applied sciences

MDPI

Review

Audio Watermarking: Review, Analysis, and Classification of the Most Recent Conventional Cutting-Edge Results

Carlos Jair Santin-Cruz * and Gordana Jovanovic Dolecek *

Department of Electronics, National Institute of Astrophysics, Optics and Electronics, Puebla 72840, Mexico
* Correspondence: c.santin@inaoep.mx (C.J.S.-C.); gordana@ieee.org (G.J.D.)

Abstract

Audio watermarking has been introduced to give authors and owners control over the use of audio signals. The need for control has increased with advances in digital communications. Due to the diversity of applications and trade-offs in performance, different works on audio watermarking have been reported, and new algorithms have been proposed. Being an active area of research, an updated background on the reported algorithms helps designers understand the trends and tools used in published works. The recent emergence of innovative methods to enhance performance, the need for new requirements due to increased potential attacks, advancements in technology with a tendency toward improved audio quality, and the development of new applications have made it challenging to fit these developments into existing reviews and classifications. This paper fills this gap, presenting a review, analysis, and classification of the most recent conventional watermarking algorithms from 2016 to the present. Our study reveals the predominance of blind watermarking approaches and the widespread adoption of wavelet-domain techniques due to their favorable balance between robustness and imperceptibility. We proposed organizing, discussing, and comparing the methods based on performance criteria, imperceptibility, capacity, security, computational complexity, and robustness, and setting thresholds to categorize them. Additionally, a novel systematization based on the processes involved in the various stages of watermarking is presented. The purpose is to make it easier to identify the performance criteria that could be useful and important for different applications.

Keywords: audio watermarking; copyright protection; security; digital watermarking; codification

check for updates

Academic Editors: George Drosatos and Guangjie Han

Received: 18 September 2025
Revised: 10 October 2025
Accepted: 25 October 2025
Published: 28 October 2025

Citation: Santin-Cruz, C.J.; Dolecek, G.J. Audio Watermarking: Review, Analysis, and Classification of the Most Recent Conventional Cutting-Edge Results. *Appl. Sci.* **2025**, *15*, 11514. https://doi.org/10.3390/app152111514

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The recent development of digital communications has increased the traffic of multimedia files. Digital content can be copied without quality degradation and shared on a large scale, making the protection of intellectual content more necessary. There are three principal techniques for information hiding: cryptography, stenography, and digital watermarking [1].

Cryptography protects the signal only when it has not been decrypted; after this process, the signal can be shared, altered, or copied. In stenography, the approach differs; the information embedded is not intended to protect the host signal, and this signal is used solely as a channel to conceal information.

Digital watermarking can be defined as the process of embedding information into a host signal without compromising access to the host signal's information. Watermarking allows access to the host signal, while the embedded data remains within it. Digital

watermarking can be embedded in various types of multimedia content, including images, audio, video, and text. Audio watermarking is a crucial and challenging form of digital watermarking, as audio signals are extensively used in the entertainment industry and for communication purposes. Ideally, the watermark should be imperceptible; if the watermark is noticeable, this can be annoying or even make the host signal intelligible. That is because the human auditory system (HAS) is much more sensitive than the human visual system [2]. However, because imperceptibility in audio depends on listener variability, content, and listening conditions, absolute imperceptibility cannot be guaranteed. At best, we can bound detectability under specified protocols and conditions.

Since audio watermarking has numerous applications, many algorithms seeking to reduce the compromises have been reported in the literature, using different techniques. As a result, it is essential to have an updated understanding of the reported algorithms to help designers create algorithms that minimize trade-offs and fulfill the requirements for the desired application. The previous works, such as [3–5], reviewed some reported works and the principal lines of investigation. However, these review papers did not cover the most recently proposed algorithms. In the pursuit of improved performance and reduced compromises, algorithms have incorporated various tools into the watermarking process. Previous works may not address some of these tools, as they focus on classifying watermarking as a single process. These classifications may be insufficient for classifying new watermarking systems.

We aim to review, analyze, and classify the most recent conventional audio watermarking algorithms that have not been previously reviewed in the literature. We reviewed forty-one papers published since 2016, including three conference papers [6–8] and thirty-eight journal papers [9–45]. The search was conducted across IEEE Xplore, ScienceDirect, SpringerLink, MDPI, and Google Scholar. The main keywords included “audio watermarking”, “blind watermarking”, and “digital watermarking”.

The inclusion criteria required peer-reviewed studies that presented or evaluated watermarking algorithms for audio signals. Exclusion criteria included studies without performance evaluation or focused on image/video watermarking. In addition to the inclusion and exclusion criteria described above, the citation count (as indexed in Google Scholar and Scopus) was used as a secondary indicator of research impact. This metric helped prioritize the most influential or representative works within each category, while ensuring that all selected papers met the technical relevance and methodological criteria established for this review.

The novelty of this work lies in its exploration of the underlying mechanisms through an analysis of the processes employed in existing algorithms, thereby helping to identify the main techniques used in various methods and how they can be combined or modified to achieve better performance. Additionally, a classification based on performance criteria results is presented, establishing thresholds to designate subjective criteria such as robustness, capacity, and imperceptibility.

This paper primarily reviews the conventional audio watermarking techniques, which have been vital in protecting intellectual property for many years. However, many deep learning watermarking methods have also been proposed recently. Nevertheless, the conventional audio and deep learning watermarking techniques represent two distinct approaches to embedding information into audio signals with the goal of copyright protection.

Both techniques are important and have their advantages and disadvantages. Conventional methods offer established techniques with well-established properties. At the same time, deep learning methods offer enhanced adaptability and robustness, but also pose potential challenges in security and computational complexity. The choice between

them depends on various factors, including specific applications, available resources, and the desired trade-offs between performance metrics.

The paper is organized as follows. The next section describes audio watermarking applications and its performance evaluation. Section 3 provides systematization based on the processes in the proposed systems. Section 4 classifies the methods according to the results reported in the performance criteria. Finally, Section 5 discusses the concluding remarks.

2. Audio Watermarking

2.1. Applications

As shown in [1–3], watermarking can be applied to a wide range of applications. However, the main applications addressed in this work include copyright protection, authentication, content identification and management, monitoring, and second-screen services.

Copyright protection: The hidden information allows the content creator to prove the ownership of their work. Additionally, this information can block unauthorized actions, including copying, unauthorized playback, or tampering, as authorized by the copyright owner.

Authentication: The watermark verifies the content's legitimacy or whether it has been tampered with. This application can be used as evidence in forensic situations for legal purposes. It can also be utilized for authentication when the voices or music have been created for artificial intelligence.

Content identification and management: Confidential information is used to identify and manage the content. The former is very useful for automating large amounts of content. It can also be helpful when content is restricted to specific users or complete access is denied.

Monitoring: The watermark can collect information about the broadcasting. This information helps detect illicit broadcasts and can be used for billing or statistical purposes.

Second screen: The watermark provides extra information that can be used by another device to show complementary material.

Watermarks may have different requirements depending on the applications mentioned. Therefore, it is essential to identify which tools better meet these requirements or which ones could be improved or adapted for a given application.

2.2. Performance Criteria

The performance of conventional audio watermarking is evaluated in terms of how it may satisfy specific requirements imposed by the application.

However, the following performance criteria should usually be considered [1–3]: imperceptibility, robustness, capacity, and computational complexity.

Imperceptibility: This requirement is of the utmost importance in audio watermarking. Imperceptibility is more challenging in audio files than in other types of multimedia. Ideally, the audio watermarked signal listener should not be able to detect the difference between the signal with and without a watermark. Watermark perceptibility decreases the audio quality, which can be annoying. But it is important to emphasize that even watermarks classified as imperceptible do not achieve complete imperceptibility but only to a certain degree.

Robustness: The ability of the watermark to recover after one or more attacks on the watermarked signal is referred to as robustness. After embedding, the signal is shared, passing through a distribution process in which it can undergo modifications. These modifications can make it more difficult to recover the watermark.

Capacity: The amount of information that can be embedded into the host signal is referred to as capacity. Typically, the watermark is digital; therefore, capacity represents

the number of bits that can be embedded in the host signal. The host signals have different durations, so the capacity is often measured over a fixed duration.

Security: The information embedded in the host signal should be confidential. Security refers to the ability to deny access to the embedded watermark to unauthorized individuals or resist an attack that aims to delete or modify the watermark.

Computational Complexity: This performance criterion refers to the necessary resources or time required to embed and recover the watermark message. It is essential when the algorithm is implemented in a device with limited computational resources or in real-time applications.

The authors in [1,4] have reported the trade-offs between imperceptibility, robustness, and capacity. These trade-offs are known as "the magic triangle" because improving one performance criterion harms the other two. However, there are different trade-offs; for example, performance in terms of computational complexity and security can be compromised when another performance criterion is improved.

3. Systematization Based on the Processes

In [46], two main processes (the embedding and extraction) are presented. A review of the literature reveals that the application of various techniques often involves two distinct phases: embedding and retrieval. For example, the encoding of the watermark message occurs during embedding and is decoded during recovery. Therefore, this work considers the entire process, regardless of whether one part is in embedding and the other in recovery. Five processes are identified to describe the functioning of the proposed watermarking systems. These processes are as follows:

- Audio Signal Preprocessing
- Embedding/Recovery Method
- Watermark Process
- Adaptive Process
- Auxiliary Signal Process

The block diagram in Figure 1 shows these processes in the flow of an audio watermarking system.

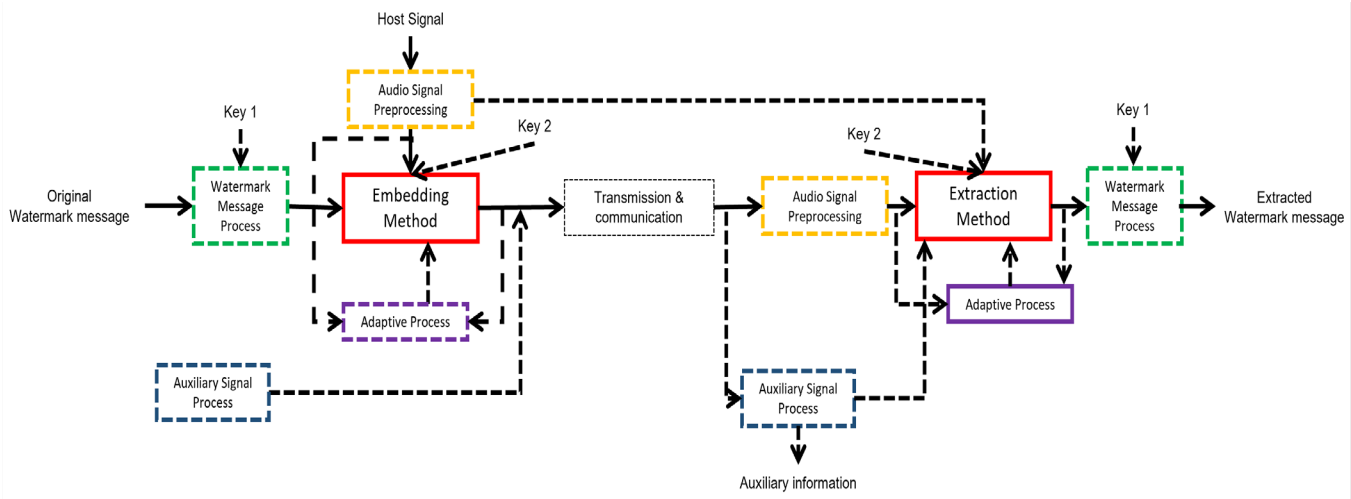


Figure 1. Block diagram summarizing the proposed process-based classification.

Not all systems perform all processes, or the flow may differ; therefore, dotted lines indicate the processes, inputs, and outputs that can be omitted in some systems. One of the inputs that is very important to highlight is the host signal in the recovery stage. The reason is that the use of this information in the recovery of some applications is impossible.

Also, the performance would differ due to this information. As a result, the literature typically categorizes algorithms as blind, semi-blind, or non-blind based on the presence or absence of the original host signal during the recovery stage. In the literature review, 90% of the algorithms are blind, as shown in Figure 2, indicating a marked trend in this type of watermarking. Some works require the host signal or information to recover the watermark, such as works [32,38], which are non-blind. Works [39,44] are considered semi-blind in this review because they require some information from the SVD transform of the host signal.

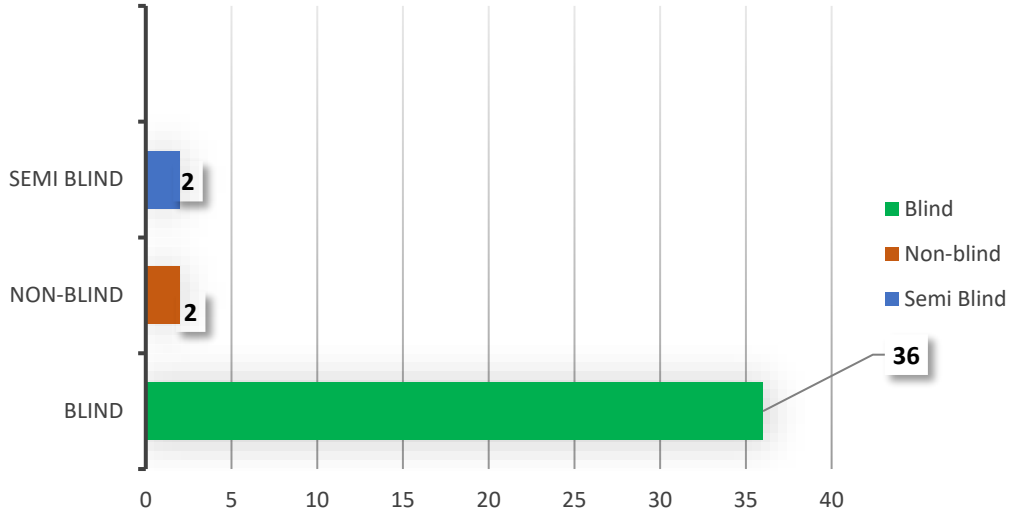


Figure 2. Representation of the requirement for the host signal to recover the watermark in reviewed works.

Additionally, although the figure treats them as part of the embedding and extraction processes, in some cases, the process may occur in only one stage—either embedding or extraction.

Unlike prior taxonomies that classify watermarking algorithms primarily by embedding domain (e.g., time, frequency) or technique (e.g., spread spectrum, echo hiding, patchwork), the stage-based taxonomy proposed in this work organizes the processes involved in each phase of the watermarking pipeline. This perspective facilitates cross-comparison of techniques that share similar processing mechanisms despite belonging to different conventional categories. Furthermore, this functional organization highlights how various classical approaches align within specific stages, providing a clearer understanding of their interactions and dependencies.

3.1. Audio Signal Preprocessing

The watermark message is embedded, modifying the host signal. Next, the message is recovered by detecting this modification. During the recovery stage, the watermarked signal must be processed to obtain the same representation employed in the embedding process. This representation is referred to as a domain in [4,5], and the following sections describe the domains considered.

Time Domain: Some works consider a time domain where the modification is performed directly on the samples; transforms can be applied, but the coefficients of the host signal are not altered. The works that utilize this domain to embed information are [9,20,28,33].

Transform Domain: When a transform is performed, the coefficients are altered. In that case, the inverse transform is necessary to obtain the watermarked signal in the same format as the host signal. Several transforms have been employed in the watermarking

process to achieve better performance. The transforms reported are summarized in the following domains:

- Wavelet Transform: According to [47], "wavelets provide a flexible basis for representing a signal that can be regarded as a generalization of Fourier analysis to non-stationary processes, or as a filter bank that can represent complex functions that might include abrupt changes in functional form or signals with time-varying frequency and amplitude." The use of this representation in different variations has been explored in different digital watermarking algorithms to decrease the trade-offs. The wavelet transforms include discrete wavelet transform (DWT), lifting as wavelet transform (LWT), stationary wavelet transform (SWT), dual-tree complex wavelet transform (DTCWT), and integer wavelet transform (IWT). Whether alone or in combination with other domains, the wavelet domain is the most frequently utilized in the most recently reported works. Approximately 60% of the reviewed works in this paper employ some variation of this domain.

  "Wavelet transforms decompose the signal into sub-bands characterized by two types of coefficients: approximation coefficients, which capture low-frequency components, and detail coefficients, which capture high-frequency components" [8]. The watermark is most commonly embedded in the approximation coefficients, likely because classical attacks—such as MP3 compression—primarily affect the high-frequency components. This decomposition can be performed at multiple levels; however, increasing the level of granularity raises computational complexity due to the additional signal processing required. Consequently, two-level decompositions are the most commonly employed in the reviewed works. Table 1 summarizes the variety of wavelets, the coefficients, and the decomposition level used.

  Table 1 presents the variety, coefficients, and level of wavelet used in the reviewed works.

Table 1. Variety, coefficients, and level of wavelet used in the reviewed works.

Work	Variety	Coefficients	Level
[6]	DWT	Approximation	6
[7]	DWT	Not reported	2
[8]	DWT	Not reported	4
[10]	DWT	Approximation	3
[11]	DWT	Approximation	2
[12]	DWT	Detail	5
[13]	DWT	Approximation	4
[14]	DWT	Approximation	5
[15]	DWT	Detail	3
[16]	DWT	Approximation	2
[17]	DWT	Not reported	2
[18]	LWT	Approximation	3
[19]	LWT	Approximation	3
[20]	SWT	Not reported	Not reported
[30]	DWT	Not reported	9–11
[32]	DWT	Not reported	Not reported
[33]	LWT	Not reported	Not reported
[35]	DTCWT	Approximation	3
[36]	DWT	Not reported	5–16
[37]	DWT	Approximation	2
[38]	DWT	Approximation	3

Table 1. *Cont.*

Work	Variety	Coefficients	Level
[39]	DTCWT	Detail	2
[40]	IWT	Approximation	2
[42]	DWT	Approximation	1
[43]	DWT	Detail	2

- Cosine Transform: The signal is transformed from the time domain to the frequency domain, like a sum of the cosine functions in various frequencies [31]. The discrete cosine transform (DCT) is employed in works [21–23,30–33,42,43,45]. A variant of DCT, called quantum discrete cosine transform (qDCT), uses quantum computing and quantum information processing and is the domain where the watermark is embedded [23].
- Matrix Decomposition: The signal or some representation of this is presented as a vector. This transformation converts the vector into a square matrix and employs a factorization technique to embed information about the values of one of the factorization elements. Among the reported factorization techniques are singular value decomposition (SVD) and LU decomposition.
    - SVD: Based on linear algebra, "SVD is a method of matrix decomposition where a rectangular matrix A can be broken down into the product of three matrices: a unitary matrix U, a diagonal matrix S, and a unitary matrix V" [7,31]. SVD can be represented as follows:

$$A = USV\hat{}T, \quad (1)$$

      where $V*V\hat{}T = I$, being I an identity matrix, while U and V are orthogonal matrices, and S is a diagonal matrix, the elements of which are called singular values [25]. SVD is utilized in the watermarking process of the following works [7,8,25,26,31–33,36,42,44,45]. The watermark is usually embedded in S; only [8] reports using U and S in the process.
    - LU decomposition: According to [22], "a square matrix A can be decomposed into the product of two matrices, L, a lower triangular matrix, and U, an upper triangular matrix." This method is used in [14,22].
- Fourier transform: The transform of the time domain signal samples to the frequency domain is performed through the Fast Fourier Transform (FFT) in works [27,34] and the Short-Time Fourier Transform (STFT) in [39].
- Spikegram: According to [29], the host signal is decomposed over a dictionary to render a sparse vector with only a few non-zero coefficients and add the watermark in this domain.
- The fractional Charlier moment transform (FrCMT): It is considered a generalization of the Charlier moment transform, where a signal function is projected on fractional Charlier polynomials [35].
- Graph-based transform (GBT): An edge-adaptive transform with efficient depth-map coding has been applied to audio compression [48] and was also employed in [38] for watermark embedding.
- Singular-spectrum analysis (SSA) is a technique used in [25] to identify and extract meaningful information by decomposing a signal into several additive oscillatory components.

The use of multiple domains is employed to leverage the performance of each domain. For example, as mentioned in [8], "the use of DWT and SVD provided with the perceptually transparent embedding of a large amount of data and good robustness, respectively." The methods regarding the domain of message embedding are summarized in Table 2.

Table 2. Domain where the watermark message is embedded.

Specific Domain	Works
DWT, SVD	[7,8,36,37]
DWT, DCT	[30,43]
DCT, SVD	[31,45]
DWT, DCT SVD	[32,42]
Time, SWT	[20]
Time, LWT, DCT, SVD	[33]
DTCWT, FRCMT	[35]
DWT, GBT, SVD	[38]
DTCWT, STFT, SVD	[38]
LWT, DCT, SVD	[40]

3.2. Embedding/Recovery Method

In this stage, the samples or coefficients in the final domain are modified to insert information, and the inverse process that recovers information from the watermarked signal is also considered. Unlike the change in domain, the process of embedding and recovery is usually different here. In this work, these processes are considered together. Due to their similarities, these processes are classified as quantization-based, least significant bit (LSB), additive, and others.

Quantization-based: The information is presented in binary form and embedded by quantifying the host signal differently for bit "0" or "1,". The extraction is usually performed when a threshold or a signal meets a specific condition. Quantization index modulation (QIM) [7,10,11,20,33,42] and its variation rational dither modulation (RDM) [13,19,30] are the most used techniques based on quantization.

According to [20], QIM is one of the most robust methods introduced in audio watermarking. It is simple and requires little computing time. According to the information presented in the reviewed works, a general QIM can be expressed for the following equation:

$$\hat{c} = \begin{cases} f\left(\frac{c}{\Delta} + \alpha\right)\Delta, \; if \; w = x \\ f\left(\frac{c}{\Delta}\right)\Delta + \frac{\Delta}{2}, \; if \; w = y \end{cases}, \tag{2}$$

where c represents the coefficient or some representation of this; Δ represents the step of quantization, which is usually set and must be known for extraction; α is a constant; w, x, and y are the binary information to embed; and $f(.)$ is a function that can be a floor or round function, which in [7,20,42] change with the value of x or y. In the process of recovering the message, the signal is usually quantized. After performing various operations that differ in each work, the watermark bit is selected based on a threshold. Only in [20] is the bit chosen to evaluate if the coefficient after quantization is even or odd.

The variation in RDM exhibits that the quantization step (Δ) is recursively derivable from previous coefficients. Therefore, this parameter is computed both in embedding and extraction. To control inaudibility when RDM is employed, the maximum tolerance level at the frequency corresponding to the coefficient is utilized in the calculation.

Quantization is used in other works as a tool to embed watermarks. Works [8,12,15,26,35,37] embed information by establishing quantified coefficients for even or odd numbers to differentiate between one and zero in binary information. The extraction is performed similarly, quantizing and evaluating whether it is even or odd. Work [14] adds a threshold to the quantized coefficient to embed a watermark. The threshold is chosen based on whether the binary information is one or zero. In [35], the energy of the coefficients is quantized, while in [12,15] the signal is quantized.

Works [22,31] are considered quantization-based, even when they do not use the parameter Δ; the coefficients are replaced by the nearest even or odd number (even if the binary information is 0, and odd if it is 1) in the Fibonacci sequence; because of that, it is possible to say that the Δ is variable and corresponds to the difference between two consecutive numbers in the Fibonacci sequence. The final domain in these works is matrix decomposition, where the change is applied to all matrix elements, and the extraction is performed by determining whether the majority of numbers in the matrix are even or odd.

Additive: One of the most commonly used methods for embedding watermarks is reported in several works [6,9,24,28,29,32,36,38–40,44]. According to [5], a generic additive watermarking model in the time domain can be expressed as shown in (3), where $x(n)$ represents the samples of the digital audio signal, and $w(n)$ represents the watermark in its native domain. At the same time, α is a factor that controls its presence in the watermarked signal. The same expression can be employed for the transformation domain in x (host signal) and w (watermark message).

$$y(n) = x(n) + \alpha w(n) \tag{3}$$

LSB: In this technique, the watermark is embedded into the least significant bits of the host signal. The LSB method is frequently used in image watermarking [3], as well as in audio watermarking in [16,34].

Other: The remaining works explore various methods for embedding information. In the continuation, a brief description of some selected methods is given.

In works [17,23,44], information is embedded in binary form, establishing a relationship between two elements. In work [17], the host signal is partitioned into two sub-frames through down-sampling, and the energy of the sub-frames is allocated according to the information to be embedded. Additionally, ref. [17] introduces a novel technique to reverse the embedded information using machine learning (ML). Although a set threshold can recover the information, it is not utilized in the final proposal. In [23], a technique is employed to embed the watermark by adjusting the difference in eigenvalues between two adjacent frequency bands. The watermark recovery also utilizes the difference in adjacent frequency bands. A stereo signal is used in [44], where the relation between the two channels embeds the watermark. This embedding incorporates two features: segmental singular values summation (SSVS) and segmental singular values difference (SSVD), which are used to extract the relation between these features.

According to [18], the shape of the sorted sequence gathered in each frame reflects a particular statistical distribution of the LWT coefficients. This distribution's shape is modified to embed the information, and the shape of this distribution on the receiver side gives the recovered watermark. In [21], the watermark coefficient is provided by the product of the average value of the coefficients in a frame and a tiny constant. This operation is only performed when the binary information is 1. And the bit is revered by a threshold. In work [25], the amplitude of specific oscillatory components is modified. For extraction, automatic parameter estimation is performed (to detect the singular values that were modified automatically), followed by polynomial fitting to enhance extraction. A

novel technique is presented in [27], where a frame is divided into four sub-frames. Each sub-frame is numbered, and two bits are embedded in each process. The energy of the sub-frame corresponding to the value of these two bits is then modified to the closest Lucas number. The energy of the other sub-frames is modified according to a series of conditions. The two bits are recovered, computing the difference in each sub-frame with the nearest Lucas number. In [43], the watermark is embedded in specific coefficients; the product of the watermark replaces these with a strong factor.

3.3. Watermark Process

Several works report the processing of the watermark message before the embedding process. According to the literature review, these processes can be grouped based on the objectives they aim to achieve, including methods for security, compression, error correction, and domain transformation.

Security: This process aims to protect access to message information from unauthorized individuals. The literature review found that one of the most frequently reported methods for this purpose is the use of chaotic maps [10,13,20,32,35,38]. Typically, the chaotic signal is converted to binary form using a threshold, and then the message is modified using an exclusive OR operator. When the watermark is determined to be an image, a commonly used method to scramble the bits is Arnold transform; this process is employed in [14,15,37,39,40,42,44]. In the case of works [18,30,41], bit scrambling is also reported, although the specific method used is not specified. Another approach is multiplying the signal by a pseudo-random sequence, as reported in [29]. The pseudo-random sequence possesses specific characteristics and can undergo spread spectrum modulation, which enhances security and aids in robustness; this method is employed by [6,9].

Compression: Works [33,37,44] reported using watermark message compression techniques to achieve a better embedding capacity. In [33], the image is compressed using compressive sampling. An adaptive scaling filter is used in [37]. A particular case is presented in [44], where medical images are embedded by utilizing wavelet-based image fusion to merge two images and embed them as a watermark.

Error correction: Bose–Chaudhuri–Hocquenghem coding (BCH) is reported in works [7,39,42] for error correction.

3.4. Adaptive Process

Various techniques presented for watermarking can be observed to involve parameters that adjust the functionality and impact of the watermark on different performance criteria. However, due to the significant differences between various host signals, other attacks, and different watermark messages, as well as how the process affects these factors, setting these parameters does not always result in uniform performance or guarantee minimum performance values.

Certain works present an adaptive process in which one or more parameters change in response to one or more inputs. The former typically occurs during embedding, although it can also happen during recovery, as seen in [6,9].

To differentiate the various methods used in this process, we categorize them based on their operating principles or the tools they utilize, resulting in six different groups, as shown in Figure 3.

The following sections briefly describe the functionality and how the different watermarking system proposals employ them.

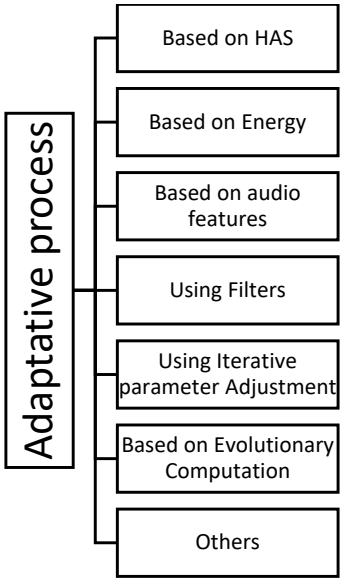


Figure 3. Classification of adaptive processes.

3.4.1. Adaptive Process Based on HAS

Since the physical characteristics of sound differ from how humans perceive and process it, several works consider this effect to maintain imperceptibility. As mentioned in [2], "HAS is responsible for how sound is perceived."

One of the most useful effects of the HAS is the so-called masking thresholds, in which the perception of a sound depends not only on its own characteristics but also on the presence of other sounds that may occur at nearby times and frequencies. In [2], this process is described as "the phenomenon that one faint but audible sound (the masked) becomes inaudible in the presence of another louder audible sound (the masker)." The embedding strength is adjusted by considering the masking thresholds in [6,9], which are divided into tonal and non-tonal masks. The study found better performance using only non-tonal masking. Similarly, on the recovery side, sub-band shaping and equalization are applied, taking into account the characteristics of the watermarked signal. This classification encompasses part of the embedding process that utilizes RDM [13,19,30], where the quantization step is recursively derived from previous coefficients and controls the maximum tolerable level of the human auditory system.

3.4.2. Adaptive Process Based on Energy

A promising method for dealing with adaptive watermarking is considering the energy present in the segments where the information is embedded. For example, segments with relatively high energy and very small silent periods are used for watermark embedding in [20]. Another method, as demonstrated in [10,11], involves enhancing inaudibility by applying specific coefficients to maintain energy balance after embedding. A different approach consists of selecting audio segments for embedding based on their energy levels. In [8], frames are ordered based on energy, and the information is embedded accordingly. In [23], the segmentation into small frames is performed to locate frames with high-energy voice signals, allowing the watermark to be embedded around them. In [28], energy similarity is used to cluster sparse subspaces and add the watermark message within these subspaces. In [10,11], a group of coefficients is reserved to maintain energy balance after embedding, thereby achieving high inaudibility.

3.4.3. Adaptive Process Based on Audio Features

To enhance performance in audio watermarking, the extraction of different features that identify the signal is applied, as classical frequency and time analysis are insufficient. According to [49], "Audio feature extraction is one of the cornerstones of current audio signal processing research and development. Audio features are contextual information that can be extracted from an audio signal." This information is utilized to enhance performance in audio watermarking.

In [12], an adaptive payload is proposed by considering audio features. By extracting the following audio features—pitch, formants, zero-crossing rate, energy, volume, silence ratio, centroid, roll-off, flux, short-time energy, and skewness—principal component analysis (PCA) is performed to determine that energy, short-time energy, and zero-crossing rate are the most relevant features for the payload. These features are then used to determine the payload during the embedding process.

To determine the best frames for embedding, ref. [31] utilizes a fuzzy system that provides the feasibility of watermark embedding with five degrees (bad, not bad, average, good, excellent). The inputs to the fuzzy system are three audio features: short-time energy, zero-crossing factor, and music edges.

3.4.4. Adaptive Process Using Filters

One of the tools used to adapt parameters in the embedding and recovering of the watermark message is filtering. A least squares Savitzky–Golay smoothing filter is used in reception in [6] to reduce message distortion caused by variations in the host signal. A two-tap FIR low-pass filter (LPF) is employed in [13] to minimize modulation noise. In [10], improving inaudibility and embedding capacity are achieved by utilizing a pair of spectral shaping (SS) filters to predict the audio signal envelope and aggregate under high-energy spectral peaks. The SS filters verify that all vectors possess the correct norms for effective extraction and that they satisfy a given threshold.

3.4.5. Adaptive Process Using Iterative Parameter Adjustment

This process establishes specific requirements, and embedding is achieved through an iterative parameter adjustment process that meets these requirements.

In [27], an embedding rate is set, and compliance with the signal-to-noise ratio (SNR), objective difference grade (ODG), and bit error rate (BER) requirements is evaluated. At each value, the frame size, frequency band, and Δ (the minimum energy difference in each sub-frame with respect to the Lucas sequence) are adjusted.

In [32], an SNR-constrained heuristic search method is proposed for optimizing the scaling parameter. This method sets a constraint on the SNR and iteratively searches for the scaling parameter. Embedding and extraction are performed, and some attacks are tested to ensure the SNR constraint is met and the message is correctly extracted, measured by correlation with the original message.

Aiming at payload, robustness, and transparency, ref. [37] proposes an iterative system for adjusting the watermark message scaling factor, frame size, and embedding strength parameters. This work concludes that the scaling factor mainly affects the payload and BER and is correlated with the frame size, with little impact on SNR. Conversely, the embedding strength primarily affects the SNR.

3.4.6. Adaptive Process Based on Evolutionary Computation (EC)

The adaptive process can be seen as an optimization problem, and as mentioned in [50], EC has been proposed to solve complex continuous optimization problems. In [14], a genetic algorithm is employed to determine the optimal number of samples per frame

used to embed each watermarking bit reliably, in order to determine the best DCT sub-band for embedding information and the frame length. In [22], a genetic algorithm is employed to achieve optimal SNR and embedding capacity. In [25], the differential evolution search method optimizes the quality of the watermarked signal by iteratively improving candidate parameters.

3.4.7. Others

Other processes used in the proposed work, as identified in the literature review, that do not fit into the proposed categories include those presented in [7,44]. In [7], the generator polynomial of BCH is adaptive. Although the procedure for varying this parameter is not specified, it is known that different polynomials provide different error correction capacities and compression ratios, which suggests that this will depend on the characteristics of the host signal. In [44], based on the relationship between the singular values of the right and left channels, SSVS is used as a parameter to generate adaptive embedding.

3.5. Auxiliary Signals

In addition to the watermark signal, some proposed works suggest embedding extra information. This information is typically embedded to enhance robustness. We can classify auxiliary signals based on their purpose: synchronization, reconstruction, and assistance in extraction. The following paragraphs provide more detail on these applications.

3.5.1. Synchronization

The works [13,18–20,30,32,37,42,43] that use synchronization methods demonstrate improved performance against advanced attacks.

In [13], a synchronization code is interwoven with the watermark. An auxiliary signal is embedded in the time domain [18,20,33]. In [18], a sinusoidal signal is embedded. In [20], a signal is embedded at the beginning and end of each frame using QIM. The spread spectrum technique is employed in [33] to embed synchronization bits. The 11-level approximation coefficients are used to embed a sinusoidal signal in [30], where the zero crossings of this signal are taken as synchronization points. In [19], the three-level detail coefficients are used for the synchronization code via RDM; additionally, another auxiliary signal is embedded. For synchronization in [29], a Baker sequence is embedded in the first 13 bits of each second of the host signal. A 512-bit code created from an SHA-512 function is inserted into the S-values in the SVD transform using QIM in [42]. An eight-bit code is embedded before each watermark bit in [43]. A part of the host signal is considered for inserting the synchronization signal in [32,37], though the method is not described in detail.

3.5.2. Reconstruction

This technique proposes a solution to manipulating audio signals, where a signal containing information from the original audio is embedded to reconstruct deleted or altered parts. In [19], a highly compressed version of the host signal is embedded as fragile watermarking in the second- and third-level detail sub-bands using AQIM. This category [43] is included, where only a compressed version of the host signal is embedded as the watermark using compressed sensing in the DCT domain.

3.5.3. Assistance in Extraction

In [17], another function of an auxiliary signal is proposed. In this case, bits created by a linear-feedback shift register (LFSR) are added to the signal and used in extraction to train ML algorithms, specifically SVM (rbf kernel, quadratic kernel) and KNN. From the results of the three algorithms, the bit with the majority vote is taken as the recovered

bit. According to the report, this increases robustness since there is no fixed threshold for deciding the value of the recovered bit, as in most proposals.

4. Classification According to Performance Criterion

This section presents the reported results regarding the performance criteria: imperceptibility, capacity, security, and computational complexity.

4.1. Imperceptibility

Imperceptibility is a subjective characteristic; however, there are reported methods to obtain measurable values. The most used methods in the reported works here are signal-to-noise ratio (SNR) and ITU-R recommendations [51,52].

SNR is an objective measure of imperceptibility, which only considers the difference between signals with and without watermarks.

The authors in [35] use a psychoacoustic model to obtain a grade of imperceptibility through a computational approach. Table 2 shows the ODG, the scale, and the corresponding subjective values.

In [36], a procedure is proposed for evaluating imperceptibility in persons. This procedure uses the "double-blind, triple-stimuli with hidden reference" approach. The standard requires 20 subjects to obtain the subjective grade of difference (SDG). The scale is shown in Table 3.

Table 3. Scales for ITU-R BS.1387 [51] and ITU-R BS.1116 [52].

Quality	Impairment	ODG	Quality
Excellent	Imperceptible	0	5
Good	Perceptible but not annoying	−1	4
Fair	Slightly annoying	−2	3
Poor	Annoying	−3	2
Bad	Very annoying	−4	1

The literature employs two additional methods to evaluate imperceptibility through subjective tests. In works [8,23,43], the imperceptibility is measured through the mean opinion score (MOS), described in [53], and recommended to evaluate telephone transmission quality. The ABX test is used in works [25,28] to measure imperceptibility. According to [25], the test compares two choices to identify if a listener can perceive a difference. Due to SDG, the subject gives a grade of the difference between the original signal and the watermarked signal, and the scope of the recommendation is focused on general audio. This work considers this method to be more convenient for evaluating imperceptibility subjectively. As indicated in [17], the minimum value of SNR for acceptable performance is 20 dB. According to Table 3, to have a "good audio quality," the values of ODG and SDG should be higher than -1 and 4, respectively. Therefore, these values are considered a threshold for subgrouping the works according to imperceptibility, with "adequate imperceptibility" and "poor imperceptibility." The methods are grouped and presented in Table 4. It is essential to note that the values depend on the host signal. Consequently, this work shows an average value when the works report imperceptibility with several host signals.

Observing Table 4, we can see that the SNR is the most commonly used method for measuring imperceptibility. The former was employed in other areas, such as communications. However, not considering the HAS in evaluations may not provide an accurate measure of imperceptibility performance. It is recommended to evaluate imperceptibility using the three methods; however, all three approaches are not always available due to the time and equipment required to use them.

Table 4. Scales for ITU-R BS.1387 [51] and ITU-R BS.1116 [52].

Method	Threshold	Work
SNR	Not reported	[7,9,12,25,26,29,36,45]
	<20 dB	[10,13,18,19,30]
	≥20 dB	[6,8,11,12,14–17,20–25,27,28,31,32,34,35,37–44]
ODG	Not reported	[7,8,11,12,14,15,20,24,34–37,39,41–44]
	≥−1	[6–8,13,16–19,21–23,25–33,38,40,45]
SDG	Not reported	[6–11,13,14,17–19,23–25,27–39,41–45]
	≥4	[12,15,16,20–22,26,40]

4.2. Capacity

Regularly, the watermark message is digital information expressed in bits. The number of bits will be higher if the duration of the host signal is longer. That is why the best form to report the capacity is the ratio of bits in a determined time. The capacity is usually presented in bits per second (bps). According to [27], it is defined as follows:

$$Capacity = \frac{W}{N} \text{bps}, \tag{4}$$

where W is the number of bits that comprise the watermark message, and N is the length in seconds of the audio signal before the watermarking process.

There are works whose primary goal is to achieve a high capacity. However, in some applications, a high capacity is optional. According to [17], the minimum capacity of a watermarking algorithm should be 20 bps. Authors in [14,15,22] describe their algorithm as "high capacity", with values above 1000 bps. Hence, we propose here three groups to classify the proposed algorithms according to their capacity: low capacity below 20 bps, medium capacity from 20 bps to 1000, and high capacity above 1000 bps. The classification of approaches regarding capacity is shown in Table 5. We can observe that the reported capacity is variable in [27–29] because it depends on the host signal characteristics. Most works report a medium capacity; however, works such as [8,15,37,40] achieve a capacity more than twice the threshold set for high capacity.

Table 5. Reported capacity of the reviewed algorithms in bps.

Group	Work
Low Capacity	[25,28]
Medium Capacity	[6,9–13,16,20,21,23,26,29–31,33,35,41,42,45]
High Capacity	[8,14,15,17,19,22,27,37,39,40]
Not Reported	[7,24,32,34,36,38,43,44]

4.3. Security

Works presented in the literature usually do not report evaluations of security and its quality. In the reviewed literature, only [29,35,44] detail tests for security. A cepstrum analysis is performed in [29] to prove that the existence of a watermark cannot be detected. In [35], using the wrong key, the recovered watermark has a BER value of about 0.5. The authors in [44] proposed a false-positive-free SVD-based audio watermarking algorithm that addresses the issue of obtaining a fake watermark at the receptor by using counterfeit values during the extraction process. A BER value is not reported; however, a test is

conducted, which reveals that the watermark cannot be altered with counterfeit values in the receptor.

Some authors have reported using keys to extract the message, ensuring a certain degree of security. The most commonly reported keys are pseudo-noise (PN) sequences, chaotic codification, scrambling bits, the Arnold transform, and parameters used in the embedded process that are required for extraction. In the case of [41], it is mentioned that two methods are employed as keys, scrabbling the message through Fibonacci–Lucas transformation (FLT) and using the parameter quantization threshold as a key. Reviewed works are classified according to the necessity of one or more keys to recover the message, as shown in Table 6. It can be observed that seventeen works do not report that some key is necessary to embed and recover the watermark message. Scramble bits and the Arnold transform are used more as a key; however, the second one is only utilized when the watermark message is an image.

Table 6. Methods used to give security in a watermarking algorithm.

Group	Method	Work
No key reported	Not Apply	[7,8,12,16,17,21–25,27,30,31,33,34,36,45]
Reported Key	PN	[6,9,28,29]
	Scramble Bits	[13,18,19,26,41,43]
	Chaotic Maps	[10,20,32,38,42]
	Arnold Transform	[14,15,37,39,40,44]
	Parameters	[25,35,41]

4.4. Computational Complexity

Computational complexity is essential if the algorithm will be implemented by hardware with limited resources or real-time applications. Usually, computational complexity is presented as a time of embedding or extraction; the processing time depends on the hardware characteristics that are also given. The evaluations are generally performed by simulating the processes in software. However, this criterion is only sometimes included in evaluating the proposed algorithms. Table 7 presents which works reported the results of a computational complexity test; only 25% of reviewed works report computational complexity. The description of the test and results are given in the following:

Table 7. Works grouped according to computational complexity test.

Group	Work
No computational complexity report	[6–16,19–22,24–26,30,31,33–36,38,39,41–44]
Computational complexity report	[17,18,23,27–29,32,37,40,45]

In [29], the time of embedding or extraction is not mentioned. Still, it is reported that it was tested for real-time decoding in a personal computer with an Intel CPU at a frequency of 2.5 GHz and DDR memory of 512 MB using MATLAB-7.

An average time of 0.93 s is reported in [45]. The time of embedding or extraction is not specified. The test was performed in MATLAB2018b on a Windows 10 laptop with 16 GB RAM and an IntelCore-i7-8650U2.11GHz processor.

Authors in [37] reported an average time of 3.6 s in the embedding process and 0.6 s in the extraction process in a host signal of 20 s length. Details of the hardware and software of the test were not mentioned.

Authors in [32] compare the running time with three methods using four signals of different lengths. The average time for a 30 s length is 529.13, 550.25 s for a 60 s length, 578.37 for a 120 s length, and 629.446 for a 180 s length.

In [18], the simulation was performed in MATLAB, (where the software version is not reported), on an Intel I7-4790 CPU with 32G RAM, obtaining an embedding time of 1.84 s and an extraction time of 0.92 s.

Using an Intel Core-i5 CPU at a frequency of 2.4 GHz and DDR3 memory of 4096 MB with MATLAB 2012 [27] reports an average embedding time of 1.46 and 0.89 s extracting time.

To perform the test, ref. [28] used an Intel Core-i7-6700 CPU at a frequency of 2.6 GHz and 8 GB RAM; the embedding time varied from 0.237 to 7.837 s, while the extraction time varied from 0.414 to 7.832 s.

An average embedding time of 1.03 s and an average extracting time of 6.88 s is reported by [17] using a personal computer with an Intel Core-i5 CPU at a frequency of 2.4 GHz and DDR3 memory of 4096 MB using an MATLAB-2012.

In [23], the average embedding time is 1.2612 s, and the average extracting time is 0.6012 s.

Only [29] reports that the algorithm can work in real-time and it is only reported in the recovery of the watermark. In [40], the authors mention that "there is a limitation to exploring the proposed audio watermarking algorithm in real-time applications due to its computational complexity". The shortest time reported in this work is 1.0632 s for 10 s length of the host signal; taking this threshold, the algorithms with longer times are not also viable for real-time applications. It is possible to observe that only [28] reports times below the second in the embedding process, although it can be inferred that the embedding process in [45] is also below because the whole process takes less than 1 s. In the extraction, refs. [21,28,31,37,45] reported times below 1 s. It can be deduced that the processes to embed the watermark message have more computational complexity than to extract it. It is challenging to apply it in a real-time situation on the transmission side, which is an area of research for application where this performance is required.

4.5. Robustness

One of the advantages of watermarking is that the signal can be shared without restriction. Still, this can cause the watermarked signal to suffer different attacks, which can be intentional or unintentional. A watermarking algorithm should guarantee the recovery of the watermark message after any attack. This section first presents a novel classification of known attacks and then a classification of robustness presented by the algorithms.

Attacks

Watermarked signals can suffer a substantial quantity of attacks, and since each attack can be made with different parameters, it is challenging to define a general degree of robustness. Instead, the works subject the watermarked signal to some defined attacks. The attacks reported in the literature are as follows:

(a) Resampling: The sampling frequency is changed; it usually goes from 44.1 kHz to another frequency, and later, to recover the message, it is returned to 44.1 kHz. The frequencies reported in the resampling attack are the following: 11.0025 kHz, 16 kHz, 22.05 kHz, 36 kHz, and 48 kHz.

(b) Requantization: The bits used in quantization are changed; usually, the host signal uses 16 bits of quantization and reported attacks of re-quantization change for eight or twenty-four bits.

(c) Low-pass filter (LPF): Filter the signal with an LPF with a given cut frequency. Reported cut frequency values are 3.5, 4, 6, 8, 9, 10, 11, 15, and 20 kHz. The filter will impact a wider bandwidth if the cutoff frequency is lower. Consequently, the robustness is affected because the filter can affect frequencies where the watermark is embedded.
(d) Noise corruption: Adding noise is usually zero-mean Gaussian noise with a given SNR. 15, 20, 30, and 35 dB are commonly reported SNR in tests.
(e) MPGE-1 layer three compression: Compressing and decompressing an audio watermarked signal at a given bit rate, where the bit rates reported to test robustness are 32, 64, 128, 192, and 256 kbps.
(f) Cropping: Some samples are removed; the attack is described with the number of samples removed or the percentage removed from the host signal.
(g) Amplitude scaling: Scaling the amplitude by a factor.
(h) Pitch scaling: Increase or decrease the pitch by a percentage.
(i) DA/AD conversion: Convert the digital audio file to an analog signal and convert it to a digital signal.
(j) Echo addition: Adding an echo signal with a time delay and a percentage of decay.
(k) Jittering: Deleting or adding one sample for every given number of samples.
(l) Bandpass filter (BPF): Filter the signal with a BPF with a pair of given cut frequencies; as mentioned in c, the cutoff frequencies will influence the affected bandwidth and consequently, the aggressiveness of the attack.
(m) Mp4 conversion: Compressing a decompressing audio watermarked signal through Mp4 at a given bit rate, also known as AAC conversion.
(n) Equalizer: Increasing and decreasing the power spectral density in some frequencies.
(o) Brumm addition: Adding a sinus tone of a given frequency with an amplitude of a certain factor of the maximum dynamic range.
(p) Zero adding: Zeroing out all samples below a threshold.
(q) Time shift: Shifting some samples of the audio signal.
(r) High pass filter (HPF): Filter the signal with an HPF with a given cut frequency.
(s) De-noising: Processing the audio signal to quiet sounds with a characteristic noise spectral.
(t) Reverberation: Add a reverberation that is multiple smoothed copies of the audio signal in a set time.
(u) Pitch-invariant time-scale modification: The time is changed, whereas the audio pitch is preserved, having a longer duration with a slower tempo or a shorter duration with a faster tempo.
(v) Collusion: Different watermarks are embedded in copies of the host signal, and the watermarked signal is an average of the watermarked copies.
(w) Multiple watermarking: The host signal is sequentially watermarked with different messages.
(x) Speed scaling attack: Pitch and time are modified at the same time.
(y) Mask attack: Delete information under the masking threshold.
(z) Replacement: Taking advantage of similitudes in the same host signal and replacing these with a similar part of the same signal.

The authors in [2] proposed testing robustness with two groups of attacks: the basic robustness test and the advanced robustness test.

The basic robustness test is compounded by the attacks: a, b, c, d, e, f, g, h, i, j, k, p, t, and u.

The advance robustness test includes all the attacks in the basic robustness test, adding v and w.

Another classification is proposed in [5] with the following structure:

Basic attacks are a set of attacks conformed by b, d, g, e, j, and another attack denoted as filtering, which c, l, and r may compound.

Advanced attacks include a subgroup called desynchronization attacks conformed by f, h, k, q, u, x, y, and z.

In [51], the attacks are classified into eight groups depending on how the signal is manipulated: dynamics, filter, ambiance, conversion, lossy compression, noise, modulation, time stretch, pitch shift, and sample permutation.

In some applications, the effect of attacks over the watermark message is used to have information about manipulating the audio signal. Among the 41 algorithms revised in this work, only a few of them mentioned this characteristic: the method in [11] is fragile, and in [19], a robust and fragile message is embedded, while [43] was mentioned to be semi-fragile. Additionally, since each attack can be made with different parameters and new attacks can be created, it takes work to guarantee a general grade of robustness. Instead, the robustness is defined regarding some defined attacks. After extracting the watermark message, BER is reported to give an idea of the robustness. Some works document the normalized correlation coefficient (NCC) when the watermark message is an image.

According to [52], a resultant BER value below 0.2 will satisfy most watermarking applications' requirements. Therefore, we propose here to categorize the robustness of each algorithm into "fully robust" for algorithms that have reported BER values of 0, "robust" for BER values greater than 0 and less than 0.2, "not robust" for BER values exceeding 0.2, and "not reported" when the robustness has not been tested for the attack. The reported results are shown in Table 8. Watermarking and psychoacoustic attacks were not reported in the reviewed algorithms. Some works, such as [22,34,36,40,44], do not report the BER value; instead, they inform the NCC, so this work considers 20% to have failed to group the algorithms.

Table 8. Reported robustness to filtering attacks.

Attack	Category	Work
a	Fully Robust	[6,9,10,12–14,17,19–21,26,29,33,35,37–39,41,42,45]
a	Robust	[8,16,23,25,27,34,36,40]
a	Not Robust	[24,43]
a	Not Reported	[7,11,15,18,22,28,30–32,44]
b	Fully Robust	[6,9–11,13], [16] *, [17,19–21,23,29,35,37,38,41]
b	Robust	[8,12], [16] *, [18,26,27,34,36,40,45]
b	Not Reported	[7,14,15,22,24,25,28,30–33,39,42–44]
c	Fully Robust	[6,9,10,12,13,17–20], [23] *, [29,30,33], [35] *, [37], [42] *, [45]
c	Robust	[14,16], [23] *, [27,34], [35] *, [38,40,41], [42] *
c	Not Robust	[24,43]
c	Not Reported	[7,8,11,15,21,22,25,26,28,31,32,36,39,44]
d	Fully Robust	[6,9,14,17,21,26,29], [35] *, [37] *, [38,39,42]
d	Robust	[8–10,12,13,16,18–20,24,25,27,30,34], [35] *, [37] *, [36,40,41,44,45]
d	Not Reported	[7,11,15,22,23,31–33,43]
e	Fully Robust	[6,12,17,29], [35] *, [37,39,41,45]
e	Robust	[8–10,13–16,18,19,21,23,25–27,30], [35] *, [38], [43] *

Table 8. *Cont.*

Attack	Category	Work
e	Not Robust	[20,24,43]
e	Not reported	[7,11,20,22,24,28,31–34,36,40,42,44]
f	Fully Robust	[10,13,15,17,29], [35] *, [38,41]
f	Robust	[23,30], [35] *, [37,39,42]
f	Not Robust	[44]
f	Not reported	[6–9,11,12,14,16,18–22,24–28,31–34,36,40,43,45]
g	Fully Robust	[9,10,13,18–20,23,29,35,37,38,45]
g	Not reported	[6–8,11,12,14–17,21,22,24–28,30–33,36,39–44]
h	Fully Robust	[8,13,20,29]
h	Robust	[45]
h	Not Robust	[38]
h	Not reported	[6,7,9–12,14–19,21–28,30–37,39–44]
i	Fully Robust	[39]
i	Robust	[10,13,18,19,30]
i	Not reported	[6–9,11,12,14–17,20–29,31–45]
j	Fully Robust	[23] *, [35,39]
j	Robust	[10,13,16,18–21], [23] *, [36,41,42,45]
j	Not reported	[6–9,11,12,14,15,17,22,24–34,37,38,40,43,44]
k	Fully Robust	[15,20]
k	Robust	[10,13,18,19,23,30,45]
k	Not reported	[6–9,11,12,14,16,17,21,22,24–29,31–44]
l	Fully Robust	[25,28,39]
l	Not reported	[6–24,26,27,29–38,40–45]
m	Fully Robust	[9,25,39,45]
m	Not reported	[6–8,10–24,26–38,40–44]
n	Fully Robust	[8,16]
n	Not reported	[6,7,9–15,17–34,36–45]
o	Fully Robust	[13]
o	Not reported	[6–12,14–45]
p	Fully Robust	[13,19,20]
p	Not reported	[6–12,14–18,21–45]
q	Fully Robust	[13,18]
q	Robust	[35]
q	Not reported	[6–12,14–17,19–34,36–45]
r	Fully Robust	[7,20]
r	Not reported	[6,8–19,21–45]
s	Fully Robust	[28]
s	Robust	[41]

Table 8. *Cont.*

Attack	Category	Work
s	Not reported	[6–27,29–40,42–45]
t	Fully Robust	[28]
t	Not reported	[6–27,29–45]

* The attack was tested with different parameters, giving a different grade of robustness for each parameter.

It is possible to observe that few works test most of the attacks. These works cannot be classified as robust due to the lack of proof against several attacks.

It is essential to mention that works reported as fragile do not have the primary goal of resisting attacks. The authors in [11] do not report a grade of robustness to any attack because the algorithm aims to detect tampering. The method [43] is reported as not robust in almost all attacks tested because its application is to recover original audio from tampering. As a difference, the authors in [19] reported robustness because another robust watermark is embedded besides a fragile watermark for detecting and recovering tampering.

5. Conclusions

This paper reviews and classifies conventional audio watermarking approaches to help identify methods that satisfy the requirements of specific applications or determine whether certain methods could be improved. This review includes a newly proposed approach to watermarking, grouping the techniques into processes performed during both embedding and extraction. The division into processes facilitates the identification of new design trends and incorporates or enhances processes in new proposals. In the audio signal process that describes where the watermark is embedded, we can see that although the wavelet transform is used with above-average performance in most of the presented works, some new domains, such as spikegram, have reported promising results. Combining different domains and tools proposed in the literature may be an exciting area to explore.

In the embedding/recovery process, we observe that QIM or its variants, along with additive processes, are the most commonly used techniques in the proposed works. One reason is their ease of implementation, and when combined with different methods, they report good robustness values.

For the watermarking process, there is a clear trend toward increasing security by encoding the message before embedding. Compression and error correction processes can help reduce trade-offs.

There is a tendency to include an adaptive process in new proposed works because testing shows that results can vary significantly with fixed parameters, making it difficult to guarantee the system's proper functioning regardless of the host signal.

Including auxiliary signals has demonstrated increased robustness, especially against desynchronization and tampering attacks, which were not always considered in previous works. It is essential to highlight the results of this process and their potential inclusion in future studies, given the increasing frequency of this type of attack. The methods are classified according to the thresholds mentioned in the revised works to help developers adequately measure imperceptibility.

Regarding capacity, the terms 'low' and 'high' could be ambiguous, as recent research has largely surpassed the capacities reported in works from the last decade. That is why the ranges proposed in this study provide a practical solution.

In recent years, there has been a noticeable increase in the development of new methods and techniques. This growth is driven not only by the traditional goal of protecting

copyright and preventing unauthorized use of audio content, but also by the emergence of new forms of misuse, such as the application of generative AI. These developments demand more stringent control and monitoring of audio signals to ensure authenticity, traceability, and rightful ownership.

Recent studies have observed the use of new artificial intelligence tools, such as EC (evolutionary computation) and ML (machine learning). In future work, we will consider the reviews and classifications of AI audio watermarking schemes, as well as their comparisons with conventional approaches.

Author Contributions: Conceptualization, C.J.S.-C., and G.J.D.; methodology, C.J.S.-C.; formal analysis, C.J.S.-C.; investigation, C.J.S.-C.; resources, C.J.S.-C.; writing—original draft preparation, C.J.S.-C.; writing—review and editing, G.J.D.; visualization, C.J.S.-C.; supervision, G.J.D.; project administration, C.J.S.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors gratefully acknowledge the support of the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) for the realization of this work.

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

References

1. Nematollahi, M.A.; Vorakulpipat, C.; Rosales, H.G. *Digital Watermarking: Techniques and Trends*, 1st ed.; Springer: Singapore, 2016.
2. Lin, Y.T.; Abdulla, W.H. *Audio Watermark; A Comprehensive Foundation Using MATLAB: A Comprehensive Foundation Using MATLAB*, 2015th ed.; Springer International Publishing: Berlin/Heidelberg, Germany, 2014.
3. Shelke, R.D.; Nemade, M.U. Audio watermarking techniques for copyright protection: A review. In Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016.
4. Bajpai, J.; Kaur, A. A literature survey—Various audio watermarking techniques and their challenges. In Proceedings of the 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence), Noida, India, 14–15 January 2016.
5. Hua, G.; Huang, J.; Shi, Y.Q.; Goh, J.; Thing, V.L. Twenty years of digital audio watermarking—A comprehensive review. *Signal Process.* **2016**, *128*, 222–242. [CrossRef]
6. Attari, A.A.; Shirazi, A.A.B. Robust and transparent audio watermarking based on spread spectrum in wavelet domain. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019.
7. Safitri, I.; Ginanjar, R.R. Multilevel adaptive wavelet BCH code method for copyright protection in audio watermarking system. In Proceedings of the 2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, Indonesia, 28–29 November 2017.
8. Kaur, A.; Dutta, M.K. A blind watermarking algorithm for audio signals in multi-resolution and singular value decomposition. In Proceedings of the 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2018.
9. Li, R.; Xu, S.; Yang, H. Spread spectrum audio watermarking based on perceptual characteristic aware extraction. *IET Signal Process.* **2016**, *10*, 266–273. [CrossRef]
10. Hu, H.-T.; Hsu, L.-Y. Incorporating spectral shaping filtering into DWT-based vector modulation to improve blind audio watermarking. *Wirel. Pers. Commun.* **2017**, *94*, 221–240. [CrossRef]
11. Renza, D.; Lemus, C. Authenticity verification of audio signals based on fragile watermarking for audio forensics. *Expert Syst. Appl.* **2018**, *91*, 211–222. [CrossRef]
12. Kaur, A.; Dutta, M.K.; Soni, K.; Taneja, N. Localized & self adaptive audio watermarking algorithm in the wavelet domain. *J. Inf. Secur. Appl.* **2017**, *33*, 1–15. [CrossRef]
13. Hu, H.-T.; Hsu, L.-Y. Supplementary schemes to enhance the performance of DWT-RDM-based blind audio watermarking. *Circuits Syst. Signal Process.* **2017**, *36*, 1890–1911. [CrossRef]
14. Kaur, A.; Dutta, M.K. An optimized high payload audio watermarking algorithm based on LU-factorization. *Multimed. Syst.* **2018**, *24*, 341–353. [CrossRef]
15. Kaur, A.; Dutta, M.K. High embedding capacity and robust audio watermarking for secure transmission using tamper detection. *ETRI J.* **2018**, *40*, 133–145. [CrossRef]

16. Karajeh, H.; Khatib, T.; Rajab, L.; Maqableh, M. A robust digital audio watermarking scheme based on DWT and Schur decomposition. *Multimed. Tools Appl.* **2019**, *78*, 18395–18418. [CrossRef]
17. Pourhashemi, S.M.; Mosleh, M.; Erfani, Y. A novel audio watermarking scheme using ensemble-based watermark detector and discrete wavelet transform. *Neural Comput. Appl.* **2021**, *33*, 6161–6181. [CrossRef]
18. Hu, H.-T.; Chang, J.-R.; Lin, S.-J. Synchronous blind audio watermarking via shape configuration of sorted LWT coefficient magnitudes. *Signal Process.* **2018**, *147*, 190–202. [CrossRef]
19. Hu, H.-T.; Lee, T.-T. Hybrid blind audio watermarking for proprietary protection, tamper proofing, and self-recovery. *IEEE Access* **2019**, *7*, 180395–180408. [CrossRef]
20. Mushgil, B.M.; Adnan, W.A.W.; Al-hadad, S.A.-R.; Ahmad, S.M.S. An efficient selective method for audio watermarking against DE-synchronization attacks. *J. Electr. Eng. Technol.* **2018**, *13*, 476–484.
21. Jeyhoon, M.; Asgari, M.; Ehsan, L.; Jalilzadeh, S.Z. Blind audio watermarking algorithm based on DCT, linear regression and standard deviation. *Multimed. Tools Appl.* **2017**, *76*, 3343–3359. [CrossRef]
22. Mosleh, M.; Setayeshi, S.; Barekatain, B.; Mosleh, M. High-capacity, transparent and robust audio watermarking based on synergy between DCT transform and LU decomposition using genetic algorithm. *Analog. Integr. Circuits Signal Process.* **2019**, *100*, 513–525. [CrossRef]
23. Wu, Q.; Ding, R.; Wei, J. Audio watermarking algorithm with a synchronization mechanism based on spectrum distribution. *Secur. Commun. Netw.* **2022**, *2022*, 1–13. [CrossRef]
24. Chen, K.; Yan, F.; Iliyasu, A.M.; Zhao, J. Dual quantum audio watermarking schemes based on quantum discrete cosine transform. *Int. J. Theor. Phys.* **2019**, *58*, 502–521. [CrossRef]
25. Karnjana, J.; Unoki, M.; Aimmanee, P.; Wutiwiwatchai, C. Singular-spectrum analysis for digital audio watermarking with automatic parameterization and parameter estimation. *IEICE Trans. Inf. Syst.* **2016**, *E99.D*, 2109–2120. [CrossRef]
26. Bhat, K.V.; Das, A.K.; Lee, J.-H. Design of a blind quantization-based audio watermarking scheme using singular value decomposition. *Concurr. Comput.* **2020**, *32*, e5253. [CrossRef]
27. Pourhashemi, S.M.; Mosleh, M.; Erfani, Y. Audio watermarking based on synergy between Lucas regular sequence and Fast Fourier Transform. *Multimed. Tools Appl.* **2019**, *78*, 22883–22908. [CrossRef]
28. Wang, S.; Yuan, W.; Unoki, M. Multi-subspace echo hiding based on time-frequency similarities of audio signals. *IEEE ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2349–2363. [CrossRef]
29. Erfani, Y.; Pichevar, R.; Rouat, J. Audio watermarking using spikegram and a two-dictionary approach. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 840–852. [CrossRef]
30. Hu, H.-T.; Chang, J.-R. Efficient and robust frame-synchronized blind audio watermarking by featuring multilevel DWT and DCT. *Clust. Comput.* **2017**, *20*, 805–816. [CrossRef]
31. Mosleh, M.; Setayeshi, S.; Barekatain, B.; Mosleh, M. A novel audio watermarking scheme based on fuzzy inference system in DCT domain. *Multimed. Tools Appl.* **2021**, *80*, 20423–20447. [CrossRef]
32. Su, Z.; Zhang, G.; Yue, F.; Chang, L.; Jiang, J.; Yao, X. SNR-constrained heuristics for optimizing the scaling parameter of robust audio watermarking. *IEEE Trans. Multimed.* **2018**, *20*, 2631–2644. [CrossRef]
33. Novamizanti, L.; Budiman, G.; Astuti, E.N.F. Robust audio watermarking based on transform domain and SVD with compressive sampling framework. *TELKOMNIKA* **2020**, *18*, 1079. [CrossRef]
34. Salah, E.; Amine, K.; Redouane, K.; Fares, K. A Fourier transform based audio watermarking algorithm. *Appl. Acoust.* **2021**, *172*, 107652. [CrossRef]
35. Yamni, M.; Karmouni, H.; Sayyouri, M.; Qjidaa, H. Robust audio watermarking scheme based on fractional charlier moment transform and dual tree complex wavelet transform. *Expert Syst. Appl.* **2022**, *203*, 117325. [CrossRef]
36. Singha, A.; Ullah, M.A. Development of an audio watermarking with decentralization of the watermarks. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *34*, 3055–3061. [CrossRef]
37. Elshazly, A.R.; Nasr, M.E.; Fouad, M.M.; Abdel-Samie, F.E. Abdel-Samie. Intelligent high payload audio watermarking algorithm using colour image in DWT-SVD domain. *J. Phys. Conf. Ser.* **2021**, *2128*, 012019. [CrossRef]
38. Zhang, G.; Zheng, L.; Su, Z.; Zeng, Y.; Wang, G. M-Sequences and sliding window based audio watermarking robust against large-scale cropping attacks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1182–1195. [CrossRef]
39. Islam, S.; Naqvi, N.; Abbasi, A.T.; Hossain, I.; Ullah, R.; Khan, R.; Islam, M.S.; Ye, Z. Robust dual domain twofold encrypted image-in-audio watermarking based on SVD. *Circuits Syst. Signal Process.* **2021**, *40*, 4651–4685. [CrossRef]
40. Suresh, G.; Narla, V.L.; Gangwar, D.P.; Sahu, A.K. False-Positive-Free SVD based audio watermarking with integer wavelet transform. *Circuits Syst. Signal Process.* **2022**, *41*, 5108–5133. [CrossRef]
41. Maiti, C.; Dhara, B.C. A blind audio watermarking based on singular value decomposition and quantization. *Int. J. Speech Technol.* **2022**, *25*, 759–771. [CrossRef]
42. Narla, V.L.; Gulivindala, S.; Chanamallu, S.R.; Gangwar, D.P. Gangwar. BCH encoded robust and blind audio watermarking with tamper detection using hash. *Multimed. Tools Appl.* **2021**, *80*, 32925–32945. [CrossRef]

43. Hu, Y.; Lu, W.; Ma, M.; Sun, Q.; Wei, J. A semi fragile watermarking algorithm based on compressed sensing applied for audio tampering detection and recovery. *Multimed. Tools Appl.* **2022**, *81*, 17729–17746. [CrossRef]
44. SAlshathri, S.; Hemdan, E.E.-D. An efficient audio watermarking scheme with scrambled medical images for secure medical internet of things systems. *Multimed. Tools Appl.* **2023**, *82*, 20177–20195. [CrossRef] [PubMed]
45. Zhao, J.; Zong, T.; Xiang, Y.; Gao, L.; Hua, G.; Sood, K.; Zhang, Y. SSVS-SSVD based desynchronization attacks resilient watermarking method for stereo signals. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 448–461. [CrossRef]
46. Cruz, C.J.S.; Dolecek, G.J. Exploring Performance of a Spread Spectrum-Based Audio Watermarking System Using Convolutional Coding. In Proceedings of the 2021 IEEE URUCON, Montevideo, Uruguay, 24–26 November 2021; pp. 104–107. [CrossRef]
47. Ramsey, J.B. Wavelets. In *The New Palgrave Dictionary of Economics*; Palgrave Macmillan: London, UK, 2018; pp. 14510–14516. [CrossRef]
48. Farzaneh, M.; Toroghi, R.M.; Asgari, M. Audio Compression Using Graph-based Transform. In Proceedings of the 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 17–19 December 2018; pp. 410–415. [CrossRef]
49. Moffat, D.; Ronan, D.; Reiss, J. An Evaluation of Audio Feature Extraction Toolboxes. Available online: https://www.semanticscholar.org/paper/3ecb74b72cc1eae55b7db18622f576aa31d006c2 (accessed on 1 August 2025).
50. Zhan, Z.-H.; Shi, L.; Tan, K.C.; Zhang, J. A survey on evolutionary computation for complex continuous optimization. *Artif. Intell. Rev.* **2022**, *55*, 59–110. [CrossRef]
51. Method for Objective Measurements of Perceived Audio Quality, ITU Recommendation ITU-R BS.1387-1. 2001. Available online: https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1387-1-200111-S!!PDF-E.pdf (accessed on 1 October 2025).
52. Methods for the Subjective Assessment of Small Impairments in Audio Systems, ITU Recommendation ITU-R BS.1116-3. 2015. Available online: https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1116-3-201502-I!!PDF-E.pdf (accessed on 1 October 2025).
53. Mean Opinion Score (MOS) Terminology, ITU Recommendation ITU-R p.800.1. 2016. Available online: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.800.1-201607-I!!PDF-E&type=items (accessed on 1 October 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.