INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Fooling Speaker Identification Systems with Adversarial Background Music

Chu-Xiao Zuo, Jia-Yi Leng, Wu-Jun Li

National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
{zuochuxiao,lengjiayi}@smail.nju.edu.cn, liwujun@nju.edu.cn

Abstract

Speaker identification (SI) systems are widely used in real-world scenarios but are vulnerable to attacks from malicious users. Although existing attacks mainly focus on speech-shaped inputs, SI models can also be broken by speech-unrelated background music (BGM) in practical use. In this paper, we propose a new attack, called BGM Attack (BGMA), that generates auditorily natural music to deceive SI models. BGMA integrates a music generation model and a SI model to modify the music-level semantic features. We propose a linear transform called differentiable spectrogram reconstruction (DSR) that acts as a bridge for conveying gradient information between the two models in BGMA. Our experiments show that BGMA can effectively break state-of-the-art SI models with generated auditorily natural music. The result of this paper highlights the need for SI models to be robust against attacks from non-speech inputs and provides a novel attack method for testing the security of SI systems.

Index Terms: speaker identification, adversarial attack, background music attack, music generation

1. Introduction

Speaker identification (SI) systems are widely used in real-world scenarios, including security-related applications like biometric authentication and online payment. Nevertheless, ensuring the reliability of these systems is crucial for their practical implementation. It is imperative to guarantee that these systems are resilient to potential threats from malicious attackers.

Traditionally, spoofing attacks, such as replay attacks [1, 2], voice conversion attacks [3, 4], and speech synthesis attacks [5], have been employed to fool SI systems. These attacks involve reusing or imitating the original voiceprint of the target speaker in order to generate deceptive audio that can deceive both the model and human listeners.

Recent research has highlighted a significant threat to deep neural networks (DNNs), which is their vulnerability to adversarial examples [6, 7]. Adversarial examples are inputs subtly modified with imperceptible perturbations, causing misjudgments of a target model. Initially discovered in the image domain, adversarial attacks based on gradient optimization methods such as fast gradient sign method (FGSM) [6], Carlini and Wagner (C&W) attack [8], projected gradient descent (PGD) [9], and Auto Attack [10] have proven to be effective to attack DNNs. Recent studies have adapted these methods to speaker verification (SV) and SI models, by manipulating either the waveform data [11, 12, 13, 14] or acoustic features [15, 16] with certain constraints. Consequently, this migration has led to the successful development of acoustic-designed attacks [17, 18, 19], demonstrating that an attacker can generate audio that resembles someone else's voice but can still be recognized as the target speaker by the SI model.

Although existing attacks mainly focus on speech-shaped inputs, it is crucial to recognize that SI models can be broken by any sound in real-world scenarios, even those unrelated to human speech. While a few researchers have explored techniques for attacking voice control systems with speech-unrelated noise [20, 21], such attacks produce meaningless noises that might be detected as abnormal. As far as we know, no speech-unrelated attack has been designed for SI systems.

This paper tries to investigate the feasibility of utilizing speech-unrelated background music to attack SI systems. Instead of inefficiently searching a preexisting music database, modifying existing music and integrating a music generation system with an SI model is more controllable. However, a problem arises in establishing a differentiable integration of the SI and music generation models. Although Lu et al. [22] use non-negative least squares (NNLS) optimization for spectrogram reconstruction to generate music waveform, their method lacks differentiability. Alternatively, vocoder networks like WaveNet [23] and HiFi-GAN [24] can be employed but require additional training and a more complex computation flow.

In order to address the problem, we propose a novel attack method called background music attack (BGMA) that generates auditorily natural music to deceive SI systems. The main contributions of this paper are listed as follows:

- We explore a new threat to deceive SI systems and propose BGMA, the first method for attacking SI systems with speech-unrelated music.
- We propose a linear transform called differentiable spectrogram reconstruction (DSR) that acts as a bridge for conveying gradient information between the two models in BGMA. DSR is the first technique that enables backpropagation through a straightforward linear transform in the spectrogram reconstruction process.
- Our experiments show that BGMA can effectively break state-of-the-art (SOTA) SI models and generate auditorily natural music superior to those generated by migration of the PGD attack. The results highlight the need for SI models to be robust against attacks from non-speech inputs.

2. Background

2.1. Speaker Identification Models

Speaker identification is a common task in speaker recognition that aims to determine if an input utterance belongs to a group of enrolled speakers. An SI model first extracts the speaker embedding of an utterance, usually with a pooling strat-

10.21437/Interspeech.2023-100

egy [25, 26, 27, 28, 29, 30]. The extracted embedding is subsequently compared to the embeddings of the enrolled speakers to acquire similarity scores. There are two main identification tasks: closed-set identification (CSI) and open-set identification (OSI). In CSI, the system assumes that the input utterance belongs to one of the enrolled speakers. Hence, the SI system returns the prediction of the speaker with the highest similarity score without offering a rejection option. In contrast, OSI is considered to be more practical since it allows input from speakers who have not been previously enrolled. A preset similarity threshold enables the system to reject utterance from unrolled speakers.

We define the problem formally with the SI system represented as F. Enrolled speakers, numbered by $\{\mathbf{1}, 2, ..., K\}$, form a speaker group G. The SI model outputs similarity scores of an input $\boldsymbol{x}$ with respect to the enrolled speakers in G, denoted as $S(\boldsymbol{x}) = (s_1, ..., s_K)$ and $(S(x))_i = s_i$. In the OSI task, the SI system identifies the speaker of $\boldsymbol{x}$ with highest similarity score, while inputs with scores below the preset threshold θ is rejected. The identification process of OSI is denoted as:

$$F(\boldsymbol{x}) = \begin{cases} \arg\max\limits_{i \in G} (S(\boldsymbol{x}))_i, & \text{if } \max\limits_{i \in G}(S(\boldsymbol{x}))_i > \theta \\ \text{reject}, & \text{otherwise} \end{cases} \tag{1}$$

2.2. Threat Model

We study how an adversarial attacker takes speech-unrelated music to fool the SI system. The detailed threat model is illustrated below.

Adversary Goals. The attacker aims to deceive the SI system so that the system misidentifies a music slice as an utterance from a speaker. In a targeted attack, the crafted adversarial music $\boldsymbol{x}^{adv}$ should be identified as a specific enrolled speaker y, represented as $F(\boldsymbol{x}^{adv}) = y$. In contrast, the un-targeted attack aims to craft a music slice that will be identified as any of the enrolled speakers. Nevertheless, this paper mainly focuses on the targeted attack, a more challenging problem. Successful targeted attacks require increasing the similarity score $S((\boldsymbol{x}^{adv}))_y$ of the specified speaker. In contrast, an un-targeted attack only needs to increase the similarity score of any enrolled speaker. By understanding and addressing the targeted attack, we can better understand the security vulnerabilities of SI systems.

Suppose $\mathbb{M}$ is a set of music slices that are auditorily natural, the adversary goal is to solve the maximization problem:

$$\arg\max_{\boldsymbol{x}^{adv} \in \mathbb{M}} (S(\boldsymbol{x}^{adv}))_y, \quad s.t. \, (S(\boldsymbol{x}^{adv}))_y > \theta. \tag{2}$$

Adversary Knowledge. As the first exploration of the potential new attack, we perform white-box attacks by assuming full access to the information of the target SI model. With the known model structure and parameters, we can obtain the gradients of the model.

Adversarial Capabilities. Generating an adversarial example typically requires imposing a perturbation budget under the L_2 or L_∞ distance [6]. However, in the case of generating auditorily natural music, the attacker is not obligated to mimic an original input. Hence, the perturbation should not be limited by a budget. We employ a music generation model to modify an initial music slice on music-level semantic features. Consequently, the attacker can focus on creating adversarial music without limited by a perturbation budget.

3. Method

We propose the background music attack (BGMA), which integrates a music generation model with the target SI model. The attack conveys gradients from the SI model to the music generation model using our DSR transform. Adversarial music is generated through iterative gradient descent updates.

3.1. Integrate a Music Generation Model

A straightforward approach to generating adversarial music is to directly apply the PGD [9] attack to preexisting natural music. However, the success of the attack is contingent upon the perturbation size, which may produce irregular noise that lacks naturalness. The gradient information guided solely by the SI model disregards the music's inherent characteristics.

To address this limitation, we propose to generate music by modifying the music-level semantic features of a music generation model in BGMA. Specifically, our music generation model utilizes the autoencoder structure proposed in [22]. The model comprises an encoder network E that extracts a semantic content code $c_{\boldsymbol{x}}$ and a style code $r_{\boldsymbol{x}}$ from the input music $\boldsymbol{x}$, as well as a decoder network D that converts the semantic codes into a Mel-spectrogram $\bar{\boldsymbol{X}} \in \mathbb{R}^{n \times T}$ of the newly generated music. The autoencoder is trained on a music-style transfer task based on the Unsupervised Image-to-Image Translation (MUNIT) [31] framework. We only modify the content code in BGMA as we observed a limited impact when altering the style code in our experimental study.

3.2. Differentiable Spectrogram Reconstruction

Sending the generated music to the SI model necessitates recovering waveform data from the output Mel-spectrogram. A reconstruction of the magnitude spectrogram is required, which presents an obstacle. The existing work [22] utilizes NNLS to do the reconstruction, but the method lacks differentiability. One potential solution is to transfer the Mel-spectrogram between the two models directly. However, music generation models often require a higher-dimensional Mel-spectrogram (e.g., 256) to accurately capture music frequencies, while SI models utilize lower dimensions (e.g., 40 or 80). Alternatively, utilizing a vocoder network necessitates additional training and involves substantial costs. Hence, we propose DSR, a straightforward linear transform to reconstruct the magnitude spectrogram from the Mel-spectrogram. DSR addresses the issue of minimal norm least-square reconstruction by employing the Moore–Penrose inverse matrix [32] of the Mel-filter banks, along with a historical spectrogram incorporated at each optimization step.

The problem is formally defined as follows: given a magnitude spectrogram $\boldsymbol{X} \in \mathbb{R}^{m \times T}$, the Mel-spectrogram is obtained as $\bar{\boldsymbol{X}} = \boldsymbol{M}\boldsymbol{X}$, where $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ represents the Mel-filter bank. Due to the lossy nature of the matrix multiplication with $n < m$, recovering $\boldsymbol{X}$ from $\bar{\boldsymbol{X}}$ and $\boldsymbol{M}$ equals solving the linear equations that have infinitely many solutions. It is mathematically infeasible to get an accurate reconstruction. Instead, we propose the DSR transform, which can find an approximate solution in the attack process. During iteration step t, the DSR transform approximates $\boldsymbol{X}_t$ using historical information from the preceding step, according to the following theorem:

Theorem 1 *Suppose $\boldsymbol{M}^+ \in \mathbb{R}^{m \times n}$ is the Moore-Penrose inverse matrix of Mel-filter bank $\boldsymbol{M} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{M} = \boldsymbol{U}\Sigma\boldsymbol{V}^{tr}$ is the singular value decomposition (SVD) of*

$\boldsymbol{M}$. *Denote* $\boldsymbol{V}_{11} \in \mathbb{R}^{n\times n}, \boldsymbol{V}_{12} \in \mathbb{R}^{n\times(m-n)}, \boldsymbol{V}_{21} \in \mathbb{R}^{(m-n)\times n}, \boldsymbol{V}_{22} \in \mathbb{R}^{(m-n)\times(m-n)}$ *as four partitions s.t.*

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_{11} & \boldsymbol{V}_{12} \\ \boldsymbol{V}_{21} & \boldsymbol{V}_{22} \end{bmatrix}. \tag{3}$$

Then we have

$$\boldsymbol{X} = \boldsymbol{M}^{+}\bar{\boldsymbol{X}} + \boldsymbol{B}\boldsymbol{X}, \tag{4}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{V}_{12}(\boldsymbol{V}_{12})^{tr} & \boldsymbol{V}_{12}(\boldsymbol{V}_{22})^{tr} \\ \boldsymbol{V}_{22}(\boldsymbol{V}_{12})^{tr} & \boldsymbol{V}_{22}(\boldsymbol{V}_{22})^{tr} \end{bmatrix}, \tag{5}$$

and $(\boldsymbol{V}_{ij})^{tr}$ is the transpose of $\boldsymbol{V}_{ij}$. By updating $\boldsymbol{X}_t$ with $\boldsymbol{X}_{t-1}$ and $\bar{\boldsymbol{X}}_t$ in the iteration step t according to Equation (4), we propose DSR as:

$$\boldsymbol{X}_t = \boldsymbol{M}^{+}\bar{\boldsymbol{X}}_t + \boldsymbol{B}\boldsymbol{X}_{t-1}. \tag{6}$$

Utilizing a simple linear transform, DSR establishes a differentiable pathway for conveying gradients between the two models.

3.3. Pipeline of BGM Attack

During the initialization step, a music slice is passed through the short-time Fourier transform (STFT) and the encoder E to obtain the initial semantic codes $\boldsymbol{c}_0$ and $\boldsymbol{r}_0$, as well as the initial phase matrix $\boldsymbol{P}_0$ and magnitude spectrogram $\boldsymbol{X}_0$. Subsequently, the decoder D and DSR transform are employed to update and generate new music:

$$\begin{aligned} \bar{\boldsymbol{X}}_t &= D(\boldsymbol{c}_{t-1}, \boldsymbol{r}_{t-1}), \\ \boldsymbol{x}_t &= ISTFT(\boldsymbol{M}^{+}\bar{\boldsymbol{X}}_t + \boldsymbol{B}\boldsymbol{X}_{t-1}, \boldsymbol{P}_0), \end{aligned} \tag{7}$$

where $ISTFT$ is the inverse short-time Fourier transform. The generated music $\boldsymbol{x}_t$ is forwarded to the SI model to acquire the approximated gradients of the identification loss. Then, we modify the content code by momentum stochastic gradient descent:

$$\begin{aligned} \boldsymbol{g}_t &= \beta \cdot \boldsymbol{g}_{t-1} + \frac{\partial \mathcal{L}(S(\boldsymbol{x}_t), y)}{\partial \boldsymbol{c}_{t-1}}, \\ \boldsymbol{c}_t &= \boldsymbol{c}_{t-1} + \eta \cdot \boldsymbol{g}_t. \end{aligned} \tag{8}$$

The loss function $\mathcal{L}$ calculates the similarity score for the target speaker y:

$$\mathcal{L}(S(\boldsymbol{x}_t), y) = \max(-(S(\boldsymbol{x}_t))_y + c,\ 0), \tag{9}$$

where c is a confidence parameter with $c \geq \theta$. The design of the loss function is based on the targeted attack in previous work [9]. The whole pipeline of BGMA is summarized in Algorithm 1.

Algorithm 1 BGMA

Input:
SI system F, SI Model S, music encoder E and decoder D, a natural music $\boldsymbol{x}$, target speaker y;
number of iterations N, step size η, momentum factor β.
Output:
Adversarial music $\boldsymbol{x}^{adv}$;

$(\boldsymbol{c}_0, \boldsymbol{r}_0) \leftarrow E(\boldsymbol{x})$; $\boldsymbol{X}_0, \boldsymbol{P}_0 \leftarrow STFT(\boldsymbol{x})$; $\boldsymbol{g}_0 \leftarrow \boldsymbol{0}$;
for $t = 1$ to N **do**
  Generate music $\boldsymbol{x}_t$ by Eq. (7);
  Send $\boldsymbol{x}_t$ to SI system;
  if $F(\boldsymbol{x}_t) = y$ **then**
    $\boldsymbol{x}^{adv} \leftarrow \boldsymbol{x}_t$;
    return $\boldsymbol{x}^{adv}$;
  end if
  Get gradients by $\nabla\mathcal{L}(S(\boldsymbol{x}_t), y)$;
  Update $\boldsymbol{g}_t$ and $\boldsymbol{c}_t$ by Eq. (8);
  $\boldsymbol{r}_t \leftarrow \boldsymbol{r}_{t-1}$;
end for
$\boldsymbol{x}^{adv} \leftarrow \boldsymbol{x}_N$;
return $\boldsymbol{x}^{adv}$

4. Experiment

4.1. Setup

4.1.1. Datasets

We collect a dataset for the music generation model to train a style transfer task following [22]. The dataset includes 8000 seconds of both guitar and piano solos. We initialize the generation of adversarial music using the HD Classical subset of the MUSAN music dataset [33]. For the SI task, we use TIMIT [34] and randomly divide it into training, validation, and test sets in a ratio of 8:1:1. The sampling rate of all waveform data is uniformly set to 16 kHz, and down-sampling is required for music data. Music slices are randomly clipped to 7 seconds, matching the maximum length of utterances in the TIMIT test set.

4.1.2. Music Generation and Victim SI Models

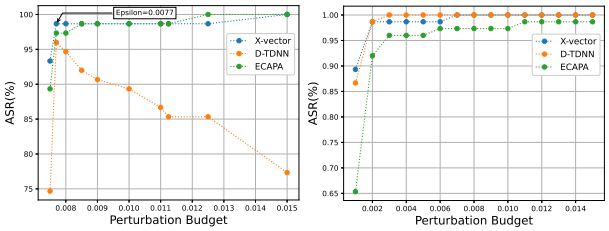
The music autoencoder follows the MUNIT implementation [22], and audio preprocessing involves applying STFT (size of 2048, hop-length of 160) and converting the spectrogram to a 256-size Mel-spectrogram. Three SOTA models (X-vector [25], D-TDNN [26], ECAPA-TDNN (ECAPA) [27]) trained on the TIMIT training set are selected as victim SI models. Preprocessing for SI models includes a 512-size STFT (hop-length of 160) and conversion of the spectrogram to a 40-size log Mel-spectrogram. Cosine similarity is employed to measure the similarity of speaker embedding as suggested in previous studies [25, 26, 27]. The optimal similarity threshold is determined by evaluating performance on a validation set using a random speaker verification task. The threshold is selected where the false acceptance rate (FAR) equals the false rejection rate (FRR) [35]. Threshold values (θ) are set to 0.2956, 0.2959, and 0.2718 for X-vector, D-TDNN, and ECAPA, respectively. The corresponding Top1 identification accuracy on the test set is 99.37%, 97.78%, and 99.05%.

4.1.3. Attacker Settings

In the PGD attack, we set $\epsilon = 0.008$ and $\eta = 0.002$. In BGMA, we set $\eta = 0.002$. The momentum parameters of the attack optimization are all set as $\beta = 1$ and the confidence parameter is set as $c = 1$.The max iteration step for both PGD and BGMA is fixed as 600.

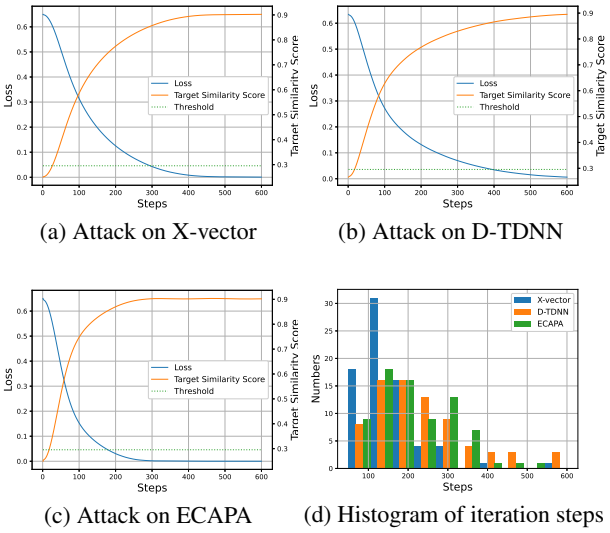
4.1.4. Evaluation Metrics

The strength and quality of adversarial music are assessed by the attack success rate (ASR) and Mean Opinion Score (MOS) [22]. ASR quantifies the frequency of successful attacks. The evaluation is conducted by targeting three random speakers for each music slice. For MOS evaluation, we collect the initial natural music, the music produced by BGMA and PGD, subsequently randomizing their order. We recruit participants to evaluate music offline, using the same listening equipment. They are asked to analyze whether the music sounds auditorily natural or artificially modified. We collect scores ranging



(a) Coarse search with varying η (b) Grid search with fixed η

Figure 1: *A parameter search of PGD migration.*



(a) Attack on X-vector (b) Attack on D-TDNN

(c) Attack on ECAPA (d) Histogram of iteration steps

Figure 2: *BGMA with the DSR differentiable transform approximating the spectrogram.*

from 1 to 5 based on 9000 responses to music slices.

4.2. Explore the PGD Attack

We first explore the direct migration of the PGD attack [9] bounded by L_∞ distance. Since larger perturbation results in more noticeable noise and lower MOS scores, we conduct a parameter search to find the minimal perturbation budget ϵ and proper step size η that achieve the best ASR. Initially, a coarse search is conducted with $\eta = \epsilon/4$. The results of ASR with varying ϵ are presented in Figure 1(a), showing that the optimal result is obtained at $\epsilon = 0.0077$ approximately. Subsequently, we set $\eta = 0.002$ and conduct a grid search. The results are presented in Figure 1(b). To compare PGD with BGMA, we choose $\epsilon = 0.008$ for a balance of the ASR and MOS. Although the PGD attack effectively deceives the SI model, the generated music exhibits noise that is perceptible to the human ear[1].

4.3. Effectiveness of BGMA

To show the effectiveness of DSR and BGMA, we first target all enrolled speakers with one initialization music slice as an example. Figure 2(a), Figure 2(b), and Figure 2(c) present the average loss and target similarity score $(S(\boldsymbol{x}^{adv}))_y$ of the attacks on the three SOTA SI models. BGMA achieves an ASR of 100% on all the targets, effectively reducing the loss and increasing the target similarity score throughout the iterations.

We perform attacks with all the initialization music to demonstrate that BGMA can launch attacks from different starting points. Figure 2(d) shows a histogram of the iteration steps when targeting one specific speaker by BGMA. The results suggest that most of the iteration steps are below the preset 600-step limit, except for a few challenging initial points. The results of ASR and MOS are summarized in Table 1. The PGD attack, which utilizes precise gradients, is more efficient. Nevertheless, the adversarial music generated by BGMA achieves comparable ASR to that of the PGD, while surpassing it in terms of auditory naturalness, as supported by a significantly higher MOS score of BGMA. In contrast to PGD, the music generated by BGMA does not exhibit a white-noise-like sound[1].

[1]The demos, codes, and datasets are available at `https://github.com/tartarleft/BGMA`.

Table 1: *Attack random speakers from all music. Nat denotes natural initialization music in the MUSAN music subsets.*

Victim	Attack	ASR(%)	MOS
X-vector	Nat	0	4.5
	PGD	**100**	2.02
	BGMA	98.22	**3.01**
D-TDNN	Nat	0	4.57
	PGD	**100**	1.72
	BGMA	96.44	**2.97**
ECAPA	Nat	0	4.4
	PGD	**99.86**	1.86
	BGMA	97.77	**3.12**

Table 2: *ASR (%) of BGMA without/with music augmentation (MA).*

Victim\Training	W/O MA	MA
X-vector	98.22	98.67
D-TDNN	96.44	91.56
ECAPA	97.77	100

4.4. Attack SI Models Trained with Music Augmentation

The MUSAN augmentation dataset comprises five subsets of music. Training SI models with these subsets can be considered as a defense against adversarial music. Hence, we attack models trained with MUSAN music subsets as an attack under defense. The results shown in Table 2 indicate that BGMA is still effective except for a slightly decreased ASR on D-TDNN.

5. Conclusion

In this paper, we explore a new threat and propose a novel attack called BGMA that can generate speech-unrelated adversarial music to fool SI systems. We also propose a DSR transform that conveys gradient information in BGMA. The experiments show that BGMA can effectively break state-of-the-art (SOTA) SI models and generate auditorily natural music superior to those generated by migration of the PGD attack.

6. Acknowledgment

This work is supported by NSFC Project (61921006, 62192783), and Fundamental Research Funds for the Central Universities (020214380108).

7. References

[1] M. Witkowski, S. Kacprzak, P. Zelasko, K. Kowalczyk, and J. Galka, "Audio replay attack detection using high-frequency features." in *INTERSPEECH*, 2017, pp. 27–31.

[2] H. Wu, H. Kuo, N. Zheng, K. Hung, H. Lee, Y. Tsao, H. Wang, and H. Meng, "Partially fake audio detection by self-attention-based fake span discovery," in *ICASSP*, 2022, pp. 9236–9240.

[3] Z. Wu, T. Virtanen, T. Kinnunen, E. Chng, and H. Li, "Exemplar-based unit selection for voice conversion utilizing temporal information," in *INTERSPEECH*, 2013, pp. 3057–3061.

[4] C. Wen, T. Guo, X. Tan, R. Yan, S. Zhou, C. Xie, W. Zou, and X. Li, "Time domain adversarial voice conversion for ADD 2022," in *ICASSP*, 2022, pp. 9221–9225.

[5] P. L. D. Leon, B. Stewart, and J. Yamagishi, "Synthetic speech discrimination using pitch pattern statistics derived from image analysis," in *INTERSPEECH*, 2012, pp. 370–373.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[8] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*, 2017, pp. 39–57.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[10] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *ICML*, vol. 119, 2020, pp. 2206–2216.

[11] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," *CoRR*, vol. abs/1711.03280, 2017.

[12] Y. Xie, Z. Li, C. Shi, J. Liu, Y. Chen, and B. Yuan, "Real-time, robust and adaptive universal adversarial attacks against speaker recognition systems," *J. Signal Process. Syst.*, vol. 93, pp. 1187–1200, 2021.

[13] W. Zhang, S. Zhao, L. Liu, J. Li, X. Cheng, T. F. Zheng, and X. Hu, "Attack on practical speaker verification system using universal adversarial perturbations," in *ICASSP*, 2021, pp. 2575–2579.

[14] J. Villalba, Y. Zhang, and N. Dehak, "x-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification," in *INTERSPEECH*, 2020, pp. 4233–4237.

[15] F. Kreuk, Y. Adi, M. Cissé, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *ICASSP*, 2018, pp. 1962–1966.

[16] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM i-vector based speaker verification systems," in *ICASSP*, 2020, pp. 6579–6583.

[17] J. Li, X. Zhang, J. Xu, L. Zhang, Y. Wang, S. Ma, and W. Gao, "Learning to fool the speaker recognition," in *ICASSP*, 2020, pp. 2937–2941.

[18] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," in *INTERSPEECH*, 2020, pp. 4228–4232.

[19] C.-X. Zuo, J.-Y. Leng, and W.-J. Li, "Speaker-specific utterance ensemble based transfer attack on speaker identification," in *INTERSPEECH*, 2022, pp. 3203–3207.

[20] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: Exploiting the gap between human and machine speech recognition," in *WOOT*, 2015.

[21] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. A. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Security*, 2016, pp. 513–530.

[22] C. Lu, M. Xue, C. Chang, C. Lee, and L. Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," in *AAAI*, 2019, pp. 1061–1068.

[23] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016, p. 125.

[24] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*, 2020.

[25] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[26] Y.-Q. Yu and W.-J. Li, "Densely connected time delay neural network for speaker verification," in *INTERSPEECH*, 2020, pp. 921–925.

[27] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *INTERSPEECH*, 2020, pp. 3830–3834.

[28] Y.-Q. Yu, S. Zheng, H. Suo, Y. Lei, and W.-J. Li, "Cam: Context-aware masking for robust speaker verification," in *ICASSP*, 2021, pp. 6703–6707.

[29] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP*, 2019, pp. 6046–6050.

[30] L. Fan, Q.-Y. Jiang, Y.-Q. Yu, and W.-J. Li, "Deep hashing for speaker identification and retrieval," in *INTERSPEECH*, 2019, pp. 2908–2912.

[31] X. Huang, M. Liu, S. J. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, vol. 11207, 2018, pp. 179–196.

[32] J. C. A. Barata and M. S. Hussein, "The moore–penrose pseudoinverse: A tutorial review of the theory," *Brazilian Journal of Physics*, vol. 42, no. 1, pp. 146–165, 2012.

[33] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *CoRR*, vol. abs/1510.08484, 2015.

[34] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[35] M. M. Kabir, M. F. Mridha, J. Shin, I. Jahan, and A. Q. Ohi, "A survey of speaker recognition: Fundamental theories, recognition methods and opportunities," *IEEE Access*, vol. 9, pp. 79 236–79 263, 2021.