Perception-Aware Attack Against Music Copyright Detection: Impacts and Defenses

Rui Duan, Zhe Qu, Shangqing Zhao, Leah Ding, Yao Liu, and Zhuo Lu

***Abstract*—Recently, adversarial machine learning attacks have posed serious security threats against practical audio signal classification systems, including speech recognition, speaker recognition, music copyright detection. Most existing studies have mainly focused on ensuring the effectiveness of attacking an audio signal classifier via creating a noise-like perturbation on the original signal, which remains a gap in preserving the human perception of adversarial audios. This paper presents a novel perspective to create adversarial audios by integrating the human perception model into the attack formulation to generate well-perceived adversarial examples. Different from conventional approaches which primarily focused on using L_p norm to preserve the audio quality, we adopt a human study to understand how human participants react to different types of music perturbations, build a Siamese Neural Network (SNN) based model to characterize the human perception. The new findings of the human perception study guide us to formulate a new computationally efficient, multiple-feature-based perception-aware (CEMF-PA) attack, which manipulates different audio signal features to find an optimal perturbed music signal against music copyright detection. This novel attack vector opens a new door to generating highly effective, well-perceived adversarial audio signals via manipulating the auditory features. Experimental results show that the proposed attack is effective against YouTube's copyright detection. Finally, we propose the defense strategy design to make the copyright detection more robust to adversarial music signals generated by the CEMF-PA attack.**

***Index Terms*—Security, computer audio systems, applications, machine learning.**

I. Introduction

ADVERSARIAL machine learning attacks, originating from the image domain [2], [3], [4], [5], have recently become a serious security issue in audio signal processing system designs leveraging machine learning, including speech and speaker recognition [6], [7], [8], [9], [10], [11], [12] and music copyright detection [13].

Adversarial machine learning attacks attempt to create a small perturbation on the original audio signal such that a machine learning classifier can yield an incorrect output. For example, a small change in a speech command could make Amazon Echo [14] and Google Assistant [15] recognize a different, yet malicious command [8], [16]. And manipulating copyrighted music might bypass the copyright detection in YouTube [13]. One key component in adversarial audio signals is the perturbation, which is designed to cause misclassification and at the same time be small enough to be hardly noticed. To quantify the perturbation, existing studies [11], [17] usually use a mathematical distance (e.g., the Euclidean distance [13], or more generally, the L_p norm [4]) between the original and perturbed audio signals. Therefore, the perturbed signal with the minimized distance to the original one could be considered as a good candidate under the constraint that it can successfully spoof the classifier.

However, the L_p norm based methods only measure the magnitude distance between two signals; but the human perception is much more complex than computing the magnitude distance. There exists a gap between the mathematical distance and the eventual human perception. Although the two may be related in some way (e.g., zero distance meaning no signal perturbation), there is still no direct relation to indicate an increase or decrease of the distance in mathematics would be human-perceived as the same. For example, adding a perturbation that is the same as the original music signal is equivalent to increasing the volume of the music, which does not quite change the human perception of music quality. Indeed, a few studies [4], [9] have pointed out similar issues and indicated that new methods are needed to measure the perceptual similarity between the original and perturbed signals; but there is limited work on systematically designing adversarial machine learning from the human perception perspective. In this paper, we create a new mechanism to craft adversarial audio signals. We focus on generating adversarial music signals to bypass a music copyright detector and hardly raise human attention. To study how a change of a music signal affects human perception, we first conduct a human study where volunteers quantify their perceived deviations between the original and perturbed signals as ratings on a Likert scale. We use regression analysis to build an approximate mathematical relation between the change of music and the human-perceived deviation rating obtained from the human study. Considering the human

Received 28 March 2023; revised 2 October 2024; accepted 19 December 2024. Date of publication 26 December 2024; date of current version 15 May 2025. *(Corresponding author: Zhe Qu.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of South Florida Institutional Review Board under Application No. STUDY003214, and performed in line with Exempt Determination.

Rui Duan is with the School of Science and Engineering, University of Missouri-Kansas City, Kansas City, MO 64110 USA (e-mail: ruiduan@umkc.edu).

Zhe Qu is with the School of Computer Science and Engineering, Central South University, Changsha 410017, China (e-mail: zhe_qu@csu.edu.cn).

Shangqing Zhao is with the School of Computer Science, University of Oklahoma, Tulsa, OK 74135 USA (e-mail: shangqing@ou.edu).

Leah Ding is with the Department of Computer Science, American University, Washington, DC 20016 USA (e-mail: ding@american.edu).

Yao Liu is with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: yliu@cse.usf.edu).

Zhuo Lu is with the Department of Electrical Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: zhuolu@usf.edu).

Digital Object Identifier 10.1109/TDSC.2024.3522849

perception process of a music perturbation as a black box, we obtain inputs and outputs of the black box from our human study: the input is the audio feature deviation from the original music signal to the perturbed one; the output is the averaged deviation rating by human volunteers. As a result, we reverse-engineer this black-box perception process via regression analysis and build a computationally efficient Siamese Neural Network (SNN) based model to predict the human rating of a music signal deviation.

We then formulate the adversarial music attack as an optimization problem of jointly manipulating multiple audio features of a music signal to generate the perturbed music signal to bypass copyright detection while suppressing human attention based on the SNN human modeling. We call this strategy computationally-efficient, multiple-feature based perception-aware (CEMF-PA) attack. Experimental results show that the CEMF-PA attack can produce effective adversarial music to bypass YouTube's detection while achieving a significantly higher perceptual quality compared to existing attack strategies. Finally, we propose an ensemble-based adversarial audio fingerprinting method to defend against the CEMF-PA attack. Our main contributions are summarized as follows.

- We propose a new attack vector that integrates the human perception model into the generation of adversarial examples. Specifically, we build an SNN model based on a human study to characterize the relationship between the audio feature deviation and the human-perceived deviation for music signals. By leveraging the inherent efficiency of SNNs, our model addresses the computational limitations of existing methods, providing a scalable and accurate solution. Moreover, we demonstrate that the SNN architecture, traditionally applied in natural language processing and computer vision, can be effectively adapted to the audio domain, thus offering a novel approach for processing music signals with high computational efficiency and human-aligned accuracy.
- We create a new benchmark for generating adversarial music signals via manipulating multiple music features, which can jointly preserve human perception and enhance the attack effectiveness. We formulate the CEMF-PA attack that makes small changes to multiple musical features to generate adversarial signals against copyright detection. Experimental results show that the attack is more effective than existing attack methods.
- We propose a new defense perspective by combating the adversarial examples within a specific human perception range. In particular, we propose an ensemble-learning-based fingerprinting method that integrates human perception into the defense design. Experimental results show that it can accurately detect adversarial music generated by the CEMF-PA, ICML20, and psychoacoustic attack, and have better performance than the existing defense methods.

The rest of the paper is organized as follows: we present related work in Section II. We introduce the background and the motivation of our study in Section III. Section IV elaborates our human study with regression analysis. We formulate the perception-aware attack framework, create a realistic attack, and conduct experiments in Sections V, VI, and VII, respectively. Section VIII introduces the specific defense strategy, and Section IX presents the discussions and limitations. Finally, we conclude this paper in Section X.

II. Related Work

We first present and discuss research related to our study.

Adversarial audio attacks: Most adversarial attacks [11], [16], [17], [18], [19] control the energy of the perturbation within a bounded L_p ball such that a created adversarial audio example resembles the original signal in its waveform format. There are also a few recent studies [6], [7], [8], [20], [21] focusing on creating inaudible or stealthy signals as attacks. For example, dolphin attacks [20] leveraged the nonlinearity of the microphone circuits to achieve inaudible attacks on speech recognition. Hidden voice attacks [6], [21] crafted the obfuscated commands to circumvent the human recognition. In [7], [8], malicious commands can be embedded into songs. These studies generally use various strategies to effectively hide the presence of the attack. However, these works lack comprehensive human studies to understand how the perturbed signals impact human perception before their attack designs.

Human evaluation of audio quality: Human perception studies [6], [7], [8], [11], [16] have been adopted to evaluate the stealthiness of adversarial audio examples as the SNR metric may not be appropriate to well reflect the human perception [8], [16]. Exiting work [6], [7], [8], [11], [16] designed human perception studies from different perspectives and evaluated the attack performance based on the results of human study. For instance, [22] conducted a comprehensive human study to evaluate the synthetic speech quality to reveal the impact of deep-learning based speech synthesis to human. These studies focused on analyzing the results of the human evaluation, rather than integrating human factors into the designs. There are few studies [23], [24] focusing on defining human-involved metrics for singing scoring systems. The systems were designed to generate an absolute score to indicate the singing performance given the recording of a human's singing via linear weighting [23] or non-linear neural network [24] on audio features, but these works lack the integration of human-involved metrics into the generation of adversarial signals.

Defenses against adversarial attacks: Adversarial training [2], [25], [26], [27], [28], [29], [30] and certified defense [31], [32], [33], [34] are popular among the methods to provide more robustness against adversarial attacks. The essential idea of adversarial training [2], [5] is to train the machine learning model with both clean and adversarial examples, and there are also some studies [35], [36] forcing the machine learning model to only train with the strong adversarial examples. Certified defense is to find an upper bound of the adversarial loss which guarantees the robustness to any attack in the same threat model. Existing work [31] can provide a provable defense to the neural networks via convex layer wise adversarial training. Both types of defenses adopt the L_p-norm-based measure of distance. On the other hand, there are also some works, investigate defending

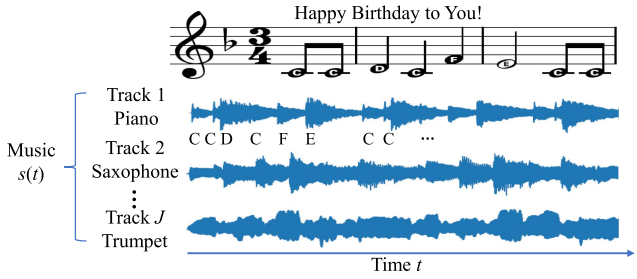


Fig. 1. Music with multiple track signals by different instruments, and each track contains a series of notes.

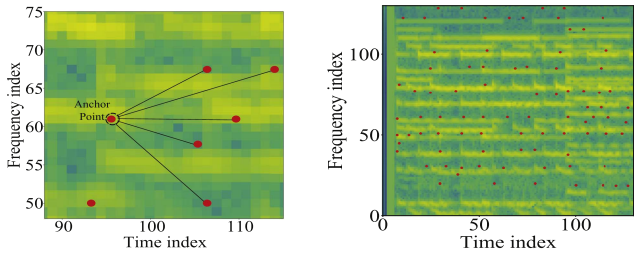


Fig. 2. Fingerprinting: finding all spectrogram peaks.

the adversarial examples from the audio signal processing perspective. For example, existing works [16], [37] evaluate the audio down-sampling method against the adversarial audios, and audio quantization [38] and low-pass filtering [17] have also been demonstrated as effective methods against adversarial examples. However, these defense methods overlook human perception. Defenders should aim to make the model robust against perturbed signals within a reasonable range of human perception, which would enhance the meaningfulness of the defense.

III. Background and Design Motivation

In this section, we briefly introduce the background and describe our motivation and design intuition.

A. Representation of Music Signal

As an example shown in Fig. 1, a digital music signal $s(t)$ at sample time $t \in \{0, 1, 2, \ldots, T\}$ (where T is the number of signal samples) can be represented as the sum of audio track signals [39], i.e., $s(t) = \sum_{j=1}^{J} s_j(t)$, where J is the number of tracks, and the track signal $s_j(t)$ is a time-series of harmonic notes [40], [41], [42], [43]. A note, similar to a phoneme of speech [7], [20], is the smallest signal unit of a piece of music consisting of a fundamental frequency and a set of harmonics [44], [45], [46].

B. Audio Fingerprinting

A key technique for audio signal classification is audio fingerprinting [47]. The technique and its variants have been widely adopted in audio signal watermarking [48], [49], integrity verification [50], music information retrieval [51], [52], [53], broadcast monitoring [54], [55], [56] and copyright detection [13]. The audio fingerprint detection system usually retrieves the information from stored fingerprints (e.g., in hash format). And the essential idea in audio fingerprinting is to consider certain high-energy areas of an audio signal in the spectrogram as its fingerprints. As an example shown in Fig. 2(a) [51]: an energy peak (anchor point) is paired with other peaks within a certain target area in a signal's spectrogram, then the fingerprints are computed based on the frequency information of the peaks and the time intervals between them. Fig. 2(b) shows there are many peaks in a signal's spectrogram that lead to a large number of fingerprints for audio signal classification and identification.

C. Adversarial Audio Attacks

Given a classifier with prediction function $f(\cdot)$ which takes the input audio signal $s(t)$ and outputs the correct label $f(s(t)) = y$, existing adversarial audio attacks [9], [18], [57] aim to add a small signal perturbation $\delta(t)$ to the original audio signal $s(t)$, and then supply the perturbed signal $\hat{s}(t) = s(t) + \delta(t)$ to the classifier that accordingly generates an incorrect label. The method of creating $\delta(t)$, which mainly inherits from the fundamental framework in the image domain [4], [5], can be formulated as

$$\begin{aligned} &\text{minimize} && \|\delta(t)\|_p \\ &\text{subject to} && f(\hat{s}(t)) \neq y, \end{aligned} \tag{1}$$

where $\|\delta(t)\|_p$ denotes the L_p norm of the perturbation $\delta(t)$ [2], [4]. The objective of (1) is to minimize the change of the perturbed signal $\hat{s}(t)$ from the original $s(t)$. Since it is computationally difficult to solve (1), many variants of formulating the adversarial audio attacks have been proposed for distinct attack scenarios, such as speech recognition [9], [18], [57], speaker recognition [11], [16], and music copyright detection [13]. To still make $\hat{s}(t)$ look like $s(t)$, these formulations limit the L_p norm of the perturbation $\delta(t)$ within a given threshold ϵ, i.e., $\|\delta(t)\|_p \leq \epsilon$. The L_∞, L_2, and L_0 norms are commonly adopted in the literature to create adversarial attacks targeting various audio signal classifiers [11], [16], [17], [18], [19].

D. Motivation and Design Intuition

Although existing adversarial audio attacks mathematically limit the magnitude of the perturbation $\delta(t)$ via $\|\delta(t)\|_p \leq \epsilon$, it is still not clear whether such a constraint is the most effective to make the perturbation unnoticeable by human beings. For example, a few studies [4], [9] have noted the concern on whether the L_p norm metric is appropriate to measure the signal similarity from the human perception perspective. In other words, there is no evidence to show that the deviation in human cognition can be represented by $\|\delta(t)\|_p$. As a result, we are motivated to investigate the problem. Our goals are twofold: i) relating the change of a music signal to the deviation of human perception

and ii) finding a new way to create the perturbation that is unnoticeable by human beings as much as possible.

E. Threat Model

In this paper, we consider an attacker that aims to find a perturbation $\delta(t)$ to a music signal $s(t)$ such that $\hat{s}(t) = s(t) + \delta(t)$ leads to an incorrect output of an audio signal classifier, which is similar to the goal of existing audio attacks [5], [13], [16], [17], [18], [57]. At the same time, the attacker is designed to be aware of how $\hat{s}(t)$ affects the human perception and minimizes its perceived deviation from $s(t)$. We assume that the attacker has no knowledge of the algorithm design or parameter choices in the classifier, but has access to the classification result of any input signal. We also assume that the attacker has no access to the classifier's training database. A representative commercial scenario is that an attacker wants to bypass YouTube's copyright detector [13] and use copyrighted music content in an unauthorized way to attract more online views for advertisement revenue gain.

IV. Reverse-Engineering Human Perception of Music Signals

In this section, we present how to quantify the human perceived deviation of music signals. We first analyze the key features for the signal quality, then conduct the human study, and present the study results and regression analysis.

A. Audio Features for Human Perception

Based on existing studies in audio engineering [23], [24], [58], there are four widely-used features: pitch, rhythm, timbre, and loudness. Pitch is the subjective perception of highness or lowness of a sound, and is referred to as the fundamental frequency ω_0 of a note [59], [60]. Rhythm is described as the tempo of the musical sound [23], which depends on the length of each note and the time intervals between adjacent notes. Timbre is the mixture of the harmonics, which brings the "color" to music [60], and it is similar to the characteristics of the speech [61]. Loudness measures the intensity of an audio signal and can be seen as the energy level or the volume of the signal [23].

In the following, we briefly introduce the commonly-used methods to compute the feature deviations between two signals $s(t)$ and $\hat{s}(t)$ in the literature. For each feature, the procedure is the same and shown in Fig. 3: $s(t)$ and $\hat{s}(t)$ each will be separated into frames with a small time interval (e.g., 16 ms [24]). The signal samples in each frame are used to generate a feature value (e.g., pitch value). The feature values from all frames constitute a time-series data vector. Then, an algorithm called Dynamic Timing Warping (DTW) [62] is used to quantify the similarity between the time-series vector for $s(t)$ and the one for $\hat{s}(t)$, and generate a vector of frame-wise deviation values for the feature. The advantage of DTW over the Euclidean distance is that DTW can reduce the time distortion [63] via finding an optimal path between two time-series vectors. For instance, the red line in Fig. 3 indicates the DTW path between $s(t)$ and $\hat{s}(t)$.

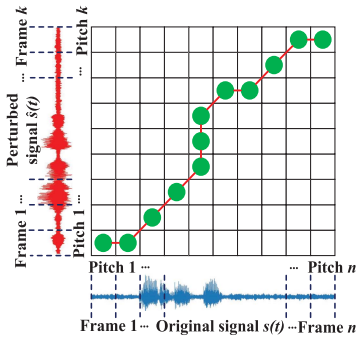


Fig. 3. Computing deviation values via DTW.

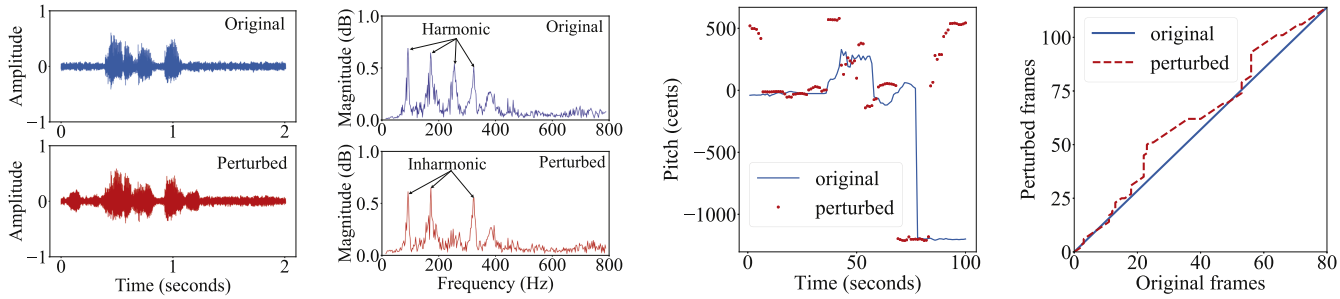
- Pitch: The pitch value in each frame is the basic frequency ω_0 obtained via pitch estimation, which is a maximum likelihood estimation problem [64] via finding ω_0 from harmonics $\sum_{m=1}^{M} m\omega_0$. The estimated pitch values from all frames form a time series for each signal and then DTW is used to generate the vector of frame-wise pitch deviation values between the two signals.
- Rhythm: Rhythm computation is based on pitch estimation. A deviation value for rhythm between two frames is computed as the linear regression error in DTW during computing the deviation value for pitch [58]. All these values generated during DTW form the vector of frame-wise deviation values for rhythm.
- Timbre: The timbre value for each frame is computed as a Mel-Frequency Cepstrum Coefficient (MFCC). The vector of frame-wise deviation values for timbre is the result of the DTW between the MFCC vectors for $s(t)$ and $\hat{s}(t)$.
- Loudness: Loudness is closely related to the L_p norm used in existing adversarial attack formulations (1). The loudness for each frame is usually calculated as the short-term log-energy [23], which is the logarithm of the total energy of the frame. After two short-term log-energy vectors for $s(t)$ and $\hat{s}(t)$ are obtained, the DTW between them generates the vector of frame-wise deviation values for loudness.

The last step for each feature is to aggregate the computed vector of frame-wise deviation values into a single value to represent the overall feature deviation. According to existing studies [24], [65], the non-linear average calculation is commonly adopted for pitch and rhythm aggregations, and linear averaging is used for timbre and loudness. After the aggregations, the resultant four feature deviation values form a final feature deviation vector to describe the audio characteristic deviation from $s(t)$ to $\hat{s}(t)$.

B. Impacts of Audio Feature Deviations

To understand how pitch, rhythm, timbre, and loudness change in a perturbed signal, we show the feature deviations caused by an adversarial example in [13] in Fig. 4.

As [13] adopted an L_p norm based formulation to create adversarial audio and limited the L_p norm of the perturbation, Fig. 4(a) shows that there is a minor waveform change in the time-domain between the original and perturbed music signal.



(a) Waveforms (b) Harmonic note spectrum (c) Pitch deviation (d) Rhythm deviation

Fig. 4. Impacts of a noise-like perturbation on the music features: a 2-second attack example "Boom Boom Pow" from existing work [13]. Specifically, (a) and (b) shows the waveforms and spectrums, respectively; (c) and (d) show the pitch contours and the rhythm DTW paths between the perturbed and original signals, respectively.

This indicates that the perturbation only incurs a small energy or loudness change to the original signal.

Next, we look at the waveform change in the frequency-domain and compare the power spectrum in Fig. 4(b). The observed change is more evident than the time domain in Fig. 4(b): the third harmonic in the original harmonics is suppressed, which leads to inharmonicity in signal and negatively impact timbre and accordingly the audio quality.

If we look at the pitch contours (i.e., the curves drawn by connecting all pitch values over time) for the original and perturbed signals in Fig. 4(c), we find the evident difference of the pitch between the two signals. Fig. 4(d) shows the optimal DTW path of the perturbed signal to the original one. Intuitively, a music signal with the minimal rhythm deviation should have a nearly straight line DTW path. Fig. 4(d) shows that the DTW path of the perturbed signal is tortuous compared with the original one.

Note that creating adversarial music inevitably causes some distortions of the original signal. Fig. 4 demonstrates that there may exist some way to better coordinate such distortions among all audio features to mimic the original signal's quality as much as possible since they are eventually perceived by humans. If we look at the basic adversarial audio attack formulation used in recent research [11], [13], [17], the L_p norm of the additive noise is only relevant to the loudness feature without a clear relation to the other three features. It is evident that L_p norm is much easier to compute than pitch, rhythm, and timbre via gradient descend. At the current stage, we do not focus on the computational aspect but on the human perception aspect and continue to understand how these features affect human perception.

C. Human Study Procedures and Setups

To understand how different features affect human perception. We conduct a human study with the procedure shown in Fig. 5: we first generate a dataset that consists of pairs of original and perturbed music signals. For each pair, we can compute (according to the procedure in Section IV-A) the deviation values for the four features, which form a feature deviation vector. Then, we invite every human participant to assign a deviation rating to each pair based on his/her perceived difference. Next, considering the feature deviation vectors as the inputs and the human ratings as

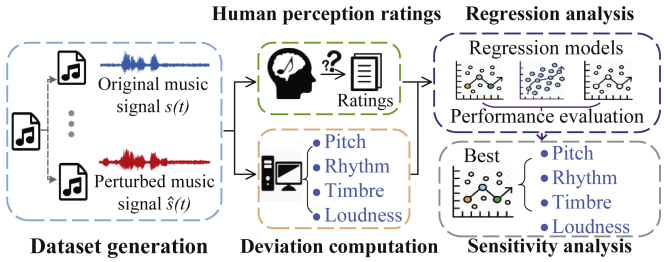


Fig. 5. The human study procedure and steps.

the outputs, we use regression analysis to find the best model to describe the relation between the vectors and the ratings. In this way, we can reverse-engineer the human perception process to build an approximation model to quantitatively predict how much a perturbed signal is perceived by a human.

Dataset Generations: Since there is no publicly available dataset that provides various versions of perturbed music signals, we propose to generate our own dataset with the following requirements: (i) sufficient diversity of music genres, (ii) sufficient perturbations from the pitch, rhythm, timbre, and loudness, and (iii) slight or moderate perturbation to avoid making participants feel overly noisy.

We build a dataset of 60 pairs of original and perturbed music clips from the genres of Pop, Hip-hop, Rock, Jazz, Classical, R&B, Country, and Disco. To make participants concentrate on each small perturbation, we crop each music clip to a 5-second WAV format (16 kHz, 16-bit PCM, Mono) to avoid audio compression. As there is no guideline or reference to standardize the dataset generation for our study, we aim to create perturbed signals with different feature deviations and varying intensities for human participants such that the data is diverse for regression analysis. We use two main mechanisms to create perturbed music clips.

Additive noise: An intuitive method is to inject additive noise into the original music. The noise will affect all four features at the same time. To broadly affect the original music, we consider injecting the noise from three aspects: amplitude, frequency and time. To control the amplitude of the noise, we can choose the signal-to-noise (SNR) level from 0 dB, 5 dB, 10 dB, and

15 dB [66]. To inject frequency-sensitive noise, we use both white noise [67] (covering all frequencies with equal intensity) and colored noise (with the power concentrated at certain frequencies). To make noise time-varying, we set random duration and interval of the noise, but the total injection duration is less than the half of the original music length. Since existing audio perturbations (e.g., in [13]) cause noise-like sounds, the noise data rated by participants should help build a model to properly predict the deviations of noise-like perturbations.

Additive notes: To ensure distinctive deviations among all music features, we also inject additive notes to the original music. To inject notes with the pitch manipulation, we randomly choose notes with the pitch value from 27.5 Hz to 4186 Hz [68] (88 notes space). For rhythm manipulation, we randomly select the additive notes with different lengths and ensure the intervals between adjacent notes are less than 50% of the original signal's length. To create timbre deviation, we select different instruments to play the additive notes as long as the notes are within the valid pitch ranges of those instruments.

Human Participation: We recruited 35 participants who are college students with ages falling between 20 and 35. All the participants are volunteers without any compensation. Each participant was asked to listen to each pair of the original and perturbed music clips, and then assign a deviation rating on a Likert scale [69] according to his/her overall music perception: $0-1$ perfect perceptual quality with imperceptible noise, $1-2$ good perceptual quality with quiet noise, $2-3$ noticeable with slight noise, $3-4$ noticeable and noisy, and $4-5$ very noisy. More specifically, $1-2$ means volunteers can only notice some small perturbation after listening to a part of music clips many times, and $2-3$ indicates the deviation can be noticed by listeners but not noisy. During the experiments, all the volunteers were given the same earphone with the same initial volume setting. They can listen to a music clip as many times as they want.

Ethical Considerations: Our study involved human participants that assigned ratings by listening to music. The full protocol was reviewed and exempted by our Institutional Review Board (IRB), which has determined that the study involves the minimal risk for human participants (i.e., the risk is no more than the one that they face during their daily lives). We follow the approved protocol to inform them of the full study procedure and protect their identities without publishing any personally identifiable information.

Reverse-Engineering via Regression Analysis: Given the computed feature deviations from the original and perturbed music clips as well as the human participant ratings of their perceived deviation, we aim to find the best regression model $M^* \in \mathcal{M}$ in the model set $\mathcal{M}$ to minimize the mean squared error (MSE) of regressed prediction, i.e.,

$$M^* = \arg\min_{M \in \mathcal{M}} \mathbb{E}\|r - M(d_p, d_r, d_t, d_l)\|_2^2, \tag{2}$$

where r is the human participant rating, d_p, d_r, d_t, and d_l are the deviation values (computed according to the procedure in Section IV-A) for pitch, rhythm, timbre, and loudness, respectively. In our study, we choose Linear Regression [23], [70], Support Vector Regression, Random Forest, Logistic Regression, and Bayesian Ridge to form the model set $\mathcal{M}$. With M^* found in (2), we use it to predict human-perceived deviation given a pair of original and perturbed signals.

TABLE I
MSEs OF DIFFERENT REGRESSION MODELS

Model:	Linear	SVR	Random Forest	Logistic	Bayesian
MSE:	1.2351	0.8558	**0.1541**	1.6572	1.2628

D. Result Analysis and Discussion

Fig. 6 box-plots all the human ratings (ranging from 0 to 5) for individual pairs of music clips from our human study. We can find in Fig. 6 that human perception is indeed subjective: each pair of music clips has a range of deviation ratings by different participants; there are always rating outliers for a pair of music clips. Fig. 6 also shows that overall, the ratings and the 25%-75% boxes are roughly evenly distributed from 0 to 5, which offers sufficient data diversity for regression analysis.

Regression Analysis: We first use each of Linear Regression, Support Vector Regression (SVR), Random Forest, Logistic Regression, and Bayesian Ridge to model the relationship between feature deviation values and the average human rating, and find the best model with the minimum MSE. The MSEs of different regression models are in Table I.

Through regression analysis, we find that Random Forest performs the best among all the five regression models. As Table I shows, Random Forest leads to an MSE of 0.1541, which is substantially better than Support Vector Regression that achieves the second with an MSE of 0.8558, but an over 5 times increase from Random Forest. The other models result in even worse MSEs. As a result, we choose Random Forest as our regression model to predict the human-perceived deviation. Specifically, given a pair of original and perturbed signals, we name the prediction output of Random Forest as quantified deviation (qDev).

Correlation Analysis: Then, we analyze to what extent qDev values and realistic human ratings move in tandem; that is, an increase or decrease of value for one will lead to the same for the other. This is important because when creating an adversarial attack against a classifier, we aim to reduce the qDev value of a perturbed signal (so its deviation rating by a human should also decrease) such that the perturbation is hardly noticed by a listener. We use Spearman's rank correlation coefficient [71], [72] to model the correlation in our study. Spearman's coefficient is a commonly used statistic measure to evaluate the relationship between two variables using a monotonic function, where value 1 or -1 indicates that the two always move in the same or opposite direction; value 0 means no correlation.

Table II lists the Spearman's coefficients between the human rating and each of the following deviation measures: L_2 norm [13], L_∞ norm [11], [13], SNR [7], [8], and qDev from Random Forest. It is seen from Table II that qDev has a very high correlation with the realistic human rating, indicating it can be quite useful for predicting a human-perceived deviation of a signal. In other words, minimizing qDev in a mathematical

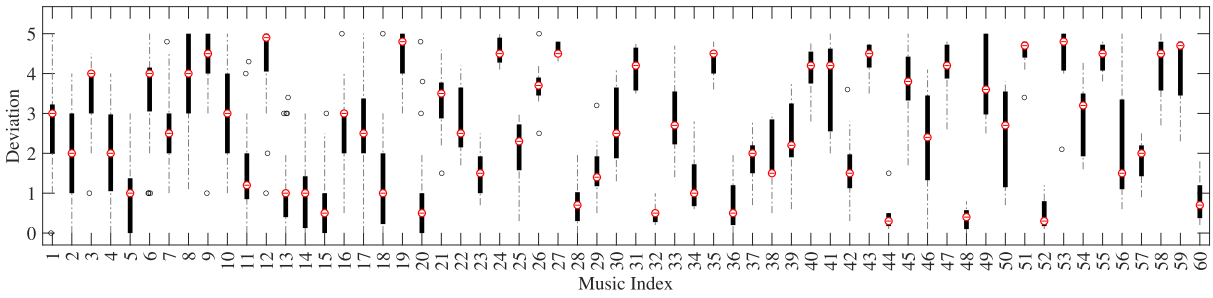


Fig. 6. Distributions of human ratings of perceived deviation for all pairs of music clips.

TABLE II
SPEARMAN'S COEFFICIENT BETWEEN THE HUMAN RATING AND A DEVIATION MEASURE

Deviation Measure:	L_2	L_∞	SNR	qDev
Spearman's Coefficient:	0.3909	0.0893	0.0134	**0.9608**

TABLE III
SENSITIVITY ANALYSIS FOR EACH FEATURE

Excluding:	Pitch	Rhythm	Timbre	Loudness	None
MSE:	0.1891	0.1581	0.1889	0.3539	0.1541

formulation to form an audio signal perturbation would be most likely suppress a human's attention to the signal deviation caused by the perturbation. Interestingly, we also observe that the commonly used L_p norms and SNR are in fact not well related to human perception (e.g. L_2 norm has the best correlation of 0.3909). Table II offers quantitative evidence to echo the concern raised in related studies [4], [9] that suggests new ways to measure the human perceptual similarity may be needed.

Sensitivity Analysis: To explore which feature is potentially more important than others in human perception, we conduct sensitivity analysis via the One-at-a-time (OAT) strategy [73], [74], [75]: we remove in turn pitch, rhythm, timbre, and loudness to form three-feature inputs for regression, and measure the MSE of the resultant regression. We find Random Forest is always the best in OAT analysis to minimize MSE with only three features reaming as the inputs.

Table III shows the MSE of Random Forest for each regression of excluding pitch, rhythm, timbre, and loudness in turn. From Table III, loudness that represents the energy of the perturbation appears to be the most sensitive feature to human-perceived deviation. For example, removing loudness leads to a 129% MSE increase from 0.1541 to 0.3539. But it is clear that the other features individually contribute to the overall human perception, and removing one of them causes more MSE in the regression.

Potential impact of sensitivity analysis on speech signal: The results of sensitive analysis in Table III could also provide us some insights into other audio scenarios, e.g., the adversarial audio in speech and speaker recognition. It is evident to reveal that loudness is the most sensitive factor in human perception, which is consistent with the findings in the existing work [8], [11] showing that the smaller L_p norm [11] and higher SNR [8] could lead to better perception. Beyond these findings, other auditory features also impacts the perception. For example, timbre can be an essential factor when generating the adversarial examples in the speaker recognition attack [76], and the pitch is also an important feature to preserve the quality of adversarial speech [77]. To further preserve the quality of the adversarial examples, it is still worth investigating the sensitivity of different features to the different applications, e.g., if the rhythm is not dominant to the speaker recognition, we can integrate other features to reduce the computational complexity. But the findings from the music signal clearly reveal that all the features are sensitive to human perception, and we should consider integrating them into our design.

Overall, we find in the human study that Random Forest is the best regression model to yield the minimum MSE to predict the human rating as qDev. Simpler regression models, such as Linear Regression or SVR, do not perform as well as Random Forest. This may confirm that human perception is indeed a complicated process. In addition, qDev is a much more appropriate metric than the L_p norm or SNR in terms of both MSE and Spearman's correlation with the human rating, and the features of pitch, rhythm, timbre, loudness all contribute to the overall perception.

V. PERCEPTION-AWARE ATTACK STRATEGIES

In this section, we formulate the perception-aware attack strategies. We describe how we formulate the new strategies going beyond our conference version [1] by creating a differentiable human perception modeling and adopting multiple musical features into the attack formulation.

A. Basic Formulation

A straightforward formulation for the perception-aware attack is to simply replace the L_p norm with the newly defined qDev metric in (1), which has been done in our conference version [1]. However, the computational solution to such a formulation is a brute-force search in a heuristically narrowed search space. This is due to the facts that (i) the Random Forest based qDev modeling is not differentiable and (ii) the qDev modeling is considered separately with the loss function of the learning model in [1]. In this journal version, we re-formualte the attack strategy by integrating the qDev model into the loss function as

$$\arg\min_{\delta} \mathcal{J}(\hat{s}(t)) + c\,\mathrm{qDev}(s(t), \hat{s}(t)), \tag{3}$$

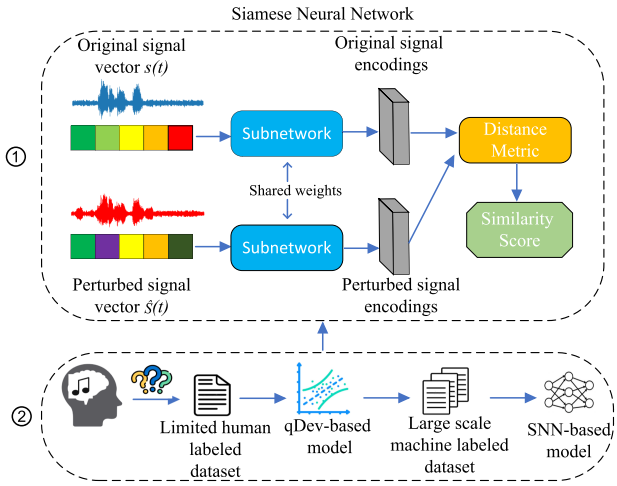


Fig. 7. The architecture and the training process of the Siamese Neural Network.

where $\mathcal{J}(\hat{s}(t))$ represents the loss of the surrogate model predict $\hat{s}(t)$ as other labels, $\text{qDev}(s(t), \hat{s}(t))$ denotes the qDev between the perturbed signal $\hat{s}(t) = s(t) + \delta(t)$ and the original one $s(t)$, and c is a constant to balance the attack effectiveness and human perception. To ensure $\hat{s}(t)$ is valid waveform, we always constrain the normalized amplitude of each of its sample points to be in $[-1, 1]$ [11].

B. Differentiable Human Perception Model Design

A technical challenge to create an efficient solution to (3) is to design a differentiable model of qDev. As shown in Table I, the Random Forest modeling of human perception has the best MSE performance but it is not differentiable. It may be easy to simply replace the Random Forest Model with a differentiable one such as Linear Regression or Support Vector Regression. However, Table I shows that their MSEs are substantially worse than Random Forest.

1) Motivation of Using Siamese Network Architecture: We consider making the Random Forest-based qDev model differentiable, thereby solving the computational efficiency problem. However, building a new computational efficiency model could impact the prediction accuracy, i.e., involving higher MSE. We need to balance its efficiency and MSE when building the new human perception model. A model that is efficient and accurate is beneficial in the long term as it can facilitate complex optimization and is likely to be in high demand for future research. To build a new human perception model, there are two key requirements we need to meet: 1) the model can take two audio signals as the input and compare the similarity, and 2) the model is differentiable and is able to achieve computational efficiency.

Siamese Neural Networks (SNN): Jointly considering these requirements, the Siamese Neural Network [78], [79], [80] (SNN) can perfectly match all these requirements, which can measure the similarity between two inputs via the same configuration subnetworks.

First, as shown in Fig. 7(1), the SNN consists of two identical subnetworks and takes in two separate inputs, such as two images [78], [79], [80] or text [81], [82], [83] vectors, and can be used to process audio signals [84], [85], [86] as well.

Second, these input vectors pass through respective layers to produce encoded feature vectors. The process is differentiable and convenient for finding gradient, which brings significant improvements in computational efficiency.

Last, the SNN model forwards the time-domain signal vectors to the encoding procedure to compute the similarity score. This approach is faster than the regression methods used in the previous section, which involves more time-consuming music feature extraction, specifically for pitch estimation. The main reason is that most music features are calculated in the frequency-domain, resulting in more complex computations compared to that in the time-domain.

Overall, SNN can meet all the requirements of building a new perception model, and it is differentiable to solve the computational efficiency problem. We find a way to convert the Random-Forest-based model into a new SNN model.

2) Training the SNN Model: Our main challenge is to establish a workable approach for constructing a suitable training dataset for SNN. Previously, the qDev [1] is trained with 60 pairs of the original and perturbed signal, such a size of training data is not feasible to train the deep neural network. However, there is no large-scale human-labeled music dataset. The direct way is to recruit more volunteers to help us rate the deviation scores. Nonetheless, the human study is cumbersome for volunteers and always has a problem to scale (e.g., each participant commonly takes 40 to 60 minutes for 60 pairs of music). It is practically difficult, if not impossible, to obtain a very large human-labeled dataset (e.g., thousands of pairs of music). To solve this challenge, we propose to use the qDev model to substitute the human to create large labeled training sets for SNN training. There are two advantages to this method: i) utilizing the qDev model does not require any extra human effort and can save time on dataset creation; ii) our conference version [1] found that Random Forest-based qDev exhibits the best MSE performance during both the training and testing phases even with different human evaluation groups. As a result, it should be acceptable to use Random Forest-based qDev modeling to construct a novel differentiable SNN model that offers improved computational efficiency for adversarial attack generation.

Specifically, we generate perturbed music clips from 8 different genres and manipulate the music clips by adding Gaussian noises, changing the pitch, rhythm, and timbre to the original music according to the same procedure in our human study. As shown in Fig. 7(2), we use the Random Forest-based qDev model to predict the deviation score from 0 to 5 to label the training data. We collect 1000 pairs of original and perturbed to train the Siamese networks. During the training, we use the sample rate of the input signal as 16 kHz, and fix the length as 30 seconds. The fixed-length input vectors are forwarded into two identical subnetworks (e.g., same parameters), then calculate the loss between the two inputs vectors, and compute the gradients of the model during backpropagation to update the weights.

3) MSE Performance of SNN Modeling: We aim to evaluate the performance of the SNN model trained by the Random

TABLE IV
MSEs OF DIFFERENT REGRESSION MODELS

Model:	Linear	SVR	Random Forest
MSE:	1.8326	2.3515	0.4916
Model:	Logistic	Bayesian	SNN
MSE:	2.1623	1.6857	0.6596

Forest model in comparison with other regression models using human ratings obtained in our human study. Table. IV shows the MSE performance of the trained SNN and other different regression models. We can clearly see that SNN achieves an MSE of 0.6596, which is lower than Linear regression, SVR, Logistic regression, or Bayesian regression. Notably, the SNN model performs comparably to the Random Forest which has the lowest MSE of 0.4916. Therefore, we find that the SNN model can learn well from the Random Forest-based qDev model and also strikes a good balance between the differentiability and the MSE performance. Table. IV shows that the differentiable SNN model has the potential to efficiently solve the optimization problem involving human rating prediction.

C. Computationally Efficient, Multiple Features Based Perception-Aware Attack Formulation

Our conference version [1] reduces the search space to find the solution in (1) by only considering manipulating the timbre feature of a music signal to find the perturbation signal, because of the non-differentiable qDev model. A larger manipulation space can enable us to find a better perturbation signal that can satisfy both attack effectiveness and human perception for the formulation in (3). Now our proposed SNN model is computationally efficient and accurate to predict the perturbed music quality. It allows us to explore a larger manipulation space to find the solution in (3). In other words, for a music signal, we aim to manipulate its pitch, rhythm, and timbre features at the same time to find an effective attack signal with good perceptual quality.

To manipulate $s(t)$ with multiple features, we first get a timbre-perturbed signal $s(t)+\delta(t)$, where the additive perturbation signal $\delta(t)=\sum_{k=1}^{K}\theta_k\delta_k(t)$ and $\delta_k(t)$ is the k-th instrumental track signal playing the same music notes by a different instrument; θ_k is non-negative linear weight for $\delta_k(t)$. We modify pitch and rhythm of the timbre-perturbed signal $s(t)+\delta(t)$ by values γ and τ, respectively, to form the perturbation signal $\hat{s}(t)$, where γ measures the pitch value in semitones and τ quantifies the rhythm in units of time. We formulate the computationally efficient, multiple features based perception-aware (CEMF-PA) attack as

$$\arg\min_{\hat{s}(t)} \quad \mathcal{S}(s(t),\hat{s}(t))+c\,\mathrm{SNN}(s(t),\hat{s}(t)) \tag{4}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\theta_k=\epsilon_\theta \tag{5}$$

$$|\gamma|<\epsilon_\gamma \tag{6}$$

$$\epsilon_{\tau\min}<\tau<\epsilon_{\tau\max}, \tag{7}$$

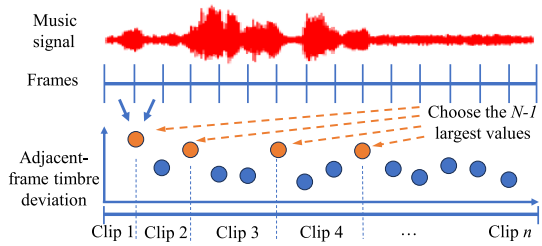


Fig. 8. Overview of dynamic clipping.

where $\mathcal{S}(\cdot,\cdot)$ denotes the audio fingerprinting based similarity score based on the target audio classifier; $\mathrm{SNN}(\cdot,\cdot)$ indicates the SNN score that represents the human perception quality, constant c is the balance factor between the attack effectiveness and human perception; ϵ_θ is the loudness threshold of θ_k; ϵ_γ and define the manipulation ranges of pitch, and $\epsilon_{\tau\min}$ and $\epsilon_{\tau\min}$ denote the minimal and maximum value to the changed rhythm, respectively. The optimization (4) is a problem of finding the optimal linear weights θ_k, pitch change γ, and rhythm change τ, which can be solved efficiently by iterative optimization algorithms.

D. Dynamic Clipping

The optimization in (4) finds out a perturbed signal $\hat{s}(t)$ based on the entire duration of the original signal $s(t)$. However, a piece of music can consist of multiple segments with audio characteristics varying within a wide range of pitch, rhythm, instruments, and vocals. For better perceptual quality and attack effectiveness, it is necessary to segment $s(t)$ into N clips. Since pitch and rhythm deviations generally lead to timbre deviation [24], [70], overall timbre change is a good factor to crop the music clips. Therefore, we clip the music signal based on evident timbre changes and create the perturbation for each clip using the clip-wise optimization based on (4). We call this procedure dynamic clipping.

Fig. 8 shows the overall process of dynamic clipping: in order to dynamically segment $s(t)$ into N clips, we first separate $s(t)$ into small frames and compute the timbre deviation between each pair of adjacent frames (using the timbre deviation calculation discussed in Section IV-A). We identify $N-1$ pairs which have the $N-1$ largest adjacent-frame deviation values, as they contain the most evident $N-1$ changes of timbre over the duration of music. We use the timing boundary between two frames in a pair as a timing position to segment $s(t)$. Thus, $s(t)$ is segmented into N clips and each clips is manipulated via shifting the pitch, changing the rhythm, and the timbre based on (4).

VI. REALISTIC BLACK-BOX ATTACK AGAINST COPYRIGHT DETECTOR

In this section, we create a realistic attack based on the perception-aware attack framework in Section V. We choose the YouTube copyright detector as our target as YouTube has exhibited some robustness against noise and perturbations [13].

Because there is no knowledge of YouTube's design, we create our own detector based on open-source information for an adversarial transfer attack. We first present how to generate additional instrumental tracks for the perturbation signal given a music signal, then describe the design of our detector as a surrogate model for YouTube's detector.

A. Perturbed Signal Generation

Perturbed signals generated by (4) require the detailed music notes of the original music. For a popular piece of music, its Musical Instrument Digital Interface (MIDI) file is usually available in online databases (e.g., FreeMidi.org[1] and Nonstop2k[2]). The MIDI file contains all instrumental tracks with music notes. We use Music21[3] to play a downloaded MIDI file with different instruments to form a perturbation. To achieve the diversity of the timbre feature, we consider an instrument set of instruments across the four families *stringed* (Guitar, Electric Guitar, Violin, Viola, Cello, Bass, Electric Bass), *woodwind* (Clarinet, Flute, Saxophone, Oboe, Bassoon), *brass* (Trumpet, Baritone, Tuba, Horn, Trombone), *keyboard* (Piano, Electric Piano). We empirically select at most two instruments from each family based on a music genre to reduce the computational complexity.

B. Surrogate Detector

Audio Fingerprints: A copyright detector takes audio fingerprinting features as the input. We select the fingerprints and their extraction method introduced in [51]. We extract fingerprints by considering the time, frequency, and amplitude data of the audio. Specifically, we use Fast Fourier Transform (FFT) to generate a spectrogram of an audio signal and extract the spectral peaks of acoustic harmonics, which are shown invariant and reproducible from signal degradation [87] and robust to distortion [51]. We apply the fast combinatorial hashing method [51] to form these fingerprints to hashes for the similarity comparison later.

Detection Design: The detection is built to compute the similarity of the fingerprints of an input signal to the detector's database. If the similarity score is higher than a similarity threshold, the detector will raise an alarm. To ensure our surrogate detector has a degree of transferability to YouTube's detector, we must adopt a threshold that is similar to YouTube's. We note that our objective is not to precisely rebuild YouTube's model, but to choose an appropriate threshold such that we can use the surrogate detector to predict the output label during the optimization in (4). Because music consists of diversities of audio features, we choose one threshold for each of 8 music genres: Pop, Hip-hop, Rock, Classical, Jazz, R&B, Country, and Disco.

Fig. 9 shows the process we use to approximately calibrate the surrogate detector's threshold towards YouTube's. This process is similar to the one proposed in [11] that estimates the threshold of a black-box model. In particular, to obtain the threshold for a music genre, we choose a song from the genre, crop it into clips, choose the most representative clip that contains the highest number of fingerprints among all the clips. Then, we randomly add instrumental track signals with different energy levels, shift the pitch and change the tempo of the signal. We send these clips to YouTube to see the copyright detection results, and set the detection threshold for the surrogate detector such that it yields the same results as YouTube does.

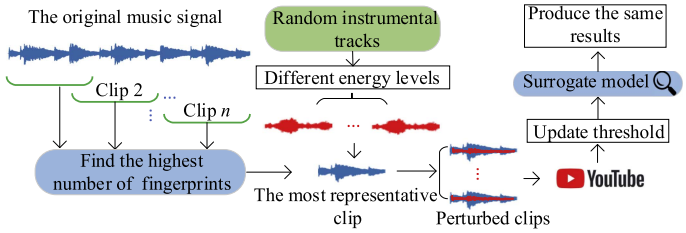


Fig. 9. Process of obtaining the threshold from YouTube.

VII. Experiments and Results

In this section, we present the experiment results. We decide the manipulation ranges of pitch and rhythm in the attack Formulation (4), then present attack effectiveness and perception quality of adversarial music against YouTube.

A. Deciding Search Ranges of Musical Features

In our attack Formulation (4), ϵ_θ denotes the overall energy level of perturbation signal while ϵ_γ, $\epsilon_{\tau\min}$ and $\epsilon_{\tau\max}$ decide the ranges of pitch and rhythm manipulations, respectively. Finding appropriate search ranges of γ and τ can help further reduce the search complexity in (4). There are mainly two factors to decide the manipulation range, (i) attack success rate (ASR) that represents the probability of a perturbed music signal bypassing YouTube's copyright detection and (ii) human perception of the perturbed music. Our goal is to change $s(t)$ with different pitch and rhythm deviations to generate various versions of $\hat{s}(t)$ to measure the ASR against YouTube as well as the perception quality of $\hat{s}(t)$.

In this set of experiments, we totally select 80 30-second music clips for pitch and rhythm manipulations. Specifically, we adjust the pitch by shifting music up or down within a range of 15 semitones. For rhythm manipulation, we change the speed of the music, ranging from $1/15$ to 15 times its original tempo. Then, we upload all manipulated music clips to YouTube to measure the ASR.

1) ASR of Manipulating Pitch and Rhythm: Fig. 10(a) and (b) show the effectiveness of attacking YouTube via manipulating pitch and rhythm, respectively. Fig. 10(a) shows that pitch manipulation achieves nearly a 50% ASR when shifting up and down by 9 semitones, and leads to nearly 100% by over 13 semitones. Fig. 10(b) illustrate that the ASR by manipulating rhythm goes above 50% when the rhythm is speeded up by 5 times or slowed down by 1/5, and increases to 100% with more speed-up or slow-down. Overall, either pitch or rhythm manipulation is effective if the change is sufficient.

[1]https://freemidi.org/
[2]https://www.nonstop2k.com/
[3]Music21 is a Python toolkit for computer-aided musicology. We use it to produce instrumental tracks playing the same musical notes

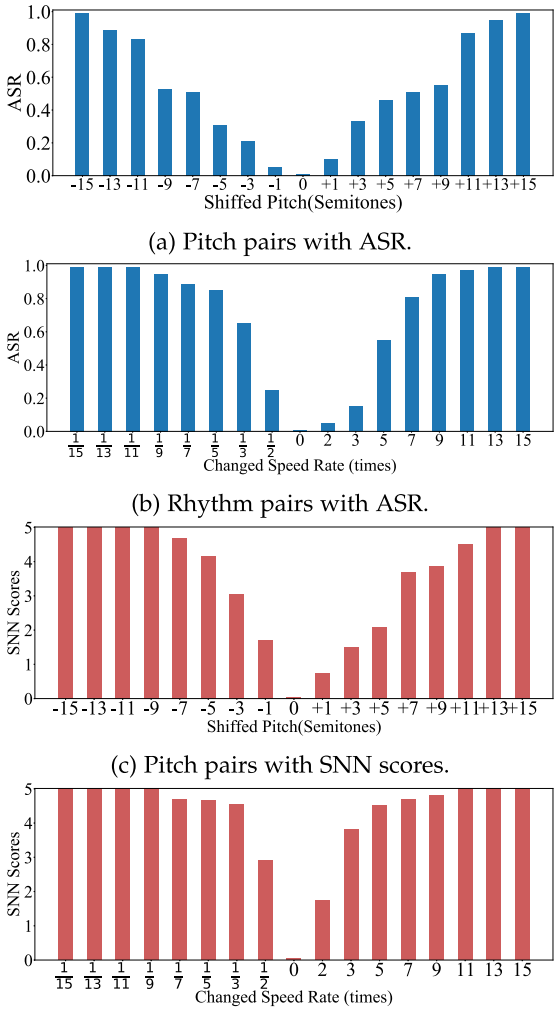


Fig. 10. The ASR and SNN scores due to pitch and rhythm manipulation.

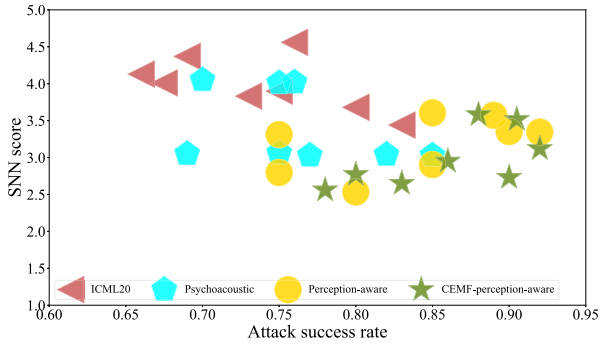


Fig. 11. Evaluation of the CEMF-PA attack on YouTube: the distribution of ASR and SNN scores.

However, the ASR results in Fig. 10(a) and (b) cannot help us to determine manipulation range, because we need to consider human perception aspect after the manipulation.

2) Perceptual Quality Due to Pitch and Rhythm Manipulations: Fig. 10(c) and (d) show the perceptual quality measured by the SNN score due to pitch and rhythm manipulations, respectively. We can observe a correlation between the changed value of pitch/rhythm and the SNN score: a larger deviation incurs a greater SNN score (i.e., indicating lower perceptual quality). For example, the SNN scores are 3 to 5 when we shift the pitch of the original music signal by over 5 semitones, where the ASR is below 50%. A similar situation can also be observed in rhythm manipulation.

3) Determining the Manipulation Range: A good manipulation space for pitch and rhythm should allow the perturbed music signal $\hat{s}(t)$ to spoof the classifier and preserve human perception at the same time. As we discussed earlier from Fig. 10, there is a conflict between human perception and attack effectiveness during manipulating pitch and rhythm. A large manipulation space could provide a sufficient diversity of the perturbation to bypass the detector, but it also involves more computational complexity. Considering both the attack performance (attack effectiveness and human perception in Fig. 10) and the computational cost, we choose the pitch manipulation range to be from shifting down by 5 semitones to shifting up by $+5$ semitones, and the rhythm manipulation range to be from slowing down by $1/3$ to speeding up by 3 times.

B. Results of CEMF-PA Against YouTube

After choosing the search range of pitch and rhythm, we measure the performance of the formulated CEMF-PA attack against YouTube copyright detector. To cover a wide range of music data, we selected 80 songs from 8 genres: Pop, Hip-hop, Rock, Classical, Jazz, R&B, Country, and Disco. We created 160 clips of 30 seconds for attack strength evaluation. We have verified that all the clips were copyright-detected by YouTube.

The default settings in (4) for our CEMF-PA attack include the number of instruments for perturbation generation $K=7$, pitch shift is from -5 to $+5$ semitones and rhythm change is from $1/3$ to 3 times. And the number of clips in dynamic clipping $N=6$.

We compare the CEMF-PA attack with three recent attack methods: the ICML20 method against YouTube in [13], the psychoacoustic attack [9], and the original perception-aware (PA) attack proposed in our conference version [1]. We create 480 adversarial clips of 30 seconds (120 clips from 8 genres by each method) and uploaded them to private YouTube channel for YouTube's copyright detection.

1) Attack Effectiveness vs Perceptual Quality: It is clear that we can always get a 100% attack success rate by generating a sufficiently large perturbation and adding it to the original music, which can, unfortunately, produce extremely noisy sound. Hence, it is necessary to pair attack effectiveness with perceptual quality.

For each genre of testing music clips, we measure the pair of ASR and SNN score for each method and draw the pair as a marker in a 2-D figure shown in Fig. 11 It can be observed that the ICML20 and psychoacoustic attacks exhibit low ASRs (i.e., attack is not effective) and high SNN score (low human perceptual quality) compared with the CEMF-PA attack. We also find in Fig. 11 that the proposed CEMF-PA attack generally achieves

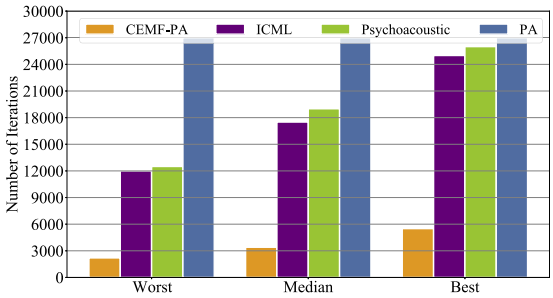


Fig. 12. The number of iterations of different attacks at various representative points.

moderately better ASRs and SNN scores than the original PA attack. The results in Fig. 11 demonstrates that the CEMF-PA attack is moderately better than the original PA attack, and substantially superior to ICML20 and psychoacoustic attacks.

2) Comparisons of Computational Efficiency: Another advantage of the proposed CEMF-PA attack is its computational efficiency. We aim to know whether the CEMF-PA attack is not only effective but also feasible to reduce the cost of generating an adversarial signal with good quality. Specifically, we compare the number of iterations used for perturbation generation in the experiments.

Fig. 12 compared the numbers of iterations needed in different attack generation methods. We consider analyzing the average number of iterations for each attack method under the cases that it achieves the worst, median and best ASR in Fig. 11 (e.g., the worst case for ICML20 is represented by the leftmost marker in Fig. 11 with 66% ASR for the music genre of Rock). It is noted from the figure that the CEMF-PA attack is the most computational efficiency algorithm in terms of the number of iterations compared with other generation methods (e.g., 3200 iterations for CEMF-PA vs 17000 iterations for ICML20in the median case). We find that the PA attack uses the grid search with fixed steps to create adversarial music, therefore, it always requires a fixed number of runs to find the solution (e.g., nearly 27000 iterations in Fig. 12). The CEMF-PA attack is much more computationally efficient than the PA attack while obtaining moderately better ASR and SNN score performance.

Potential extension of CEMF-PA: Since this study mainly focuses on generating well-perceived adversarial music signals against music copyright detection. The core of our attack strategy is (i) building a human perception model (e.g., SNN model) that can quantify the music quality and (ii) leveraging the human perception model to find a reasonable manipulation space of the perturbation to ensure the attack effectiveness and computational efficiency at the same time. We think the CEMF-PA strategy can be extended to other audio applications, e.g., crafting good perception adversarial speech against speech and speaker recognition systems. Specifically, it is necessary to rebuild the SNN model based on the human study of perturbed speech signals, e.g., asking participants to assess the similarity between the original and perturbed speech samples during the speaker recognition task, or having them evaluate the intelligibility of the original and perturbed samples in the speech recognition task. Meanwhile, understanding the contributions of various auditory features to human perception is also essential to the CEMF-PA attack. It should be a good strategy to shrink the manipulation space to improve computational efficiency.

3) Vulnerability Disclosure: The proposed perception-aware attack does not cause an immediate operational impact, such as denial of service. Following the practice of responsible disclosure, we reported the issue of music copyright detection to Google. Google initially classified the case as an abuse risk. During the communication, Google mentioned that a copyright content will be taken down from YouTube when the copyright owner makes a request. Google eventually made the decision not to track it as a security bug.

VIII. Defense Strategy Design

In this section, we propose defense strategies against perception-aware attacks.

A. CEMF-PA-Based Adversarial Fingerprinting

One of the most adopted strategies from the machine learning community is adversarial training [2], [25], [26], [27], [28], [29], [30]. Adversarial training primarily focuses on making a machine learning model robust to the adversaries via solving a min-max optimization problem that finds the model parameters to minimize the cost results from strong adversary examples. Given a bounded L_p ball, the re-trained model becomes robust against adversarial attacks. However, the PA attacker uses the human perception model instead of L_p norm to craft adversarial examples. Therefore, this creates a model mismatch [88] and can make the re-trained model ill-suited.

Motivated by the idea of adversarial training, we aim to train the audio fingerprinting model with CEMF-PA based adversarial examples to make audio fingerprinting more robust. Specifically, we can make the audio fingerprinting model store fingerprints of all such CEMF-PA based adversarial music with SNN score no greater than a threshold κ. Denote by $\mathcal{A}(s(t))$ the set of the fingerprints of adversarial signals with SNN score no greater than κ, we have

$$\text{objective} \qquad M(\mathcal{A}(s(t)), \hat{s}(t)) = L(s(t)) \tag{8}$$

$$\text{subject to} \qquad \text{SNN}(s(t), \hat{s}(t)) \leq \kappa, \tag{9}$$

where $M(\cdot,\cdot)$ represents the match function which can compare the audio fingerprints between $\mathcal{A}(s(t))$ and adversarial music $\hat{s}(t)$, and output the label decision, $L(s(t))$ indicates the label of $s(t)$. (9) limits the SNN score of adversarial music is no greater than κ.

To set an appropriate value of κ, we should consider both attack effectiveness and real-world human perception of the music. If we take a look at the distribution of the ASR and SNN scores in Fig. 11, we can find that the SNN scores representing human perception mainly ranges from 2.0 (noticeable with slight noise) to 4.0 (noticeable and noisy). Thus, we set $\kappa = 4.0$ as the maximum adversarial fingerprinting threshold to indicate that the adversarial music with the SNN score greater than 4.0 is too noisy to be included.

B. Generalized Adversarial Fingerprinting

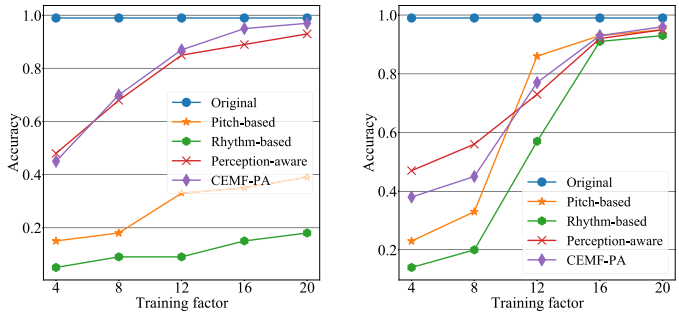
We aim to store the set of adversarial music $\mathcal{A}(s(t))$ to make the audio fingerprinting model against the adversarial examples from attackers. But there could be a mismatch between the defender's training data and the attackers' adversarial examples; because we cannot assume all the attackers using PA-based or CEMF-PA-based attacks with the same parameters. Given the fact that $\mathcal{A}(s(t))$ can include infinite values of fingerprints, the problem of creating a practical defense is how to build a generalized version of $\mathcal{A}(s(t))$ with a finite number of fingerprints that can effectively defend against adversarial signals generated by different methods within the perception threshold κ. Intuitively, $\mathcal{A}(s(t))$ should build with different manipulation methods, because different attack methods could change the audio fingerprints quite differently, i.e., pitch and rhythm-based manipulation can revise the fingerprints in different ways, and it is necessary to build a generalized version of $\mathcal{A}(s(t))$ that cover different fingerprints. We aim to explore the effectiveness of building $A(s(t))$ via ensemble learning that combines different adversarial generation methods.

To this end, we aim to explore whether ensemble learning different manipulation methods can improve the generalizability of creating $\mathcal{A}(s(t))$. We adopt different manipulation method to generate adversarial training samples, we first can make use of CEMF-PA, which jointly covered three musical features. Then, we propose to only change one feature at a time in (4), e.g., only change the pitch-shift variable γ and fix other feature variables to solve (4). And we use the same manipulation range introduced in Section VII-B to build ensemble-learning-based fingerprints. Specifically, we choose 7 different instrument tracks to revise the timbre feature, i.e., $K=7$, and we set the manipulation range of shifted pitch from -5 to $+5$ semitones and change the rhythm from $1/3$ to 3 times. In this way, we can get the pitch, rhythm, PA, and CEMF-PA-based adversarial signals to compose $\mathcal{A}(s(t))$, and we use an equal number of training samples for each method to construct $\mathcal{A}(s(t))$.

C. Effectiveness of Adversarial Fingerprinting

We measure the effectiveness of the defense provided by the proposed adversarial fingerprinting model. We choose 32 pieces of 30-second music clips from 8 different music genres. Since we cannot apply the defense to YouTube's closed-source copyright detector, we test our local audio fingerprinting detector based the fingerprinting model [51] discussed in Section VI. In experiments, we create the ensemble-based method which combines the pitch-, rhythm-, PA-, CEMF-PA-based generations to form $\mathcal{A}(s(t))$. For performance comparison purposes, we also compare the performance of the ensemble-based method with CEMF-PA-based adversarial fingerprinting that only trains adversarial signals based on the CEMF-PA attack. We set the perception threshold $\kappa = 4.0$. We collect 160 generated adversarial signals to form $\mathcal{A}(s(t))$ for each method.

1) Generating Adversarial Music Signals: We create adversarial music signals of various types and send them to both the CEMF-PA adversarial fingerprinted model and the ensemble-learning-based model to assess whether these models



(a) CEMF-PA-based adversarial fingerprinting.

(b) Ensemble-based adversarial fingerprinting.

Fig. 13. CEMF-PA-based vs ensemble-based adversarial fingerprinting under different attack generation methods.

can accurately detect these signals. From a practical perspective, attackers have no limited manipulation range when generating adversarial examples. Therefore, we consider expanding the manipulation range for each feature to create adversarial signals:

- CEMF-PA: We employ CEMF-PA to jointly manipulate the timbre, pitch, and rhythm features to craft the adversarial music. Specifically, we use 9 different instrument tracks to revise the timbre feature, i.e., $K=9$, and we set the manipulation range of shifted pitch from -7 to $+7$ semitones and change the rhythm from $1/5$ to 5 times.
- Pitch manipulation: We only modify the pitch feature based on (4) to create adversarial music signals, with the shifted pitch ranging from -7 to $+7$ semitones.
- Rhythm manipulation: We solely manipulate the rhythm feature based on (4), with a manipulation range from $1/5$ to 5 times.
- Timbre manipulation: We adopt the PA attack from our conference version [1] to create adversarial signals, selecting 9 instrument tracks to revise the timbre feature.

We collect 160 adversarial examples for each method to test with different defense methods. In addition, we also use the original music signal to test the accuracy of the fingerprinting model without any attack.

2) Results Analysis: Fig. 13 shows the effectiveness (in terms of detection accuracy) of CEMF-PA-based and ensemble-based adversarial fingerprinting models that detect adversarial music signals produced by different attack generation methods. The training factor in Fig. 13 indicates the size of $\mathcal{A}(s(t))$ for each signal $s(t)$ (i.e., for each original music signal, how many adversarial signals we need to generate based on it and add their adversarial fingerprints to the fingerprint database).

Fig. 13(a) shows the CEMF-PA-based adversarial fingerprinting has high accuracies to detect the PA and CEMF-PA based attack generations when the training factor approaches 16. We can also observe that the CEMF-PA-based fingerprinting model is not effective in detecting the pitch- and rhythm-based attacks with the highest accuracy to be 40% when the training factor is 20. By contrast, the ensemble-based adversarial fingerprinting method, as shown in Fig. 13(b), is substantially better than the CEMF-PA-based method against various attack generations. It

is noted from Fig. 13(b) that the accuracy of the ensemble-based method is low when the training factor is small, but the accuracy rises to nearly 95% when the training factor reaches 20. Overall, the ensemble-based method with sufficient training is more effective against the adversarial music signals.

We can also see from Fig. 13(a) and (b) that the training factor does not affect the accuracy of detecting the original music, which is always 100%. This indicates that both CEMF-PA-based and ensemble-based methods are accurate models in the presence of no adversarial music.

D. Comparing Ensemble-Learning-Based Adversarial Fingerprinting With Existing Defense Methods

We evaluate the effectiveness of ensemble-learning-based adversarial fingerprinting with four different major types of defense strategies:

Traditional L_p-norm-based adversarial training: The key idea of L_p-norm-based adversarial training [2], [16], [25], [26], [27], [28], [29], [30] to re-train the model with adversarial examples within an L_p ball [16], thereby making the model more robust against a similar range of potential adversarial attacks.

Commonly used audio preprocessing approaches against adversarial audio: Down-sampling [16], [37], quantization [38] and low-pass filtering [17]. Audio preprocessing methods [16], [17], [37], [38] aim to modify the audio signals before forwarding it to the model, and the fundamental concept behind audio-preprocessing is disrupting adversarial perturbations while maintaining essential audio characteristics.

Recent audio watermarking methods: Timbre Watermarking [89] and AudioSeal [90], are employed to protect the copyright of digital audio. This technology involves embedding hidden digital information into the audio signal [89], [90] and subsequently extracting it to verify the copyright of the audio content released to the public. Audio watermarking has been explored in various studies [91], [92], [93] to protect the copyright of the audio signals. Audio watermarking methods not only preserve the quality of copyrighted audio but also exhibit robustness against signal processing methods [90] and complex generative models [89].

State-of-the-art diffusion-model-based defense method: AudioPure [94], effectively mitigates the adversarial attacks in the audio domain. The basic idea behind AudioPure leverages the advantage of the diffusion model to purify the embedded perturbation in adversarial audio signals. Specifically, there are mainly two components of AudioPure. In the forward diffusion process, it first adds a small amount of noise to the adversarial audio until the distribution of additional noise converges to a standard Gaussian distribution. Then, in the reverse sampling process, it uses reverse sampling to purify the noisy audio and recover the clean audio data. This method has demonstrated that it can outperform the L_p-norm-based adversarial training against diverse adversarial attacks (e.g., L_2 norm-based attacks).

We mainly consider these four categories of defense methods to evaluate the defensive strength of the ensemble-learning-based adversarial fingerprinting.

1) Experimental Setups: Here, we present the setups of existing defense methods.

L_p-norm-based adversarial training: In contrast to our defense strategy that involves ensemble learning of auditory-feature-manipulated music signals, L_p-norm-based adversarial training [16] uses highly effective adversarial examples to re-train the model within an L_p ball. To adapt the L_p-norm defense method to audio fingerprinting, we propose to collect these highly effective L_p norm adversarial samples in $\mathcal{A}(s(t))$. Specifically, we employ the ICML20 attack to generate the L_p-norm-based adversarial examples to build the audio fingerprinting set $\mathcal{A}(s(t))$. Following the settings of ICML20 as described by [13], we select the L_2 norm as the perturbation norm and set the L_2 norm of the perturbation signal is no greater than 0.05.

Audio preprocessing approaches: i) Audio down-sampling: this method has been validated its effectiveness against adversarial speech signals in the recent studies [7], [8]. We follow a similar step in [7] to reduce the sample rate of original adversarial signals by 50% from 16000 to 8000 samples / second. ii) Audio quantization: this approach [38] aims to disrupt the perturbation via reducing the precision of the audio signal, i.e., mapping the amplitude to a limited set of discrete values. We employ the same setting with [17], [38] to set the quantization parameter as 256. iii) Low-pass filtering: the primary goal of this method is retaining the components with lower frequencies and removing the high-frequency components from perturbation. We cut off adversarial audio signal using a 2 kHz low-pass filter [17].

Audio watermarking detection methods:

i) Timbre watermarking [89]: this method embeds the spectrogram-based watermarking to detect the copyrighted audio. Timbre watermarking demonstrated high detection accuracy against common speech processing methods, such as amplitude scaling, MP3 compression, and Gaussian noise. We use the source code [89] to implement the watermarking embedding model on the music dataset. Specifically, we set the bit recovery accuracy threshold [89] for the extracted watermark at 0.5, i.e., when the bit recovery accuracy reaches 0.5, we consider this music clip to be copyright-claimed.

ii) AudioSeal [90]: this approach can precisely and robustly localize embedded watermarks in long audio signals to identify the copyrighted content. It exhibits high robustness against audio manipulation methods, such as resampling and MP3 compression. Additionally, the embedded watermark is designed to be imperceptible, ensuring the preservation of audio quality. We directly implemented their source code and set the probability threshold of the detector at 0.4 for the presence of a watermark in the input music clip. Thus, AudioSeal will identify a music clip as copyrighted when the detector's output probability reaches 0.4.

AudioPure [94], is a diffusion-model-based defense method that utilizes a pre-trained diffusion model to purify perturbations in adversarial audio signals. There are mainly two components of AudioPure: the forward diffusion process and the reverse sampling process. In the forward diffusion process, AudioPure adds some additive noise to the adversarial audio samples. Then, through the reverse sampling process, AudioPure purifies these

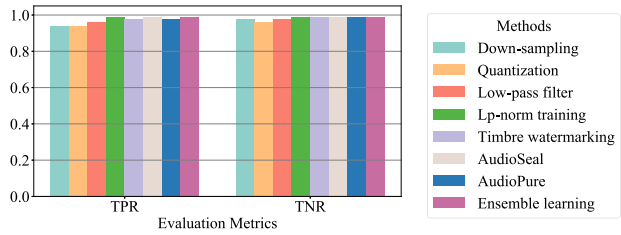


Fig. 14. The baseline classification performance of different defense methods on the clean music datasets.

noise-based adversarial examples to recover the clean audio. Following the setups described in [94], we employ DiffWave [95] and DiffSpec [96] as the defensive purifiers. AudioPure is a plug-and-play approach that can be directly integrated into the audio fingerprinting systems.

Ensemble-based adversarial fingerprinting: We compare these methods with ensemble-based adversarial fingerprinting models using the same procedure as outlined in Section VIII-B. We choose the perception threshold $\kappa = 4.0$, and set the training factor as 20 for our method and L_p norm-based adversarial training. This factor indicates the size of $\mathcal{A}(s(t))$ for each signal $s(t)$ is 20. And we choose 32 pieces of 30-second music clips from 8 genres as our original music set.

2) The Baseline Performance of Different Defense Methods Based on Clean Datasets.: An effective defense method should accurately detect adversarial examples while preserving classification performance on the original audio signal. In this section, we aim to understand the baseline performance of each defense method when detecting clean music datasets. Specifically, we collect 80 songs from 8 different genres to serve as the clean music training datasets, and we use additional 80 different clean music clips as the test datasets.

Evaluation metrics: We consider the following metrics to evaluate the performance of the classification: i) True Positive Rate (TPR): Measures the proportion of actual copyright music correctly identified by the classifier. ii) True Negative Rate (TNR): Indicates the proportion of non-copyright music that is correctly detected.

Results analysis: As shown in Fig. 14. The range of the TPR for various defense methods varies from 93.75% to 98.75%. Specifically, audio preprocessing approaches such as down-sampling and quantization exhibit lower TPRs, achieving only 93.75%. In contrast, watermarking-based methods demonstrate higher effectiveness; for instance, Timbre watermarking achieves a TPR of 97.5%, and AudioSeal reaches 98.75%. Notably, the ensemble learning method and L_p-norm training both achieve the highest TPR at 98.75%, slightly surpassing AudioSeal's 97.5%. Overall, these defense methods show a high TPR on clean music datasets.

The differences in TNR among various defense methods are generally smaller compared to those observed in TPR. For example, the most significant variance in TNR is between Quantization (96.25%) and ensemble learning (98.75%). Most defense methods achieve a TNR of more than 97.5%. In a nutshell, all defense methods, including audio watermarking, diffusion-model-based methods, and ensemble learning, can effectively detect copyright music and accurately distinguish non-copyright music.

3) Evaluation of Ensemble-Learning Fingerprinting Against Different Defense Methods. Experiment setups: To better understand the performance of different defense methods, we collect a broad range of adversarial music signals to test various defense methods. i) CEMF-PA attack: we follow the same steps in Section VIII-C-1 to generate the CEMF-PA-based adversarial examples. ii) L_p-norm-based attacks: we generate the L_p norm-based adversarial examples via the ICML20 and psychoacoustic attacks. And we set the L_2 norm of the perturbation signal is no greater than 0.05.

We collect 160 adversarial examples for each method to evaluate the performance of different defenses. Additionally, we evaluate the detection accuracy of different defense methods against existing attacks, including ICML20, psychoacoustic, and CEMF-PA atttack. We use the detection accuracy to measure the effectiveness of the defense performance. A higher accuracy indicates that a model is more robust against adversarial examples.

Defenses against CEMF-PA attacks: As shown in Fig. 15, our proposed ensemble learning method achieves the highest accuracy (92.9%), substantially outperforming the watermarking-based AudioSeal (49.4%) and Timbre watermarking (21.9%). This may be because CEMF-PA attacks can effectively manipulate the spectrogram features (e.g., timbre) that alter the embedded watermark in Timbre watermarking and AudioSeal. Additionally, ensemble learning surpasses the diffusion-model-based AudioPure (10.63%) and L_p-norm training (13.1%). This may be due to the fact that AudioPure focuses on L_p-norm-based perturbations, which are vulnerable to music-feature-based attacks. Interestingly, audio preprocessing methods appear ineffective against CEMF-PA attacks, for example, the minimal accuracy achieved with a low-pass filter is only 2%. This could be because most of the revised audio fingerprints are likely located in a frequency range below 2 kHz.

Defenses against L_p-norm-based attacks: The ensemble learning method exhibits slightly lower accuracy compared to AudioPure and L_p-norm training for the ICML20 and psychoacoustic attacks, such as ICML20: 83.7% (ensemble-based) vs. 89.2% (L_p-norm-based training) vs. 91.9% (AudioPure). The main reason could be that Lp-norm-based attacks can introduce adversarial examples with outlier music feature deviations that fall outside the training scope of the ensemble-based training. As illustrated in Fig. 4(c), the largest pitch deviation of the Lp-norm-based adversarial example can achieve over 1500 cents (15 semitones), which is slightly greater than the pitch manipulation range of our ensemble learning method, i.e., 10 semitones ranging from -5 to +5 semitones. One potential solution is increasing the manipulation range of the ensemble-based training music samples, but it could involve more computational cost while achieving only minimal improvement against the Lp-norm-based attacks.

Overall defense evaluation: The average detection accuracy of the ensemble-based method is 87.5%, which is notably higher than down-sampling (32.9%), quantization (17.1%), and low-pass filtering methods (30.7%). Although the ensemble-based

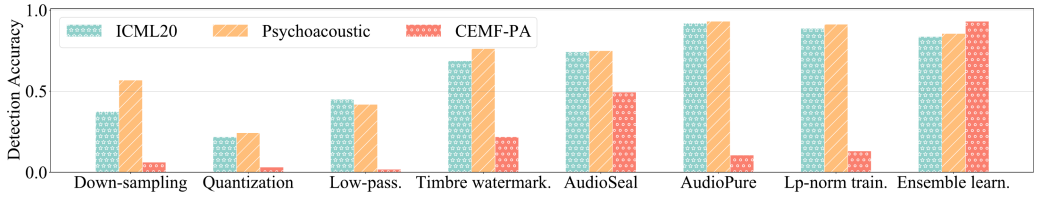


Fig. 15. The evaluation of different defense methods against various attacks.

method exhibits lower accuracy in L_p-norm-based attacks, on average, it demonstrates improvements of 23% over the L_p-norm-based training, 22.3% over AudioPure, 31.9% over Timbre watermarking, and 21.3% over AudioSeal. Overall, Fig. 15 shows that ensemble-learning-based adversarial fingerprinting is more robust than existing defense methods against various adversarial audio attacks.

IX. Discussions and Limitations

In this section, we discuss the scope of our study and potential improvements.

The scope of the SNN model: Our motivation for building an SNN model is to train the machine learning model that can capture the difference between the original and perturbed music signals in a manner similar to human perception. This model can thereby guide us to create the adversarial music that can minimize the human perception deviation. The scope of our current study does not explore the limitations of human hearing as most of the human perception studies [22], [23], [24] focused on the human-detectable perturbations. Thus, it may be challenging for our model to detect perturbations that fall outside the natural human hearing range, e.g., detecting the Dolphin attack [20]. It is still worth investigating training a machine learning model that can complement the normal hearing range.

SNN models are widely used in natural language processing (NLP) [97], [98] and computer vision (CV) tasks [99], [100], [101]. However, we have demonstrated that the SNN model can also be effectively adapted for use in the audio domain, specifically for quantifying music quality. This adaptation marks a significant progression from conventional uses in NLP and CV, highlighting the versatility of SNN models in handling diverse types of data, including text, images, and audio, thereby broadening their applicability to new fields.

Potential improvement of the SNN model: In our human study, the participants were college students aged 20-35 without professional music expertise. Consequently, our perceptual assessments capture the perspectives of general young people rather than music experts. To improve the accuracy and robustness of our SNN model, it would be beneficial to build a more generalized SNN model that includes diverse populations, such as music experts and participants across a broader age range.

Employing vocal tracks in the CEMF-PA attack: In this work, we mainly focus on using instrumental tracks to manipulate the timbre features. While there is another potential method that could enhance the perception of adversarial music signals by employing vocal tracks. There are two major issues of using the vocal tracks: (i) Manipulating a vocal track could involve a high computational cost, and directly shifting its pitch can compromise its naturalness, making it detectable to human ears. One possible solution could be creating a synthetic vocal track that mimics a natural voice, but this requires high computational resources, including training voice conversion models and producing varied vocal tracks. (ii) Vocal tracks may be less effective than instrumental tracks for audio fingerprinting attacks. This is because vocal tracks are typically shorter than instrumental tracks, and longer tracks offer more space to manipulate the audio fingerprinting to evade YouTube detection. Therefore, manipulating instrumental tracks is more straightforward than working with vocal tracks.

The computational cost: Compared to our conference version [1], we consider manipulating additional auditory features in the CEMF-PA attack, including pitch, rhythm, timbre, and loudness. This involves a higher computational cost, (e.g., requiring nearly 5000 iterations to create an adversarial example). The incurred cost still remains comparable to recent audio attacks. For instance, the recent speech attack [16] needs nearly 10000 iterations to produce a single example, while the existing speaker attack [11] requires 5000 iterations for one adversarial example. The potential solution to reduce the high computational cost is employing highly-efficient algorithms, but the primary issue comes from the extensive manipulation space. Additionally, efforts could be made to narrow down the specific range of manipulations for individual features, which is worth investigation in future work.

X. Conclusion

In this paper, we conducted a human study to reverse-engineer the human perception of music deviation and build a computationally efficient SNN model to predict the music quality. The proposed CEMF-PA attack method can alter multiple music features to deceive a music classifier, while maintaining the perceptual quality of the manipulated music. Experimental results have shown that the CEMF-PA attack is more effective than prior work. Based on the CEMF-PA attack, we proposed a new adversarial fingerprinting method and showed that the ensemble-based adversarial fingerprinting can make the music copyright detection model more robust to adversarial music.

References

[1] R. Duan, Z. Qu, S. Zhao, L. Ding, Y. Liu, and Z. Lu, "Perception-aware attack: Creating adversarial music via reverse-engineering human perception," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2022, pp. 905–919.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, arXiv: *1412.6572*.

[3] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *Artif. Intell. Safe. Secur.*, pp. 99–112, 2018.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2017, pp. 39–57.

[5] C. Szegedy et al., "Intriguing properties of neural networks," 2013, arXiv: *1312.6199*.

[6] N. Carlini et al., "Hidden voice commands," in *Proc. USENIX Conf. Secur. Symp.*, 2016, pp. 513–530.

[7] X. Yuan et al., "Commandersong: A systematic approach for practical adversarial voice recognition," in *Proc. 27th USENIX Conf. Secur. Symp.*, 2018, pp. 49–64.

[8] X. Chen et al., "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *Proc. USENIX Conf. Secur. Symp.*, 2020, pp. 2667–2684.

[9] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5231–5240.

[10] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[11] G. Chen et al., "Who is real bob? Adversarial attacks on speaker recognition systems," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2021, pp. 694–711.

[12] H. Abdullah et al., "Hear "no evil," see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2021, pp. 712–729.

[13] P. Saadatpanah, A. Shafahi, and T. Goldstein, "Adversarial attacks on copyright detection systems," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8307–8315.

[14] "Alexa Amazon," 2022, Accessed: Jan. 07, 2022. [Online]. Available: https://developer.amazon.com/en-US/alexa

[15] "Assistant Google," 2022, Accessed: Jan. 07, 2022. [Online]. Available: https://assistant.google.com/

[16] B. Zheng et al., "Black-box adversarial attacks on commercial speech platforms with minimal information," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2021, pp. 86–107.

[17] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2020, pp. 1121–1134.

[18] N. Carlini and D. Wagner, "Targeted attacks on speech-to-text," in *Proc. IEEE Secur. Privacy Workshops*, 2018, pp. 1–7.

[19] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 1962–1966.

[20] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 103–117.

[21] A. Patrick, K. R. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2019.

[22] E. Wenger et al., ""Hello, it's me": Deep learning-based speech synthesis attacks in the real world," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2021, pp. 235–251.

[23] W.-H. Tsai and H.-C. Lee, "Automatic evaluation of karaoke singing based on pitch, volume, and rhythm features," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 4, pp. 1233–1243, May 2012.

[24] C. Gupta, H. Li, and Y. Wang, "Perceptual evaluation of singing quality," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annua. Summit Conf.*, 2017, pp. 577–586.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[26] Y. Balaji, T. Goldstein, and J. Hoffman, "Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets," 2019, arXiv: *1910.08051*.

[27] Q.-Z. Cai, M. Du, C. Liu, and D. Song, "Curriculum adversarial training," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3740–3747.

[28] A. Shafahi et al., "Adversarial training for free!," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 3358–3369.

[29] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Representations*, 2018.

[30] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," 2020, arXiv: *2001.03994*.

[31] M. Balunovic and M. Vechev, "Adversarial training and provable defenses," in *Proc. Int. Conf. Learn. Representations*, 2019.

[32] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5286–5295.

[33] M. Mirman, T. Gehr, and M. Vechev, "Differentiable abstract interpretation for provably robust neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3578–3586.

[34] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput. Aided Verification*, 2017, pp. 97–117.

[35] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," 2015, arXiv: *1511.03034*.

[36] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: Increasing local stability of supervised models through robust optimization," *Neurocomputing*, vol. 307, pp. 195–204, 2018.

[37] H. Liu, Z. Yu, and N. Zhang, "When evil calls: Targeted adversarial voice over IP network," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2022, pp. 2009–2023.

[38] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," 2018, arXiv: *1809.10875*.

[39] S. Uhlich et al., "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 261–265.

[40] J. A. Moorer, "Signal processing aspects of computer music: A survey," *Proc. IEEE*, vol. 65, no. 8, pp. 1108–1137, Aug. 1977.

[41] M. Muller, D. P. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1088–1110, Oct. 2011.

[42] C. Kereliuk, B. Scherrer, V. Verfaille, P. Depalle, and M. M. Wanderley, "Indirect acquisition of fingerings of harmonic notes on the flute," in *Proc. Int. Comput. Music Conf.*, 2007, vol. 1, pp. 263–266.

[43] J.-C. Risset and D. L. Wessel, "Exploration of timbre by analysis and synthesis," in *The Psychology of Music*. Amsterdam, Netherlands: Elsevier, 1999, pp. 113–169.

[44] S. Godsill and M. Davy, "Bayesian harmonic models for musical pitch estimation and analysis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2002, pp. II-1769–II-1772.

[45] D. Manuel and S. J. Godsill, "Bayesian harmonic models for musical signal analysis," *Bayesian Stat.*, pp. 105–124, 2003.

[46] P. J. Walmsley, S. J. Godsill, and P. J. Rayner, "Multidimensional optimisation of harmonic signals," in *Proc. IEEE Eur. Signal Process. Conf.*, 1998, pp. 1–4.

[47] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 41, pp. 271–284, 2005.

[48] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in *Proc. 3rd IEEE Int. Conf. Multimedia Comput. Syst.*, 1996, pp. 473–480.

[49] I. J. Cox, M. L. Miller, J. A. Bloom, and C. Honsinger, *Digital Watermarking*. Berlin, Germany: Springer, 2002.

[50] E. Gomez and M. Bonnet, "Mixed watermarking-fingerprinting approach for integrity verification of audio recordings," in *Proc. Int. Telecommun. Symp.*, 2002.

[51] A. Wang et al., "An industrial strength audio search algorithm," in *Proc. 4th Int. Conf. Music Inf. Retrieval*, 2003, pp. 7–13.

[52] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proc. IEEE*, vol. 96, no. 4, pp. 668–696, Apr. 2008.

[53] B. Pardo, "Finding structure in audio for music information retrieval," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 126–132, May 2006.

[54] E. Allamanche, "Audioid: Towards content-based identification of audio material," in *Proc. 110th AES Conv.*, 2001.

[55] H. Neuschmied, H. Mayer, and E. Batlle, "Content-based identification of audio titles on the internet," in *Proc. 1st Int. Conf. WEB Delivering Music*, 2001, pp. 96–100.

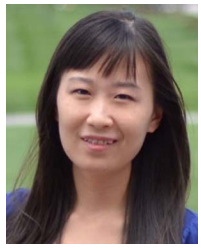
[56] J. Haitsma, T. Kalker, and J. Oostveen, "Robust audio hashing for content identification," in *Proc. Int. Workshop Content-Based Multimedia Index.*, 2001, vol. 4, pp. 117–124.
[57] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 5334–5341.
[58] E. Molina, I. Barbancho, E. Gómez, A. M. Barbancho, and L. J. Tardón, "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 744–748.
[59] W. M. Hartmann, *Signals, Sound, and Sensation*. Berlin, Germany: Springer, 2004.
[60] D. B. Loeffler, "Instrument timbres and pitch estimation in polyphonic music," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2006.
[61] F. De Leon and K. Martinez, "Enhancing timbre model using MFCC and its time derivatives for music similarity estimation," in *Proc. 20th Eur. Signal Process. Conf.*, 2012, pp. 2005–2009.
[62] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
[63] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proc. Int. Conf. Des. Mater.*, 2004, pp. 11–22.
[64] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 8, pp. 2121–2133, Nov. 2010.
[65] J. G. Rix, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2001, Art. no. 749.
[66] S. Valentini-Botinhao and J. Yamagishi, "Investigating RNN-based speech enhancement methods for noise-robust text-to-speech.," *In*SSW, 2016.
[67] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*. Hoboken, NJ, USA: Wiley, 2008.
[68] J. Li, "Real world audio adversary against wake-word detection systems," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11931–11941.
[69] T. Thiede et al., "PEAQ-the ITU standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, vol. 48, pp. 3–29, 2000.
[70] C. Gupta, H. Li, and Y. Wang, "A technical framework for automatic perceptual evaluation of singing quality," *APSIPA Trans. Signal Inf. Process.*, vol. 7, 2018, Art. no. e10.
[71] W. W. Daniel, "The spearman rank correlation coefficient," *Biostatistics:*, 1987.
[72] P. Sedgwick, "Spearman's rank correlation coefficient," *British Med. J.*, vol. 349, 2014, Art. no. g7327.
[73] J. E. Campbell et al., "Photosynthetic control of atmospheric carbonyl sulfide during the growing season," *Science*, vol. 322, pp. 1085–1088, 2008.
[74] R. Bailis, M. Ezzati, and D. M. Kammen, "Mortality and greenhouse gas impacts of biomass and petroleum energy futures in Africa," *Science*, vol. 308, pp. 98–103, 2005.
[75] J. M. Murphy et al., "Quantification of modelling uncertainties in a large ensemble of climate change simulations," *Nature*, vol. 430, pp. 768–772, 2004.
[76] Q. Wang, J. Yao, L. Zhang, P. Guo, and L. Xie, "Timbre-reserved adversarial attack in speaker identification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 3848–3858, 2023.
[77] Z. Yu, Y. Chang, N. Zhang, and C. Xiao, "SMACK: Semantically meaningful adversarial audio attack," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 3799–3816.
[78] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 539–546.
[79] G. Koch et al., "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2015.
[80] P. H. Sung and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1199–1208.
[81] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with Siamese recurrent networks," in *Proc. Natural Lang. Process. Workshop*, 2016, pp. 148–157.
[82] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2786–2792.
[83] T. Ranasinghe, C. Orsan, and R. Mitkov, "Semantic textual similarity with Siamese neural networks," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process.*, 2019, pp. 1004–1011.
[84] P. Manocha, R. Badlani, A. Kumar, A. Shah, B. Elizalde, and B. Raj, "Content-based representations of audio using Siamese neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 3136–3140.
[85] D. Droghini, F. Vesperini, E. Principi, S. Squartini, and F. Piazza, "Few-shot Siamese neural networks employing audio features for human-fall detection," in *Proc. Int. Conf. Pattern Recognit. Artif. Intell.*, 2018, pp. 63–69.
[86] R. Agrawal and S. Dixon, "Learning frame similarity using Siamese networks for audio-to-score alignment," in *Proc. 28th Eur. Signal Process. Conf.*, 2021, pp. 141–145.
[87] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," in *Proc. 112th AES*, Munich, Germany, 2002.
[88] Y. Sharma and P. Chen, "Attacking the madry defense model with l_1-based adversarial examples," 2017, *arXiv:1710.10733*.
[89] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, "Detecting voice cloning attacks via timbre watermarking," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2024.
[90] R. San Roman, P. Fernandez, A. Défossez, T. Furon, T. Tran, and H. Elsahar, "Proactive detection of voice cloning with localized watermarking," 2024, arXiv: *2401.17264*.
[91] J. Seok, J. Hong, and J. Kim, "A novel audio watermarking algorithm for copyright protection of digital audio," *etri J.*, vol. 24, no. 3, pp. 181–189, 2002.
[92] M. Hemis and B. Boudraa, "Digital watermarking in audio for copyright protection," in *Proc. 2014 Int. Conf. Adv. Comput. Sci. Inf. Syst.*, 2014, pp. 189–193.
[93] R. Shelke and M. U. Nemade, "Audio watermarking techniques for copyright protection: A review," in *Int. Conf. Glob. Trends Signal Process. Inf. Comput. Commun.*, 2016, pp. 634–640.
[94] S. Wu, J. Wang, W. Ping, W. Nie, and C. Xiao, "Defending against adversarial audio via diffusion model," 2023, arXiv: *2303.01507*.
[95] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," 2020, arXiv: *2009.09761*.
[96] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.
[97] P. Ni, Y. Li, G. Li, and V. Chang, "A hybrid Siamese neural network for natural language inference in cyber-physical systems," *ACM Trans. Internet Technol.*, vol. 21, no. 2, pp. 1–25, 2021.
[98] Y. B. de Souza and C. M. C. Carvalho, "Exploiting Siamese neural networks on short text similarity tasks for multiple domains and languages," in *Proc. Int. Conf. Comput. Process. Portuguese Lang.*, 2020, pp. 357–367.
[99] D. Chicco, "Siamese neural networks: An overview," *Artif. Neural Netw.*, pp. 73–94, 2021.
[100] K. L. Wiggers, A. S. Britto, L. Heutte, A. L. Koerich, and L. S. Oliveira, "Image retrieval and pattern spotting using Siamese neural network," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.
[101] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *Proc. Int. Conf. Pattern Recognit.*, 2016, pp. 378–383.

Rui Duan received the PhD degree from the University of South Florida. He is an assistant professor with the Division of Computing and Mathematics, School of Science and Engineering, University of Someplace. His research is centered on the intersection of AI Security and Machine Learning. He has published several notable conference papers in top-tier Security and ML conferences. His research is also related to Network Security and the Internet of Things.

Zhe Qu received the BS degree from the Department of Automation, Xiamen University, in 2015, the MS degree from the Department of Electrical and Computer Engineering, University of Delaware, in 2017, and the PhD degree from the Department of Electrical Engineering, University of South Florida. He is a professor with the School of Computer Science and Engineering, Central South University. His research interest includes focused on network security.

Shangqing Zhao received the PhD degree from the University of South Florida, in 2021. He is an assistant professor with the School of Computer Science, University of Oklahoma. His research primarily focuses on the network and mobile system design and security. His research results have been published in top-tier conferences and journals.

Leah Ding received the PhD degree from University, Buffalo, in 2013. She is an associate professor with American University. Before joining AU, she was a research principal with Accenture Labs (the R&D division of Accenture), and an adjunct professor with Johns Hopkins University. She is broadly interested in trustworthy machine learning with its applications in cybersecurity and scientific data analytics.



Yao Liu received the PhD degree from North Carolina State University, in 2012. She is an associate professor with the Department of Computer Science and Engineering, University of South Florida. Her research interests include in the security applications for cyberphysical systems, Internet of Things, and machine learning. She was an NSF CAREER Award recipient in 2016. She also received the ACM CCS Test-of-Time Award by ACM SIGSAC in 2019.

Zhuo Lu received the PhD degree from North Carolina State University, in 2013. He is an associate professor with the Department of Electrical Engineering, University of South Florida. His research has been mainly focused on both theoretical and system perspectives on communication, network, and security. He received the NSF CISE CRII award in 2016, the Best Paper Award from IEEE GlobalSIP in 2019, and the NSF CAREER award in 2021.