to the input.

The proposed technique has significant potential for real-time applications in the field of audio compression. Upon deployment, it can enhance the compression ratio and processing speed while reducing the model's complexity. This improvement would lead to reduced memory usage and lower network bandwidth requirements, making the system highly efficient for real-time audio transmission and storage in various applications, such as streaming services, communication systems, and data storage solutions.

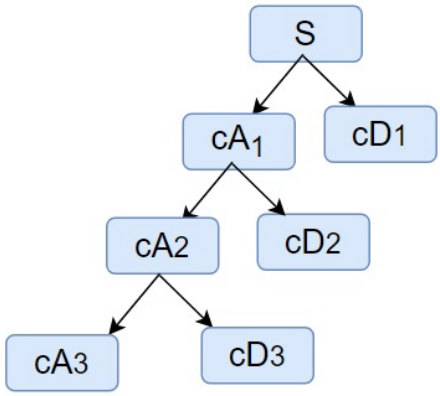


Figure 1: Three level 1D wavelet decomposition tree

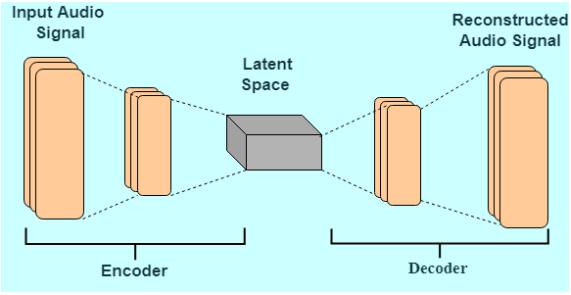


Figure 2: Encoder and decoder framework

1.1 Notations and symbols

The terminology and their complete explanations that are relevant to the abbreviations used in this work are provided in Table 1.

Table 1: Symbol and abbreviation form of some terms

Symbol / Abbreviation	Meaning / Full form
CNN	Convolutional Neural Network
DNN	Deep neural networks
PNSR	Peak signal-to-noise ratio
NCC	Normalised cross correlation
cA	Approximation coefficients
cD	Detail coefficients

2 Literature survey

In 2003, lossless audio codec standard MPEG-4(ALS) [9] was introduced. This LPC (linear prediction encoding) based technique improves the residual coding and reduce the bit rate compared to PCM like approaches. The predictor coefficient cost during decoding and demultiplexing is this codec's drawback. A new variant of this technique, MPEG-4 ALS (RLS-LMS) introduced later by combining LPC model with RLS-LMS. RLS-LMS predictors are used in place of the LPC model in this model, which eliminates the predictor coefficient from the coded stream. It has rapid speed of decoding. But, Instability in numbers accompanied by a white or lightly variable signal. MPEG-4 SLS model introduced in 2006 with improved compression rate around 50% [10]. With an extra "lossless" computational layer, this approach expands on the MPEG-4 AAC lossy compression. Enhanced Scalable-to-Lossless (SLS) released in 2010 with faster decoding and encoding speed but having lower compression rate. Laplacian distribution input data is replaced with a Gaussian distribution in this model for the BPCG Entropy Block. Another adaptive coding based lossless audio codec; Enhanced Code Excited Linear Prediction (CELP) was introduced in 2010. From the processing speed perspective, it is faster than MPEG – 4 ALS but required more storage [11]. In order to eliminate intersample correlation, this model uses code-exited sample-by-sample adaptable coding. In 2013, entropy encoding based IEEE 1857.2 was introduced which have more than 50% compression rate, but processing performance is slow [12] because of Arithmetic coding's average computational complexity. In the same year another codec Sparse Linear Predictor [31] was introduced. This codec uses sparse predictors in place of LPC predictors. Although the compression ratio was higher, the decoding speed was slower with this model. In the next year, OLS and LMS filter based cascaded OLSNLMS was introduced which possess reduced computational complexities. Another popular lossless audio encoder Free Lossless Audio Codec (FLAC) released its latest version on 23rd June 2023 using MD5 and prediction model [13]. FLAC possesses around 70% average compression ratio [14]. Wavpack [15] released its latest version (5.6.0) on November 23, 2022 which have around 40% average compression ratio [14]. Famous lossless audio encoder Monkey's, which is based on integer discrete flow, achieves around 60% compression ratio [16]. An integrated model [17] of wavelet transform and Huffman encoding based lossless audio encoder introduced in 2020. In 2017 [18], a neural network-based model was introduced in which raw audio input transformed into features and processed using CNN subsequently. Lossless audio encoder based on dynamic cluster quantization technique [4] introduced In 2020. In the proposed clustering-based technique, dynamically cluster selection and bit selection for setting up the quantization level performed. In 2020, a deep learning based audio codec [5] was introduced which was based on hidden layerwise sampled value reduction technique. This technique improves the compression ratio(%) above 70% but computational complexities also increased. A model for audio compression based on deep neural networks [19] was presented in 2021 utilizing the RNN approach. As a reparametrization technique for discrete data representations, it applies the Bernoulli distribution and uses an end-to-end learning technique. This technique acheives average Signal to Distortion Ratio (SDR) of 20.53 dB with a compression ratio(%) exceeding 70%.

A Machine learning based toolkit [20] introduced recently which is used for unsupervised learning from acoustic data. The approach is based on repetitive sequential autoencoder approach which learn from time series type data using temporal motion. In these models, on the input sequence VAE is applied and RNN is applied on the output distribution subsequently to recognize the signal. Another audio compression approach that was unveiled in 2022 is the linear predictive neural net encoder (LINNE) [21]. It compresses audio by more than 60%. Also, another lossless encoding methodology based on optimum graph encoding was released in 2022 [3], and it significantly improves processing speed and compression efficiency. In 2022, The natural gradient sign algorithm (NGSA) and normalized NGSA are two adaptive algorithms that serve as the foundation for a lossless audio codec that is called NARU [32], or natural-gradient autoregressive lossless audio compressor. The utilization of a natural gradient in this work improves the sign algorithm's (SA) convergence performance. These methods significantly speed up decoding by using multiply-add operations to determine the natural gradient at each step, assuming a p-th order autoregressive model for the input data. Nonetheless, this method achieves a compression performance of about 60%.

Even though various neural network based as well as classical lossless audio compression approaches introduced, achieving compression rates like MP3 remain a mile away. Therefore, developing a lossless audio compression technique with higher compression rate and lower processing time is the need of the hour. The proposed lossless audio encoder based on neural network that achieves a higher compression ratio to fulfil this goal.

3 The technique

The proposed audio codec consists of 1D wavelet decomposition, hierarchical binary encoding and convolutional neural network enabled compressed latent space representation technique. Figure 1 displays the 1D three level wavelet decomposition tree. cA represents the approximation coefficients and cD represents the detail coefficients. Figure 3 shows the hierarchy structure.

The novelty of the proposed work lies in the integration of wavelet transformation with hierarchical encoding within a Convolutional Neural Network (CNN) architecture. This combination allows for more efficient feature extraction and data representation, leading to a significant improvement in the compression ratio. Wavelet transformation is leveraged to capture both time and frequency domain features, which are critical for effective audio compression. The hierarchical encoding further refines the data representation by breaking down the audio signal into progressively finer details, which are then processed by the CNN to identify and compress redundant information. The proposed model has been rigorously tested against both machine learning-based techniques and state-of-the-art traditional lossless audio benchmarks. The experimental results demonstrate that our model not only achieves a higher compression ratio but also maintains superior audio quality, thereby outperforming existing lossless audio techniques.

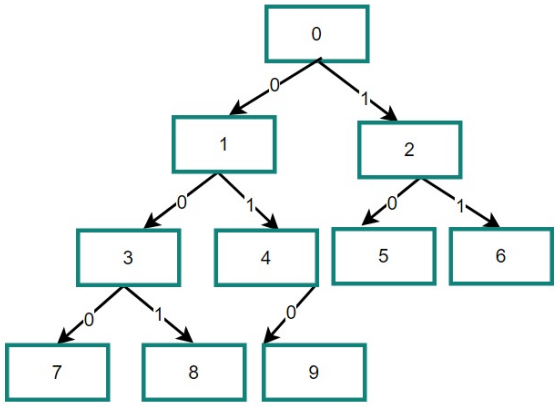


Figure 3: Hierarchical structure

Table 2 shows the nodes in each of the layer of the structure. Also, it contains the corresponding binary values of the level.

Table 2: Hierarchy levels and corresponding nodes

Level	Binary value	Nodes in the level
0	00	0
1	01	1,2
2	10	3,4,5,6
3	11	7,8,9

Algorithm 1: Encoding algorithm
Input: A slice of audio signal
Output: Compressed latent space representation of the input audio
Method: The steps are given as below

1 (Sampling): The input stream is sampled, and the sampled values are produced using the sampling process. Let f_{max} is the highest signal frequency and f_{sig} be the frequency used for sampling. The Nyquist theorem requires that the following criteria (1) be followed.

$$f_{sig} > 2x f_{max} \tag{1}$$

2 (Wavelet transformation): One dimensional three level wavelet decomposition is utilized for the input vector (created using audio sampled data) to create approximation coefficients (cA) and detail coefficients (cD). Equation (2)

represents the wavelet transformation over input audio x(t).

$$X_{a,b} = 2^{-\frac{a}{b}} \int_{-\infty}^{\infty} x(t)\Psi_{a,b}(t)dt \tag{2}$$

a and b work as frequency parameter and time respectively. $\Psi_{a,b}(t)$ demonstrates shifted and dilated variety of the mother wavelet $\Psi(t)$. It is shown in equation (3).

$$\Psi_{a,b}(t) = 2^{-\frac{a}{b}}\Psi(2^{-a}t - b) \tag{3}$$

$2^{-\frac{a}{2}}$ is constant. For wavelet transformation, which is iterative in nature, decimal valued α_{ab} coefficient is applied. A_{ab} is represented as approximation coefficient and α_{ab} as wavelet coefficient which are shown in equation 4 and 5 respectively .

$$A_{a,b}(t) = \sum_i l_{(2b-i)}A_{(a-1)i} \tag{4}$$

$$\alpha_{a,b}(t) = \sum_i h_{(2b-i)}\alpha_{(a-1)i} \tag{5}$$

The multilevel discrete 1-D wavelet transformation creates approximation coefficients (cA), detail coefficients (cD) in each level.

Step 3 (Wavelet coefficient segregation): If the coefficient is positive, add two binary digits 00 else 10 for negative numbers. Multiply the number with 10000. And get the absolute value. If the absolute value is single digit, prepend 3 zeros. If the absolute value is two digits, prepend 2 zeros and prepend 1 zero if the absolute value is three digits. Pass these 4 digits to hierarchical encoding module.

Step 4 (Hierarchical encoding):

i. Each of the digit from 0 to 9 is encoded using the hierarchical encoding technique following the figure 3 with pattern like <hierarchical level><accumulated path value> except 0.

Intput digit will be searched with the root node i.e., 0 which is on the first level. If it is matched, encode it with level number 0 and binary stream is 00. As 0 is the root node, to encode 0, no extra bit is added with level. Therefore, 0 will be coded as 00.

ii. If the input is 1 or 2, go to the table 2, corresponding level is 1 and binary stream 01. Start searching from left to right in the level 1. If the input digit is 1, it will be encoded as 010 as it is in the left node of 0. If 2, then it will be encoded as 011 as 2 is the right node of 0.

iii. If the input digits is 3 or 4 or 5 or 6, then as per the table 2, level is 2 and binary as 10. Searching start from left to right in the level 2. If the input is 3, it will be encoded as 1000 as it is in the left child of node 1. If the input is 4, it will be encoded as 1001 as it is in the right child of node 1. If the input is 5, it will be encoded as 1010 as it is in the left child of node 2. If the input is 6, it will be encoded as 1011 as it is in the right child of node 2.

iv. If the input digits is 7 or 8 or 9, then as per the table 2, corresponding level is 3 and binary as 11. Searching start from left to right in the level 3. If the input digit is 8, it will be encoded as 11001 as it is in the right child of node 3. If the input digit is 9, it will be encoded as 11010 as it is in the left child of node 9.

Step 4 (CNN encoder): These binary streams are sent to the CNN encoder for further compression.

Algorithm 2: Decoding algorithm
Input: CNN regenrated bit stream.
Output: Reconstructed original input audio.
Method: The steps are given as below

1. **Step 1 (Hierarchical decoding)**:
i. CNN reconstructed bit streams are checked. First 2 binary bits are checked, if these bits are 0, then it is the root node 0 and no extra bits need to check. It decoded as integer 0.
ii. Next 2 bits are checked, if it is 01, check the next single bit. If it is 0, decoded as integer 1, else 2.
iii. Check the next 2 bits, if it is 10, test the next 2 bits. If next 2 bits are 00, decode the integer as 3, 01 decoded as 4, 10 as 5, and 11 as 6.
iv. Test next 2 bits. If it is 11 then check next 3 bits. If next 3 bits are 000 then decoded as integer 7. If 001 then 8. If 010 then 9.
2. **Step 2 (Inverse wavelet transformation)**: Inverse wavelet is applied to reconstruct the input signal. Equation (5) is used for discrete inverse wavelet transformation.

$$x(t) = \sum_a \sum_b A_{a,b}\Psi_{a,b}(t) \tag{6}$$

3.1 CNN encoder decoder

Convolution Neural Network (CNN) [22] is very useful to extract spatial features from the dataset by using CNN kernel. In 2D CNN, Kernel slides along the two dimension. This 2D CNN is very useful in extracting features from image patterns. In the proposed technique, 2D convolutional layer is being used for feature extraction from 68 x 68 x1 input image patterns.

3.1.1 Data preprocessing

Binary data stream generated from hierarchical encodeing module preprocessed and fed as input to the CNN model. Whole binary stream is segregated into each row of 4624 columns and stored as .csv file. Each row of 4624 binary stream (0/1) is converted into into 68 x68 x 1 size image patterns sequentially. Each of the binary bit is transformed as pixel in the image. The CNN model is trained and tested with these image patterns.Figure 4 shows how the the binary strem is converted into image pattern required for CNN codec input.

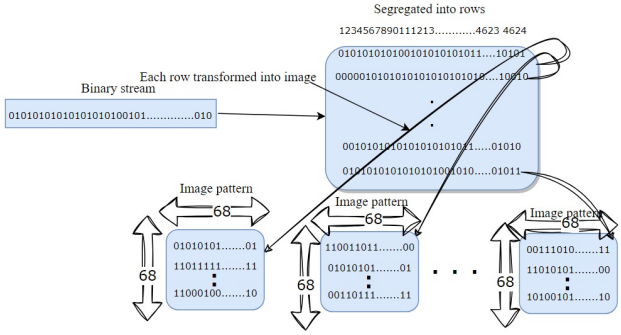


Figure 4: Image pattern formation

Table 3: CNN Model parameter configuration

Sl. no	Model Parameters	Value
1	Number of convolution layer	4
2	Convolution layer kernel size (4 layer)	(3,3)
3	Convolution layer stride	(2,2)
4	Convolution layer activation function	ReLU
5	Number of hidden units in LSTM layer	24
6	Batch size	128
7	Learning rate	0.001
8	Optimizer	Adam.
9	Loss function	MSE
10	Epochs	100

3.1.2 CNN model configuration

The proposed CNN model's computational efficiency is increased by the optimal number of setup parameters, such as convolutional layers, kernel size, stride, learning rate, and optimizer function. A trial-and-error approach is used to select the optimal parameters. Determining the optimal parameter selection is aided by observing the increased compression ratio, compression speed, and lower MSE. To compute model losses, the mean square error(MSE) is employed. In addition, the audio quality of the regenerated signal was evaluated using additional quality parameters such as PSNR, entropy, SDR, NCC, and MAE. Optimal design configuration parameters are described in table 3. The proposed CNN encoder decoder model consists of 4 convolution and 4 deconvolution layers. To extract the feature from the raw audio signal, we used 2D convolutional layer. To reconstruct the signal, deconvolutional feature extractor has been used. Adam optimization has been applied to fine tune the network. Various combinations of kernel and stride sizes were tried along with filters during the experiment. Trial and error methods were used to finalize the parameters until the least amount of loss was achieved. Input layer of the CNN accepts 68 x 68 x 1 size image generated from binary data as input, the output layer of the CNN extracts features from input, and hidden layers are used for processing purposes. With this architecture, the learning of the neural network performed for every input a weight that demonstrates a particular output. Convolution kernels are (3,3) in each layer with an activation function (ReLU). The CNN layer selected above for the experiment by trial-and-error method for getting the best result. To prevent overfitting dropout is considered as 0.1. 100 epochs were selected for evaluating the performance of the current model. Figure 11 shows the shpae and parameters of the CNN encoder model. Figure 12 shows the shpae and parameters of the CNN decoder model. Figure 13 shows the shpae and parameters of the CNN encoder decoder combined model.

3.1.3 Training

The entire CNN network is built and trained using the Tensorflow / Keras framework [15]. The ability to create network layers and train the network in according to the suggested specifications is greatly facilitated by Tensorflow / Keras. As a result, the network is trained after it converges, and appreciable decrease in training loss is seen. The final step is to evaluate, examine, and compare the total results to the established benchmark results.

The practical aspects of implementing the proposed codec are addressed by highlighting that the method can be tailored to specific needs in the audio compression field. Implementing the suggested model will lead to reduced complexity, improved compression ratio, and faster processing speeds. Additionally, it will decrease memory usage and network bandwidth requirements, making it more efficient for real-world applications

4 Experimental setup

This section discusses a number of necessary elements for the experiment, such as the environment setup, dataset preparation, data preprocessing, tools and software used, etc.

4.1 Environment

Tensorflow and Keras framework with Python 3.6 was used to implement the proposed model. Intel Core i7-4790S Processor, 16 GB RAM, 64-bit operating system, and 1 TB Hard drive were used to carry out the experiment.

4.2 Datasets

We prepared a customized dataset WLCLAC_model_training_dataset to train the WLCLAC model. Three hundred audio files, all of the same duration approximate (3 seconds), make up the dataset. Rabindra Sangeet, classical, rock, pop, and sufi are among the genres of audio songs. The training dataset was not divided into separate sets for testing. Rather, we employ 25 distinct audio tracks that fall into the five categories listed above, each lasting approximately ten seconds. The training and testing datasets are prepared using audio songs of the.wav file type, with a sampling rate of 44100 Hz. To record the

Table 4: Recording parameters

Recording parameter name	Values
Recorded file format	.wav
Recording time(training) approx.	3 seconds
Sampling rate	44100 Hz
Recording time(testing) approx.	10 seconds
Bit depth	16 bits
Channel	Mono (1)

music in.wav format and play the audio on the computer, Audacity software (Audacity 2.3.2) was utilized. The two-channel stereo audio signals are transformed to mono. The audio track has 16 bits of bit depth.

4.3 Details of recording parameters

The preparation process and parameters for the WLCLAC_audio_training_and WLCLAC_audio_testing_datasets is covered in detail in this section. Songs are recorded in.wav format using Audacity, resulting in a data collection with a 44100 Hz sample rate. The songs that comprise the customized dataset are recorded using five standard parameters. Table 4 displays the parameters' setup values.

4.4 Evaluation metrics

Evaluation of the proposed model's correctness and performance is required. In order to assess performance, several metrics are employed which are discussed below.

– Compression. Compression measures the amount of storage space the model can spare for the data. To compute the space saving, utilize equation (7) [17]. Compression ratio is a crucial metric for assessing the suggested model's capacity for compression. Here, the recommended WLCLAC approach was used to achieve average 85.72% compression while maintaining signal quality.

$$Compression(\%) = \frac{Original–compressed}{Original} \times 100 \tag{7}$$

– Mean square error(MSE). Performance of the proposed audio codec is assessed using mean square error[23]. Here, equation (8) is utilized to determine the MSE of the present model.

$$MSE = \frac{\sum_{j=1}^{n}(x_j - x'_j)^2}{n} \tag{8}$$

Where n is the number of sample points, x_j and x'_j are the actual and reconstructed values of each data point, respectively

– Entropy. Entropy is the average amount of information contained in a symbol or variable [24]. The unpredictable nature is shown by the entropy. Equation (9) is used to calculate the entropy for the proposed model.

$$H(X) = \sum_{k=1}^{m} P(x_k)\log_2 P(x_k) \tag{9}$$

H(x) indicates entropy of x. Here, x denotes the random variable. It takes values from the set of values $x_1, x_2, ...x_m$ corresponding probabilities $P(x_1), P(x_2), ..P(x_m)$ $where$ $\sum_{k=1..m} P(x_k) = 1$

– PSNR(in dB). A technique for assessing the quality of the original signal in compressed audio files is the peak signal to noise ratio (PSNR) [25] [26]. Equation (10) is used to calculate PSNR

$$PSNR = IJ\frac{\sum_{i,j} X_{i,j}^2}{\sum_{i,j}(X_{i,j} - \overline{X}_{i,j})^2} \tag{10}$$

Here, $X_{i,j}$ denotes the original values and is represented by $\overline{X}_{i,j}$ reconstructed values.

– Normalised cross correlation. Input and reconstructed audio signals are compared using NCC. Higher correlation is indicated by a higher NCC. Two identical signals result in a score of 1. NCC is calculated using equation (11).

$$NCC = \frac{\sum_{i,j} X_{i,j}\overline{X}_{i,j}}{\sum_{i,j} X_{i,j}^2} \tag{11}$$

Where $\overline{X}$ are the reconstructed values, and X are the input values to the model.

– Mean Absolute Error(MAE). The average of the variations between the generated values and the original values is referred to as the "mean absolute error" [27]. This measure displays the variations between the input and the reconstructed value. Equation (12) gives the following illustration of it:

$$MAE = \frac{\sum_{j}^{n}|X_j - X'_j|}{n} \tag{12}$$

– Signal distortion ratio(SDR): The experiment uses SDR[28] to measure the reconstructd signal's audio quality. The SDR uses decibels (dB). SDR is a measure of how close the reconstructed signal(S_{recon}) was to the original signal(S_{orig}). The calculation is as follows. The proposed method produces an average Signal to Distortion Ratio (SDR) of 42.45 dB.

$$SDR = 10log_{10}\frac{|S_{orig}|^2}{|S_{recon} - S_{orig}|^2} \tag{13}$$

5 Results and analysis

The wavelet decomposition module receives all of the audio from the WLCLAC_audio_training_dataset. Using the proposed hierarchical binary encoder, the appropriate wavelet coefficients are encoded into binary patterns. The CNN encoder-decoder model uses all of the binary data streams that match the training dataset as input. 30% of the CNN model's input training dataset is used for validation and 70% for training. The CNN model testing dataset is generated in a similar manner from the WLCLAC_audio_testing_dataset. The WLCLAC codec is tested individually using this testing dataset. In order to compute the compression independently, the three stages of the suggested codec are assessed during the experiment. The results are thoroughly explained in Section 4.1.

5.1 Experimental results

Three existing conventional lossless audio compression techniques like Monkey's Audio [29], Wavpack Lossless [15], and FLAC [13] are considered as referenced systems to evaluate the performance of the proposed model. Table 5 shows the compression performance of the proposed codec with the three existing codecs and it is evident that the current model achieves 85.72% (shown in table 5) compression which is higher than 56.45%, 51.18%, and 70.64% compressions achieves by Monkey's Audio [29], Wavpack Lossless [15], and FLAC [13] respectively. Figure 5 shows the graphical representation of the compression, PSNR, and entropy achieved using WLCLAC with reference to existing referenced lossless audio compression techniques.

Using a variety of audio tracks, the compression speed implies encoding and decoding times of the proposed technique are compared with other state-of-the lossless audio codecs. The resultset is shown in table 6 and it is evident that a reduction in the encoding and decoding times of the suggested approach translates into an increase in compression speed. Figure 6 compares the encoding and decoding speed graphically. For all the parameters, WLCLAC achieves better results.

Table 5: Performance of the proposed model in relation to the existing audio codecs

Method	Compression(%)	Entropy	PSNR(dB)
Monkey's Audio	56.45	13.35	52.98
Wavpack	51.18	13.55	52.12
FLAC	70.64	13.56	52.56
WLCLAC	85.72	13.67	56.56

Also, we have done the robustness performance evaluation of the proposed model with 3 others existing deep learning based lossless audio codecs : i) DLLAE [5] ii) Daniela N. Rim et al.[19] iii) LINNE [30]. Proposed WLCLAC model is capable of regenerating the original audio signal with very negligible deviations. Mean square error

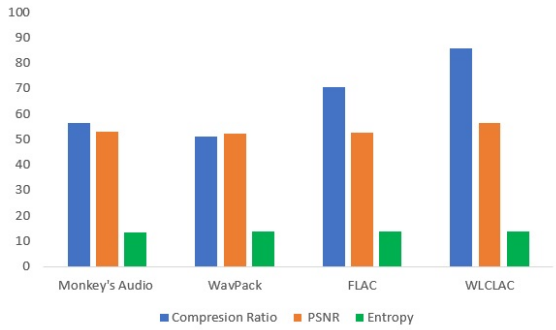


Figure 5: Comparison of the compression and quality parameters for WLCLAC

Table 6: Encoding and decoding time comparison

Method	Encoding(sec.)	Decoding(sec.)
Monkey's Audio	0.06782	0.06802
Wavpack	0.07134	0.07211
FLAC	0.06871	0.06921
WLCLAC	0.06321	0.06431

value evaluated for the proposed model is 0.001822. Also, the RMSE of the proposed model is 0.042684. MAE value calculated for the proposed model is 0.033912. Close to 0 value of the MAE and RMSE indicates the robustness of the system and close similarities between original and predicted signal. Also, another parameter called NCC used to measure the regenerated signal quality is 0.998761. Closer to 1 NCC value indicates regeneration is good. Table 7 shows the evaluated values of the parameters like MSE, RMSE, MAE, and NCC of the current model with respect to the other referenced prediction system to demonstrate the accuracy and robustness of the model. Table 8 compares the compression ratio(%) produced by the new approach to the other DNN model. According to table 8, the proposed lossless audio codec produces more compression than the existing models. Figure 7 shows the graphic comparison of category-wise compression of compression ratio (%) of the proposed model with other existing neural network based models.

Table 7: Robustness performance comparison with exiting DNN model

Method	MSE	RMSE	MAE	NCC
WLCLAC	0.001822	0.042684	0.33912	0.998761
DLLAE [5]	0.017872	0.133686	1.161451	0.985634
Daniela N. Rim et al.[19]	0.134536	2.185838	2.617224	0.981232
LINNE[30]	0.125162	0.158625	2.017224	0.986578

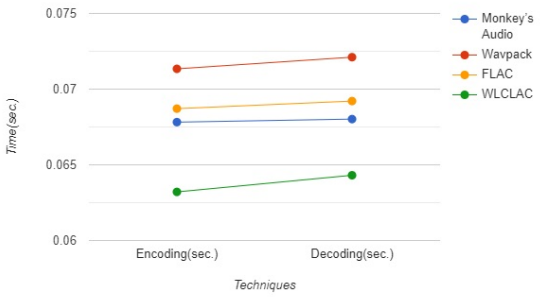


Figure 6: Comparison of the compression speed

Table 8: Categorywise compression comparison

Method	Pop	Sufi	Ghazal	Rabi	Classical
DLLAE [5]	87.18	86.47	86.01	87.01	86.12
Daniela N. Rim et al. [19]	84.21	82.34	82.15	83.81	83.27
LINNE [30]	73.76	75.32	73.38	71.13	73.48
Proposed method	85.18	85.01	86.89	85.4	86.12

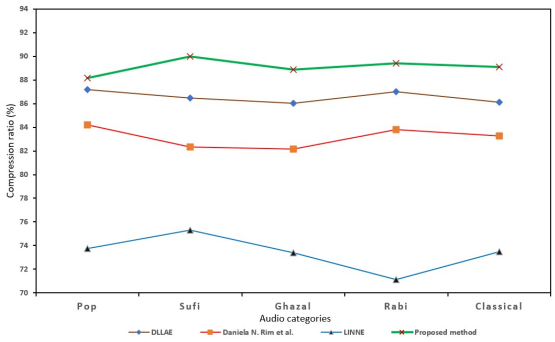


Figure 7: Comparison of the compression ratio(%)with existing DNN models

Figure 8 represents the regenerated audio signal by the WLCLAC model. Therefore, it is visible that regenerated signal is like original signal with negligible deviations. Therefore, the experimental data set demonstrates that the proposed method, when compared with existing standard audio compression approaches, acquired a higher compression ratio and improved audio regeneration quality.

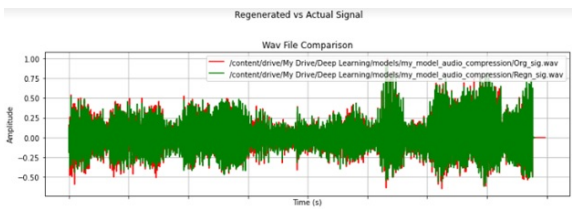


Figure 8: Comparison of the original and regenerated signal

Figure 9 shows the epochwise loss of the model. Figure 10 shows the epochwise training pregress and corresponding mse loss of the model.

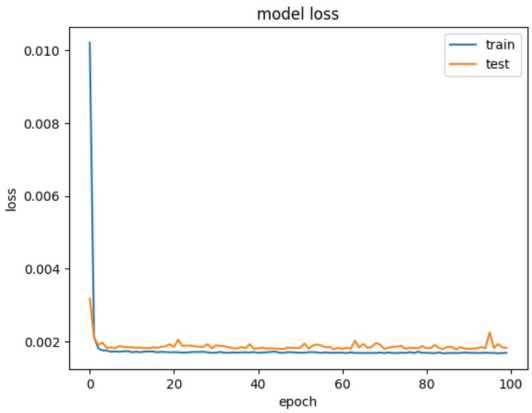


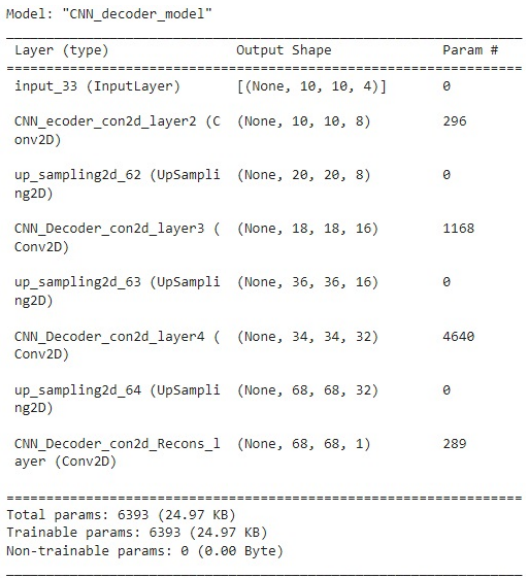
Figure 9: Model loss

```
Epoch 92/100
343/343 [==============================] - 4s 10ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 93/100
343/343 [==============================] - 4s 12ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 94/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 95/100
343/343 [==============================] - 4s 13ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 96/100
343/343 [==============================] - 5s 13ms/step - loss: 0.0017 - val_loss: 0.0023
Epoch 97/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 98/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0019
Epoch 99/100
343/343 [==============================] - 4s 13ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 100/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Running prediction..
85/85 [==============================] - 1s 5ms/step
Plotting results..
metrics name is ['loss']
```

Figure 10: Epochwise training progress

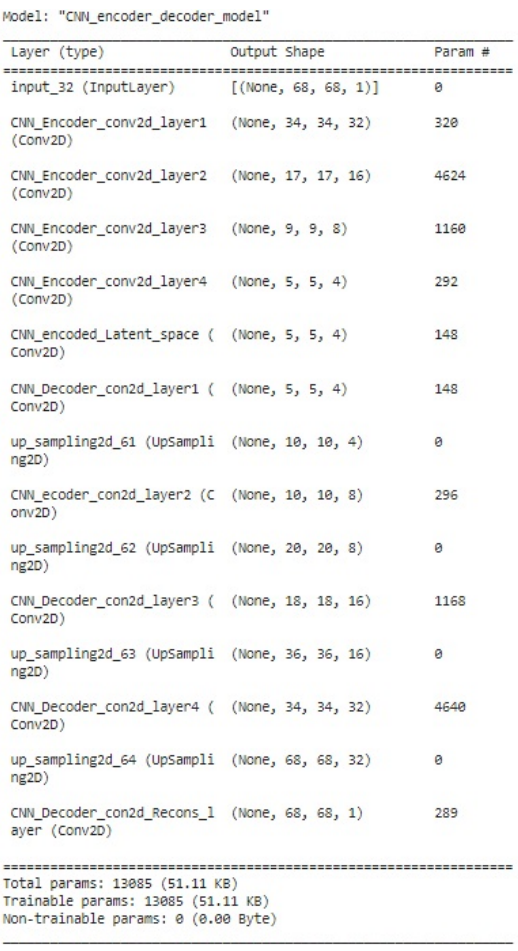
```
Model: "CNN_encoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_32 (InputLayer)       [(None, 68, 68, 1)]       0

 CNN_Encoder_conv2d_layer1   (None, 34, 34, 32)        320
 (Conv2D)

 CNN_Encoder_conv2d_layer2   (None, 17, 17, 16)        4624
 (Conv2D)

 CNN_Encoder_conv2d_layer3   (None, 9, 9, 8)           1160
 (Conv2D)

 CNN_Encoder_conv2d_layer4   (None, 5, 5, 4)           292
 (Conv2D)

 CNN_encoded_Latent_space (  (None, 5, 5, 4)           148
 Conv2D)

=================================================================
Total params: 6544 (25.56 KB)
Trainable params: 6544 (25.56 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 11: Shape and parameters of the CNN encoder model



```
Model: "CNN_decoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_33 (InputLayer)       [(None, 10, 10, 4)]       0

 CNN_ecoder_con2d_layer2 (C  (None, 10, 10, 8)         296
 onv2D)

 up_sampling2d_62 (UpSampli  (None, 20, 20, 8)         0
 ng2D)

 CNN_Decoder_con2d_layer3 (  (None, 18, 18, 16)        1168
 Conv2D)

 up_sampling2d_63 (UpSampli  (None, 36, 36, 16)        0
 ng2D)

 CNN_Decoder_con2d_layer4 (  (None, 34, 34, 32)        4640
 Conv2D)

 up_sampling2d_64 (UpSampli  (None, 68, 68, 32)        0
 ng2D)

 CNN_Decoder_con2d_Recons_l  (None, 68, 68, 1)         289
 ayer (Conv2D)

=================================================================
Total params: 6393 (24.97 KB)
Trainable params: 6393 (24.97 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 12: Shape and parameters of the CNN decoder model



```
Model: "CNN_encoder_decoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_32 (InputLayer)       [(None, 68, 68, 1)]       0

 CNN_Encoder_conv2d_layer1   (None, 34, 34, 32)        320
 (Conv2D)

 CNN_Encoder_conv2d_layer2   (None, 17, 17, 16)        4624
 (Conv2D)

 CNN_Encoder_conv2d_layer3   (None, 9, 9, 8)           1160
 (Conv2D)

 CNN_Encoder_conv2d_layer4   (None, 5, 5, 4)           292
 (Conv2D)

 CNN_encoded_Latent_space (  (None, 5, 5, 4)           148
 Conv2D)

 CNN_Decoder_con2d_layer1 (  (None, 5, 5, 4)           148
 Conv2D)

 up_sampling2d_61 (UpSampli  (None, 10, 10, 4)         0
 ng2D)

 CNN_ecoder_con2d_layer2 (C  (None, 10, 10, 8)         296
 onv2D)

 up_sampling2d_62 (UpSampli  (None, 20, 20, 8)         0
 ng2D)

 CNN_Decoder_con2d_layer3 (  (None, 18, 18, 16)        1168
 Conv2D)

 up_sampling2d_63 (UpSampli  (None, 36, 36, 16)        0
 ng2D)

 CNN_Decoder_con2d_layer4 (  (None, 34, 34, 32)        4640
 Conv2D)

 up_sampling2d_64 (UpSampli  (None, 68, 68, 32)        0
 ng2D)

 CNN_Decoder_con2d_Recons_l  (None, 68, 68, 1)         289
 ayer (Conv2D)

=================================================================
Total params: 13085 (51.11 KB)
Trainable params: 13085 (51.11 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 13: Shape and parameters of the CNN encoder decoder model

The Mean Opinion Score (MOS) is used to assess the perceptual quality of the regenerated audio signals . Table 9 displays the MOS measurement of the regenerated audio quality. According to table 9, a sound quality grade of ”5” indicates ”Excellent” sound, while a grade of ”1” indicates ”Bad” sound. The ITUR Rec. 500 quality rating is appropriate for the present quality measuring activities. because it offers a quality rating ranging from 1 to 5 [33]. The MOS (mean opinion score) number for the various categories evaluated by the current technique, 5, indicates that the reconstructed audio quality remains unaffected by this tiny data change throughout the transformation, since it is over the threshold level of human perception.

Table 9: A rating system to evaluate the decline in audio quality.

Rating	Impairment	Quality
1	Very annoying	Bad
2	Annoying	Poor
3	Slightly annoying	Fair
4	Perceptible, not annoying	Good
5	Imperceptible	Excellent

6 Conclusion and future scope

The proposed model has been trained and validated using real-time audio data. The proposed model’s performance is assessed in comparison with those of current standard lossless audio codecs. The mean square error of the robust model is very less. Compression of the proposed model is 85.72%, which is higher compared with existing lossless audio codecs. The computational time for the model is lower than the referenced systems. The future scope of the work is to enhance the computational performance and compression by enhancing the model.

References

[1] Nowak, N., Zabierowski, W.(2011): Methods of sound data compression–comparison of different standards. Radio electronics and informatics (4), 92–95.

[2] Sharma, K., Gupta, K.(2017), Lossless data compression techniques and their performance. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 256–261, IEEE. https://doi.org/10.1109/CCAA.2017.8229810

[3] Mondal, U.K., Debnath, A.(2022), Designing a novel lossless audio compression technique with the help of optimized graph traversal (lacogt). Multimedia Tools and Applications 81(28), 40385–40411.

[4] Mondal, U.K., Debnath, A.(2021), Developing a dynamic cluster quantization based lossless audio compression (dcqlac). Multimedia Tools and Applications 80(6), 8257–8280.

[5] Mondal, U.K., Debnath, A., et al.(2020), Deep learning-based lossless audio encoder(dllae). In: Intelligent Computing: Image Processing Based Applications, pp. 91–101. Springer. https://doi.org/10.1007/978-981-15-4288-6_6

[6] Mondal, U.K., Debnath, A., et al.(2023), Designing an iterative adaptive arithmetic coding-based lossless bio-signal compression for online patient monitoring system(iaalbc). In: Frontiers of ICT in Healthcare: Proceedings of EAIT 2022, pp. 655–664. Springer. https://doi.org/10.1007/978-981-19-5191-6_53

[7] Holighaus, N., Koliander, et al.(2019), Characterization of analytic wavelet transforms and a new phaseless reconstruction algorithm. IEEE Transactions on Signal processing 67(15), 3894–3908.

[8] Jmour, N., Zayen, S., Abdelkrim, A.(2018), Convolutional neural networks for image classification. In: 2018 International Conference on Advanced Systems and Electric Technologies (IC ASET), pp. 397–402,IEEE.

[9] Reznik, Y.A.(2004), Coding of prediction residual in mpeg-4 standard for lossless audio coding (mpeg-4 als). In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, p. 1024,IEEE. https://doi.org/10.1109/ICASSP.2004.1326722

[10] Yu, R., Lin, X., Rahardja, S., Huang, H. (2005), Mpeg-4 scalable to lossless audio coding-emerging international standard for digital audio compression. In: 2005 IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4,IEEE. https://doi.org/10.1109/MMSP.2005.248562

[11] Wei, B., Wang, J., Gibson, J.D. (2001), Enhanced celp coding with discrete spectral modeling. In: Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489), pp. 111–113,IEEE. https://doi.org/10.1109/ISIMP.2001.925344

[12] Gunawan, T.S., Zain, M.K.M., Muin, F.A., Kartiwi, M. (2017), Investigation of lossless audio compression using ieee 1857.2 advanced audio coding. Indonesian Journal of Electrical Engineering and Computer Science 6(2), 422–430. https://doi.org/10.11591/ijeecs.v6.i2.pp422-430

[13] Coalson, J.: Xiph. Org Foundation,"FLAC: Free lossless audio codec". https://xiph.org/flac/index.html. Accessed:15-10-2024.

[14] Tu, W., Yang, Y., Du, B., Yang, W., Zhang, X., Zheng, J.(2020), Rnn-based signal 339 classification for hybrid audio data compression. Computing 102(3), 813–827. https://doi.org/10.1007/s00607-019-00713-8

[15] http://www.wavpack.com/. Accessed: 15-10-2024.

[16] Oquab, M., Bottou, L., Laptev, I., Sivic, J.(2015), Is object localization for free?-weakly supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 685–694. https://doi.org/10.1109/CVPR.2015.7298668

[17] Debnath, A., Mondal, U.K., et al. (2020), Achieving lossless audio encoder through integrated approaches of wavelet transform, quantization and huffman encoding (laeiwqh). In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–5, IEEE. https://doi.org/10.1109/ICCSEA49143.2020.9132865

[18] Zeng, Ming, and Huahong Zeng. "Research on Violin Audio Feature Recognition Based on Mel-frequency Cepstral Coefficient-based Feature Parameter Extraction." Informatica 48, no. 19 (2024).

[19] Rim, D.N., Jang, I., Choi, H.(2021) Deep neural networks and end-to-end learning for audio compression. arXiv preprint arXiv:2105.11681. https://doi.org/10.5626/JOK.2021.48.8.940

[20] Freitag, M., Amiriparian, S., et al. (2017), audeep: Unsupervised learning of representations from audio with deep recurrent neural networks. The Journal of Machine Learning Research 18(1), 6340–6344. https://doi.org/10.48550/arXiv.1712.04382

[21] Mineo, T., Shouno, H.: A lossless audio codec based on hierarchical residual prediction. (2022), In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 123–130, IEEE. https://doi.org/10.23919/APSIPAASC55919.2022.9980327

[22] Kadhim, A.R., Khudeyer, R.S. and Alabbas, M., 2024. Facial Sentiment Analysis Using Convolutional Neural Network and Fuzzy Systems. Informatica, 48(12).

[23] Wang, K., Qi, X., Liu, H. (2019), Photovoltaic power forecasting based lstm convolutional network. Energy 189, 116225. https://doi.org/10.1016/j.energy.2019.116225

[24] Shannon, C.E. (1948), A mathematical theory of communication. The Bell system 366 technical journal 27(3), 379–423.

[25] Kutter, M., Petitcolas, F.A.: Fair benchmark for image watermarking systems.(1999),In: Security and Watermarking of Multimedia Contents, vol. 3657, pp. 226–239 International Society for Optics and Photonics. https://doi.org/10.1117/12.344672

[26] Manju, M., Abarna, P., Akila, U., Yamini, S.(2018),Peak signal to noise ratio & mean square error calculation for various image patterns using the lossless image compression in ccsds algorithm. International Journal of Pure and Applied Mathematics 119(12),14471–14477.

[27] Jie Y. Design and Application of Neural Network-based Bp Algorithm in Speech Translation Robot. Informatica. 2023 Aug 4;47(7).

[28] Krizhevsky, A., Sutskever, I., Hinton, G.E.(2012), Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25. https://doi.org/10.1145/3065386

[29] https://monkeysaudio.com/index.html. Accessed: 15-10-2024.

[30] Mineo, T., Shouno, H.(2022), A lossless audio codec based on hierarchical residual prediction. In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 123–130, IEEE. https://doi.org/10.23919/APSIPAASC55919.2022.9980327

[31] Giacobello D, Christensen MG, Murthi MN, Jensen SH, Moonen M. Sparse linear prediction and its applications to speech processing. IEEE Transactions on Audio, Speech, and Language Processing. 2012 Feb 3;20(5):1644-57. https://doi.org/10.1109/TASL.2012.21868071

[32] Mineo, T., Shouno, H.: Improving sign-algorithm convergence rate using natural gradient for lossless audio compression. EURASIP Journal on Audio, Speech, and Music Processing 2022(1), 12 (2022). https://doi.org/10.1186/s13636-022-00243-w

[33] Arnold M (2000) Audio watermarking: features, applications and algorithms. In: 2000 IEEE International conference on multimedia and expo. ICME2000. Proceedings. Latest advances in the fast changing world of multimedia (cat. no. 00TH8532), vol 2. IEEE, pp 1013–1016. https://doi.org/10.1109/ICME.2000.871531