2023 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-6891-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICME55011.2023.00019

ADVERSARIAL AUDIO WATERMARKING: EMBEDDING WATERMARK INTO DEEP FEATURE

Shiqiang Wu^{1,3}, *Jie Liu*^{3},*Ying Huang*^{2,3,*}, *Hu Guan*^{3}, *Shuwu Zhang*^{2,3}

^{1} School of Artificial Intelligence, University of Chinese Academy of Sciences

^{2} School of Artificial Intelligence, Beijing University of Posts and Telecommunications

^{3} Institute of Automation, Chinese Academy of Sciences

* Correspondence: ying.huang@ia.ac.cn

ABSTRACT

Audio watermarking is a promising technology for copyright protection, yet traditional methods are limited that must be combined with auxiliary techniques against attacks. This article proposes a new audio watermarking method that embeds watermarks through a trained neural network. It adds small imperceptible perturbations to the original audio so that its deep features point to specific watermark features. Data augmentation and error correcting coding are employed to guarantee its practicable robustness. This method is robust against many attacks without auxiliary techniques and shows better performance than other deep learning-based methods.

***Index Terms*—** audio watermarking, deep learning, adversarial example

1. INTRODUCTION

Digital watermarking is a promising technology for digital copyright protection in the Internet era [1]. In watermarking systems, information, also called watermark, is embedded into the host material under the constraints of (1) imperceptibility – the distortion introduced by the watermark must be negligible; and (2) robustness–the extractor can detect the embedded information even if the material has been distorted to some extent.

In recent years, digital watermarking technology has shifted gradually from hand-designed traditional methods to new methods based on deep learning. Because computer vision is the most prevalent application of deep learning, these new methods also focus on the digital watermarking of images and videos. As audio is a vital form of daily communication, it is also essential to explore the deep learning-based digital audio watermarking method [2].

In traditional methods, the watermark is frequently embedded into the transform domain, such as DCT [3] or SVD [4], to ensure its imperceptibility. Anti-interference embedding algorithms and other robustness auxiliary techniques are widely employed to enhance robustness. These static heuristics methods are effective against the specific attacks and difficult to combine with each other.

In this study, we embed a watermark into a feature domain mapped by a deep neural network. It has been demonstrated that the deep features of deep neural networks are generalizations of the input materials [5], and small perturbations of the input can result in substantial changes in the deep features while the perturbations are imperceptible [6]. Thus, we add small perturbations to the original audio so that the features extracted by the network point to a specific feature, also called the watermark. Moreover, audio added perturbations is the watermarked audio.

Backpropagation will be used on samples of the input audio in order to embed the watermark. Meanwhile, we employ data augmentation and error correcting code, which are more generalize and combinable than other robust auxiliary techniques, to enhance robustness.

The contributions of this study as follow,

- A novel audio watermarking framework is provided, which can embed watermarks into audio using a trained neural network.
- Data augmentation is utilized during embedding to improve the robustness of watermarks.
- Due to error correcting coding on the watermark message, our method is robust to numerous attacks.

The remainder of this article is organized as follows, Section 2 introduces the related works; Scetion 3 describes the proposed method. Section 4 contains the experiment and discussion; The summary of this work is in Scetion 5.

2. RELATED WORKS

Traditional audio watermarking methods can be categorized according to their embedding domains, a few of which are time domain methods [7] and the majority of which are transform domain ones, which adhere to the transform, embedding and then inverse-transform pipeline, and these transforms include SVD (Singular Value Decomposition), DCT (Discrete Cosine Transform), and STFT (Short-Time Fourier Transform) [8], etc. Methods for transforming domains employ several embedding approaches with distinct properties. For instance, spread spectrum [3] is more anti-interference, but host signal interference occurs, whereas

1945-788X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICME55011.2023.00019

QIM [4](Quantization Index Modulation) is the inverse. Traditional methods must incorporate some auxiliary techniques to enhance their robustness [9, 10], yet these auxiliary techniques are designed for specific attacks. Thus, the types of attacks that these techniques can withstand are limited.

Deep learning based watermarking has emerged as a potential alternative to traditional methods in the image domain. The networks are frequently constructed as an encoder-decoder architecture, where the encoder embeds the watermark information into a host material, and the decoder accomplishes the extraction. For example, HiDDeN [11] cascaded the encoder, noise layer, and decoder and enhanced the perceptual quality using a parallel-connection adversarial discriminator. Liu et al. [12] proposed a two-stage learning method to address the constraint that the noise layer in HiDDeN must be differentiable. MBRS [13] adopted adversarial training for JPEG attacks to provide additional robustness, while Distortion Agnostic [14] performs a similar purpose for unknown transformations. IGA [15] and Yu [16] introduce attention mechanisms to improve the robustness and imperceptibility of watermarking. ReDMark [17] and Tavakoli et al. [18] integrate conventional DCT or DWT with deep learning to improve the methods' performance. There are also methods that use adversarial examples to watermark images. Fernandez et al. [19] used a pre-trained DINO model to avoid semantic collapse. Jia et al. [20] embed watermark on a black-box model. EAST [21] implements multi-bit embedding on the classification logit instead of the feature map. However, These image-domain methods have not been successfully applied to audio since the one-dimensional audio provides much less contextual information.

Kong et al. [22] developed a new method for audio watermarking based on adversarial examples, which embeds and recovers the information using a private DNN-based ASR (Automatic Speech Recognition) model. However, Kong et al. do not take some strategies to guarantee robustness, so that the embedding information is utterly unrecognizable after attacks.

3. THE PROPOSED METHOD

Our method embeds the watermark information after error correcting coding into the original audio through a trained neural network. Consequently, our method consists of three components: the trained deep feature extractor, the embedding and extraction approaches, and the error correcting code.

3.1. Trained deep feature extractor

We trained an audio classification network. The part before its linear layer is employed as the deep feature extractor f_θ : $\mathcal{A} \rightarrow \mathcal{F}$, where $\mathcal{A}$ is denoted as the audio domain and $\mathcal{F}$ as the feature space. The network structure is shown in Figure 1, where the feature extractor is in the pink box.

The input audio is first fed into the differentiable MFCC (Mel Frequency Cepstrum Coefficient) layer, followed by 6 residual blocks. The output of each residual block will be added as a feature map, and then go through 1×1 convolution layer and adaptive mean pooling to obtain a 320-dimensional deep feature. This deep feature will predict the audio into 8 different categories after passing through a linear layer and softmax layer. The activation function used in this network is $g(x) = tanh(W_1 x) \odot sigmoid(W_2 x)$, where W_1 and W_2 are learnable parameters and $\odot$ means Hadamard product, as it is shown that this activation function as more suitable for audio feature extraction [23].

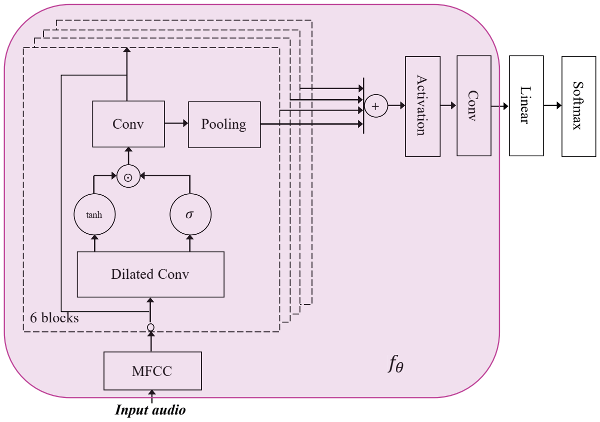


Fig. 1. Block diagram of the classification network. The part in the pink box is the deep feature extractor f_θ.

The feature extractor f_θ should meet two properties, (1) $f_\theta(A)$ will change as perturbing the input audio A imperceptibly to achieve the information embedding; and (2) the extracted features for different augmented forms of one audio should be as consistent as possible. Since the extractor f_θ we use is differentiable, information embedding can be achieved by adding perturbations to the audio via Backpropagation. A reasonable solution for satisfying feature invariance is to train the f_θ using augmentations of training data.

Data augmentation during training Consider a collection $\mathcal{T}$ of augmenting transforms that contain additive noise, random crop, lowpass filtering, and reverberation transforms. The data in the training dataset are transformed as stated above and fed into the network for training in to produce a transform-insensitive feature extractor.

3.2. Embedding and extraction approaches

3.2.1. Embedding

The watermark w is a polar vector of the same dimension as the deep feature, with elements that are either 1 or -1, which is obtained by substituting -1 for the 0 in a binary information string. The w can be considered as a vertex of a hypercube in the feature space $\mathcal{F}$.

The embedding process of our method is depicted in Figure 2. Initially, the original audio A_o is input into extractor f_θ and the loss $l^{'}$ is computed. During the iteration, a perturbation δ, determined by the loss, is added to the original

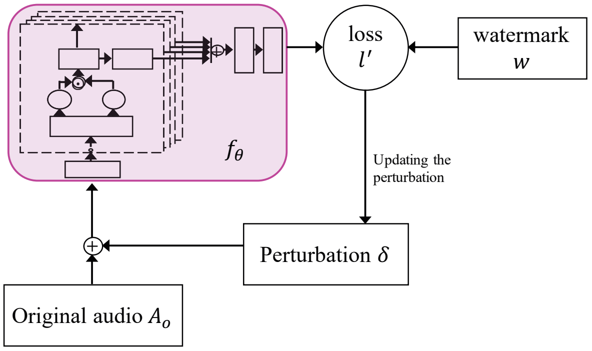


Fig. 2. The process of embedding the watermark into the original audio.

audio. Their sum is then re-fed into f_θ, and the loss is recomputed. Finally, when the iteration stops, the watermarked audio $A_w = A_o + \tilde{\delta}$, where $\tilde{\delta}$ is δ when the iteration stops.

We measure the distance between the deep features recovered by f_θ from the watermark vertex w using the loss function $L_w := \|w - f_\theta(A)\|_2^2$. On the other hand, it is anticipated that the difference between original and watermarked audio will be imperceptible, so we impose a limitation on the magnitude of the perturbation δ using the loss function $L_A := \frac{\|\delta\|_2^2}{\text{Size}(\delta)}$, where the denominator indicates averaging over all channels and samples of δ. Thus, the loss function of watermark embedding is,

$$\begin{aligned} l(A_o, \delta, w) &= L_w + \lambda L_A \\ &= \|w - f_\theta(A_o + \delta)\|_2^2 + \lambda \frac{\|\delta\|_2^2}{\text{Size}(\delta)} \end{aligned} \quad (1)$$

where λ is the weight to balance the two losses. We refer to L_w as the watermark loss to manage robustness and L_A as the distortion loss to ensure imperceptibility.

Data augmentation during embedding The collection of augmenting transforms $\mathcal{T}$ mentioned in Section 3.1 is likewise employed during the watermark embedding. These transformations are used as attacks that audio may encounter. The loss is computed by assuming that the watermarked audio may undergo attacks, and the distance between the deep features from these attacked audio and watermark w can be stated as,

$$L_{w,t} = \|w - f_\theta(\text{Tr}(A_o + \delta, t))\|_2^2 \quad (2)$$

where $\text{Tr}(A, t) \in \mathcal{A}$ denotes the application of a transformation (attack) $t \in \mathcal{T}$ to the audio A.

During watermark embedding, the transformed loss $L_{w,t}$ is first averaged according to various $t \in \mathcal{T}$ and then employed as the optimization objective. Consequently, when data augmentation is considered, the loss function of watermark embedding is as,

$$l^{'}(A, \delta, w) = \mathbb{E}_{t \in \mathcal{T}}\left[L_{w,t}\right] + \lambda L_A \quad (3)$$

where the term $\mathbb{E}_{t \in \mathcal{T}}\left[L_{w,t}\right]$ pushes the deep feature to the designated point (watermark), and ensures that the watermark information can be extracted after various transformations; furthermore, the term L_A limits the magnitude of the perturbation and keeps the watermark imperceptible.

The Eq. (3) can be solved by the typical method of adversarial attacks [6, 24],

$$\tilde{\delta} = \underset{\delta}{\text{argmin}}\, l^{'}(A, \delta, w) \quad (4)$$

Since all the augmenting transform $t \in \mathcal{T}$ and the feature extractor f_θ are differentiable, Gradient descent optimizers can be used to optimize the loss $l^{'}(A, \delta, w)$. Moreover, it is assumed that all transformations $t \in \mathcal{T}$ are equally probable.

3.2.2. *Extraction*

Simply feeding the received watermarked audio $\tilde{A}_w$ into f_θ yields a deep feature extraction. The extracted watermark information is the nearest hypercube vertex to the deep feature corresponding to the watermarked audio, as,

$$\hat{w} = \text{binary}(f_\theta(\tilde{A}_w)) \quad (5)$$

where

$$\text{binary}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

3.3. Error correcting code

The embedding approach in Eq. (4) minimizes the loss between extracted and embedded watermark, but does not guarantee that they are identical. We employ RS (Reed-Solomon) error correcting code to ensure that the extracted watermark information is as precise as feasible.

RS code is one of the most powerful known linear cyclic packet codes to correct random and multiple burst errors. We employ codes whose codewords are drawn from the Galois field $\text{GF}(2^8)$, where every 8 bits constitute a codeword, ideally suited for byte-wise processing on computers.

During the encoding phase, the created redundant codes are attached to the message codes to generate RS codes. RS codes with a certain level of inaccuracy can still reconstruct the message codes accurately.

We encode a 64-bit (8-byte) message string as a 320-bit (40-byte) watermark string and replace its zeros with -1 to obtain a 320-dimensional watermark w. This encoding can correct errors of up to 16 bytes. When there are too many errors to decode, only the first 64 bits of the 320-bit string are extracted as the message.

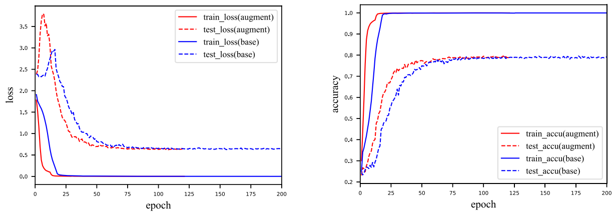
4. EXPERIMENTS AND DISCUSSIONS

4.1. Experimental settings and implementation details

Data: The classification network was trained and evaluated in the FMA-small dataset [25]. The dataset contains 8000

Table 1. The transformations or attacks used in this study

Abv.	Attack	Description
CLP	Closed loop	Audio is not attacked.
WGN	Additive white Gaussian Noise	Gaussian noise with SNR of 20dB ia added.
LPF	Lowpass filter	Audio is filtered by a Butterworth filter, with 8kHz cutoff.
CRP	Random Crop	Audio is cropped randomly, retaining 80% of duration.
RVB	Reverberation	Simulating audio propagation in a closed room.



(a) loss

(b) prediction accuracy

Fig. 3. The classification network training.

audio tracks, of which 6400 belong to the trainingset and 800 tracks each to the test and validation sets. These tracks are stereo audio in MP3 format with a samplerate of 44100Hz and a duration of 30s. The dataset consists of 8 categories, each with 1000 audio tracks. 100 audio from the validation set were used for the watermark embedding experiments.

Experimental settings: During watermark embedding, the weight in Eq. (3) is set to $\lambda = 1$. Adam [26] optimizer with a learning rate of 0.001 are used to solve Eq. (4) over 300 iterations. The settings of the training network will be mentioned later.

Data augmentations or attacks: The transformations or attacks used at training network, watermark embedding and evaluation are presented in Tab. 1. The reverberation was simulated by Pyroomacoustics [27] with a $9 \times 7.5 \times 3.5\text{m}^3$ rectangular room. To avoid the difficulty of gradient computation, we use masking with 0s instead of the random crop when training network and watermark embedding. When evaluating the proposed method, we use the actual crop.

Metrics: The imperceptibility of the watermark is evaluated by SWR (Signal Watermark Ratio), which means the energy ratio of perturbation to audio,

$$\text{SWR} = 10\lg\frac{\|\delta\|_2^2}{\|A_o\|_2^2}$$

The BDR (Bit Detection Rate) is used to evaluate the robustness of the proposed method, where CBDR (Coarse BDR) measures the robustness of 320-dimensional features and the FBDR (Fine BDR) measures the robustness of the watermark after RS decoding.

4.2. Training of the classification network

The classification network predicts the audio into 8 categories according to musical style, and the cross-entropy loss was employed for training. The network was trained twice, one with augmented data (augment) and the other without (base).

The Adam optimizer with learning rate 0.001 is also employed for network training, and training is terminated when the training loss and test loss cease dropping. Training with augmented data was iterated with 120 epochs, while the base training was iterated with 200 epochs.

The loss and prediction accuracy of training and test are shown in Figure 3. The red plots in Figure 3 indicate the 'augment' training, while the blue plots indicate the 'base' training. The solid lines indicate the training experiment, and the dashed lines mean the test.

After training, the network perfectly predicted the categories on the training set, but on the test set, only 65% of predictions were accurate. Although 'augment' converges faster, 'base' and 'augment' networks almost achieve the same end performance.

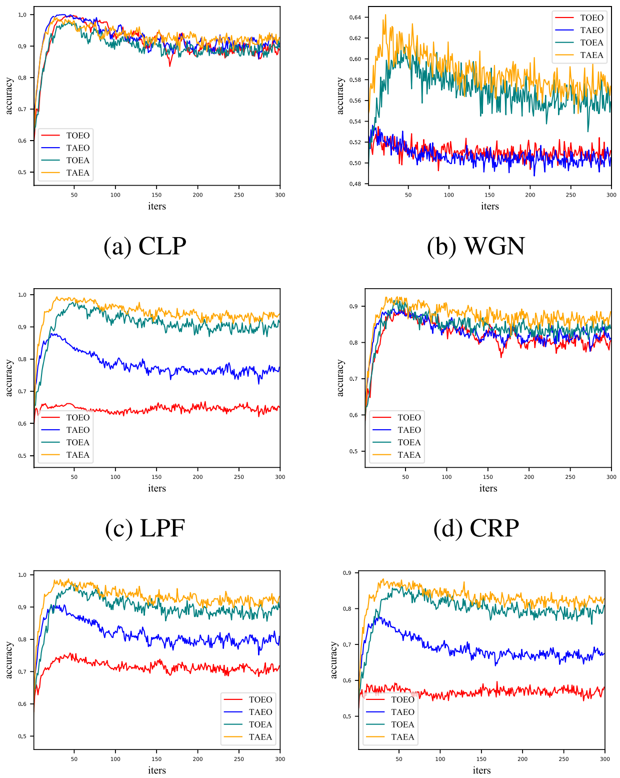
4.3. Ablation study of data augmentation

In this subsection, we conduct an ablation study on the application of data augmentation. We performed four embedding experiments, which are (1) using the 'base' feature extractor and embedding without augmented data (TOEO), (2) using the 'augment' feature extractor and embedding without augmented data (TAEO), (3) using the 'base' feature extractor and embedding with augmented data (TOEA), and (4) using the 'augment' feature extractor and embedding with augmented data (TAEA).

The CBDRs were recorded during these experiments. The results are shown in Figure 4. In the presence of attacks, there are significant improvements through data augmentations. Overall, embedding watermarking with data augmentation (TOEA, green) improves performance significantly more than the training network with data augmentation (TAEO, blue). The method achieves the best robustness when the training network and the embedded watermark both with data augmentation (TAEA, orange).

The CBDRs are different against different attacks. In particular, the robustness of the four experiments is comparable against CLP (no attack) and finally detects close to 90% of the watermark bits. The experimental results of LPF and RVB have some similarities due to the fact that they both use convolution for implementation. The difference between the various experiments in CRP is negligible, as both can only extract roughly 80% of the watermark. Embedding with data augmentations makes the proposed method more robust against WGN, but the accuracy of TAEA is still below 60%.

In TAEA, the robustness of our method is better against LPF than against CLP, and we speculate that the lowpass filtering and MFCC calculation make the watermark more robust. For this, WGN and LPF were combined and participated in the experiments, and the experimental results showed that WGN+LPF also demonstrated good robustness.

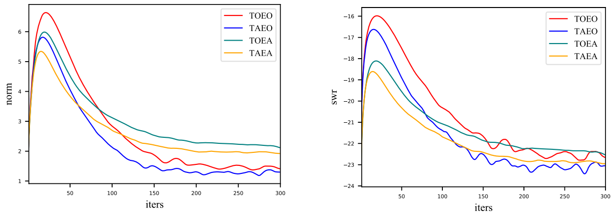


(a) CLP (b) WGN

(c) LPF (d) CRP

(e) RVB (f) WGN+LPF

Fig. 4. The CBDRs against different attacks.



(a) perturbed norm $||\delta||_2^2$ (b) SWR

Fig. 5. The imperceptibility of the watermark

We also recorded the perturbed norm $||\delta||_2^2$ and the SWRs, as shown in Figure 5. They can reflect the imperceptibility of the watermark. The norm $||\delta||_2^2$ gradually oscillates as the iteration proceeds due to the conflict between watermark loss L_w and distortion loss L_A. TOEA and TAEA correspond to a slightly larger norm than TAEO and TOEO. The data augmentation during embedding limits the range of perturbation so that only a larger magnitude of perturbation can be chosen to satisfy the robustness of the watermark. However, it is not easy to see this cost in SWR.

In a nutshell, data augmentation significantly improves the robustness of watermarking. In the subsequent experiments, we only use the results of TAEA.

Table 2. The CBDRs and FBDRs

Attack	CBDR(%)	FBDR(%)	Diff
CLP	92.75	99.29	6.54
WGN	57.37	57.80	0.53
LPF	94.00	99.42	5.42
CRP	88.50	96.55	8.05
RVB	93.30	99.33	6.03
WGN+LFP	82.73	92.23	9.50

4.4. Effectiveness of RS code

We further evaluated the FBDR of this method and compared it with CBDR. Due to the time consumption of RS decoding, we only experimented with the results after the embedding iterations were stopped. The experimental results are shown in Table 2.

RS coding has improved robustness in most attacks. The FBDR reaches more than 95% except for both WGN and WGN+LPF attacks.

Although perfect extraction is not achieved, we believe that 95% of the FBDR is robust enough in the application.

A further benefit of the proposed method is that it resists reverberation and cropping without auxiliary techniques, which is impossible with traditional methods.

4.5. Comparison with existing works

We compare the imperceptibility and robustness of the proposed method with other methods. Since deep learning based audio watermarking methods are scarce in the literature, we only found the work by Kong et al. [22]. We transferred HiDDeN [11], Landmark work for deep image watermarking, to the audio domain as HiDDen-A and trained it with 20,000 minutes of audio (part of the FMA dataset) and then compared HiDDeN-A with the proposed method.

The comparisons are shown in Table 3. Our method has better robustness than that of comparisons. The imperceptibility is slightly worse than the comparison methods, but the improvement in robustness is more pronounced.

The performance of HiddeN-A is worse than that of HiDDeN in the image domain, indicating that additional efforts are needed for directly transfering the image watermarking to audio. Moreover, Kong's method focuses more on information hiding and ignores the necessity of robustness. Our method demonstrates a more practical performance in comparison, which is rare in deep learning-based audio watermarking.

5. SUMMARY

In this article, we propose a new audio watermarking method that embeds watermarks into deep features using trained neural networks. We demonstrate that data augmentation significantly increases the watermarking's robustness. The performance of the proposed method is also enhanced via RS coding. The proposed method is more robust than existing DL-based methods.

Table 3. The performance of proposed and other methods. (Imp. means imperceptibility, Rob. means robustness.)

Performance		Kong[22]	HiDDeN-A	Our
Imp.(dB)	SWR	30.17	28.24	22.95
Rob. (FBDRs:%)	CLP	100	88.73	99.29
	WGN	0	47.93	57.80
	LPF	1	68.49	99.42
	CRP	0	79.78	96.55
	RVB	1	52.48	99.33

However, our work still deserves of improvement. The feature extractor trained with the classification challenge suffers from semantic collapse, and extracting more extensive audio features will be the focus of the following research. In addition, we will continue to improve the robustness of this work against pointwise random noise.

Acknowledgement

This work was supported by the National Key R&D Program of China (2020YFB1406900) and the Key R&D Program of Shanxi (202102010101004). It was also the research achievement of the Beijing Digital Content Engineering Research Center and the Key Laboratory of Digital Rights Services.

6. REFERENCES

[1] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker, *Digital watermarking and steganography*, Morgan kaufmann, 2007.

[2] Guang Hua, Jiwu Huang, Yun Q Shi, Jonathan Goh, and Vrizlynn LL Thing, "Twenty years of digital audio watermarking—a comprehensive review," *Signal processing*, vol. 128, 2016.

[3] Yong Xiang, Iynkaran Natgunanathan, Dezhong Peng, Guang Hua, and Bo Liu, "Spread spectrum audio watermarking using multiple orthogonal pn sequences and variable embedding strengths and polarities," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 3, 2018.

[4] Min-Jae Hwang, JeeSok Lee, MiSuk Lee, and Hong-Goo Kang, "Svd-based adaptive qim watermarking on stereo audio signals," *IEEE Transactions on Multimedia*, vol. 20, no. 1, 2017.

[5] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, "Understanding neural networks through deep visualization," *arXiv:1506.06579*, 2015.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, 2014.

[7] Paraskevi Bassia, Ioannis Pitas, and Nikos Nikolaidis, "Robust audio watermarking in the time domain," *IEEE Transactions on multimedia*, vol. 3, no. 2, 2001.

[8] Shengbei Wang, Weitao Yuan, Zhen Zhang, Jianming Wang, and Masashi Unoki, "Synchronous multi-bit audio watermarking based on phase shifting," in *ICASSP*. IEEE, 2021.

[9] Wenhuan Lu, Ling Li, Yuqing He, Jianguo Wei, and Neal N Xiong, "Rfps: a robust feature points detection of audio watermarking for against desynchronization attacks in cyber security," *IEEE Access*, vol. 8, 2020.

[10] Andrew Nadeau and Gaurav Sharma, "An audio watermark designed for efficient and robust resynchronization after analog playback," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, 2017.

[11] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *ECCV*, 2018.

[12] Yang Liu, Mengxi Guo, Jian Zhang, Yuesheng Zhu, and Xiaodong Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *ACM MM*, 2019.

[13] Zhaoyang Jia, Han Fang, and Weiming Zhang, "Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression," in *ACM MM*, 2021.

[14] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar, "Distortion agnostic deep watermarking," in *CVPR*, 2020.

[15] Honglei Zhang, Hu Wang, Yidong Li, Yuanzhouhan Cao, and Chunhua Shen, "Robust watermarking using inverse gradient attention," *arXiv:2011.10850*, 2020.

[16] Chong Yu, "Attention based data hiding with generative adversarial networks," in *AAAI*, 2020.

[17] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami, "Redmark: Framework for residual diffusion watermarking based on deep networks," *Expert Systems with Applications*, vol. 146, 2020.

[18] Alireza Tavakoli, Zahra Honjani, and Hedieh Sajedi, "Convolutional neural network-based image watermarking using discrete wavelet transform," *arXiv:2210.06179*, 2022.

[19] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze, "Watermarking images in self-supervised latent spaces," in *ICASSP*, 2022, pp. 3054–3058.

[20] X. Jia, X. Wei, X. Cao, and X. Han, "Adv-watermark: A novel watermark perturbation for adversarial examples," 2020.

[21] S. Ghamizi, M. Cordy, M. Papadakis, and Y. L. Traon, "Evasion attack steganography: Turning vulnerability of machine learning to adversarial attacks into a real-world application," in *ICCV*, 2021.

[22] Yehao Kong and Jiliang Zhang, "Adversarial audio: A new information hiding method.," in *INTERSPEECH*, 2020.

[23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199*, 2013.

[25] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "FMA: A dataset for music analysis," in *ISMIR*, 2017.

[26] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[27] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *ICASSP*, 2018.