PhoneyTalker: An Out-of-the-Box Toolkit for Adversarial Example Attack on Speaker Recognition

Meng Chen*, Li Lu*†, Zhongjie Ba*, Kui Ren*
*School of Cyber Science and Technology and Key Laboratory of Blockchain and Cyberspace Governance of
Zhejiang Province, Zhejiang University, Hangzhou, Zhejiang, China
{meng.chen, li.lu, zhongjieba, kuiren}@zju.edu.cn
†Corresponding Author

***Abstract*—Voice has become a fundamental method for human-computer interactions and person identification these days. Benefit from the rapid development of deep learning, speaker recognition exploiting voice biometrics has achieved great success in various applications. However, the shadow of adversarial example attacks on deep neural network-based speaker recognition recently raised extensive public concerns and enormous research interests. Although existing studies propose to generate adversarial examples by iterative optimization to deceive speaker recognition, these methods require multiple iterations to construct specific perturbations for a single voice, which is input-specific, time-consuming, and non-transferable, hindering the deployment and application for non-professional adversaries. In this paper, we propose *PhoneyTalker*, an out-of-the-box toolkit for any adversary to generate universal and transferable adversarial examples with low complexity, releasing the requirement for professional background and specialized equipment. *PhoneyTalker* decomposes an arbitrary voice into phone combinations and generates phone-level perturbations using a generative model, which are reusable for voices from different persons with various texts. Experiments on mainstream speaker recognition systems with large-scale corpus show that *PhoneyTalker* outperforms state-of-the-art methods with overall attack success rates of 99.9% and 84.0% under white-box and black-box settings respectively.**

***Index Terms*—Adversarial example attack, universal adversarial perturbation, generative model, speaker recognition**

I. Introduction

As the most natural human communication method using one of the most sensitive channels (i.e., the hearing), the voice plays an important role in not only human interactions, but also person identification, since antiquity. And in the era of human-computer interactions, the voice becomes a prevalent biometric for automatic speaker recognition gradually, owing to the strengths of non-contact, user-friendly and continuous authentication [1]–[3]. Meanwhile, speaker recognition benefits from the rapid progress of deep learning techniques, bringing out various mature products (e.g., voice assistant, voiceprint lock, etc.) to intelligentize people's lives and work. A report [4] showing the global voice biometric market size reached $1.1 billion in 2020 and is expected to approach $3.9 billion by 2026, also supports its rising trend in deployment and application. However, behind the bright future of speaker recognition, the shadow of deep learning's vulnerability to adversarial example attacks is becoming a severe threat gradually. Recent studies [5]–[7] proved that deep neural network-based speaker recognition could be spoofed by imposing subtle perturbations on benign voices, i.e., suffering from adversarial example-based impersonation attacks, which raises extensive public concerns and enormous research interests.

Early studies [5], [8], [9] investigate the vulnerability of speaker recognition under classical white-box adversarial example attacks (e.g., FGSM [10], PGD [11], C&W [12]). To overcome the impracticality of the white-box setting, the following works [13], [14] exploit the generalization of adversarial examples to transfer the attack from a local substitute model to the target model. But such methods suffer from performance degradation due to the weak generalization of adversarial examples. To improve the performance of black-box attacks, FakeBob [15] proposes to employ the query scores as the basis for gradient estimation, and combine it with the natural evolution strategy for perturbation calibration. Recent studies [16]–[19] even investigate to construct universal adversarial perturbations that can be applied to any voice, releasing the efforts of repetitive adversarial example generation. Although the aforementioned studies have demonstrated the feasibility of spoofing speaker recognition by adversarial example attacks, they are either input-specific, non-transferable, or time-consuming. Such problems are only partially alleviated in some works without a comprehensive solution. Moreover, previous methods put forward high requirements on the attacker's capability, thereby causing a limited impact in practice.

Toward this end, our work aims to propose an out-of-the-box toolkit for adversarial example attacks on speaker recognition, which enables any non-professional adversary to generate voice adversarial examples for impersonating a target user without professional background or specialized equipment, realizing a *DeepFake* [20] in the speaker recognition field. The basic idea is to construct universal adversarial perturbations at the phone level using a generative model, which can map any voice to the adversary-desired target user. Exploiting the generalization of the generative model by training on a large-scale corpus with rich diversity, the attack could be transferred from local substitute models to unseen target models. To realize such an attack toolkit, we face several challenges. *Voice variation*: our attack is designed to fit any adversary under any speech text, so it should be robust to the voice variation induced by person differences and text diversity. *Generation complexity*: compared with AI professionals, non-professional adversaries have no specialized equipment for adversarial example generation, introducing a critical demand for a low-complexity algorithm. *Black-box setting*: the adversary has no prior knowledge of model details inside the target system, indicating the black-box-oriented attack design.

In this paper, we first introduce the speaker recognition system and illustrate the threat model of a targeted adversarial example attack on such systems. To realize an out-of-the-box attack, we further define three design goals except for the two basic ones of adversarial example attacks. Based on these goals, we propose *PhoneyTalker*, an out-of-the-box toolkit for any non-professional adversary to perform universal and transferable targeted adversarial example attacks with low complexity. *PhoneyTalker* first decouples the whole attack into the offline training and online attacking phases, releasing the requirement for adversarial perturbation reconstruction for different voices. To realize a universal attack, *PhoneyTalker* decomposes voices into phone combinations with a forced alignment method, then employs a generative model to learn input-independent perturbations at the phone level. Considering the signal distortion during perturbation injection, *PhoneyTalker* adopts a set of digital signal processing techniques to suppress the perturbation audibility. To improve the generalization ability of adversarial examples, *PhoneyTalker* trains the perturbations on a large-scale corpus to facilitate the input diversity, and also pretrains several mainstream speaker recognition models as the substitute classifiers for the target model. Moreover, *PhoneyTalker* introduces a loss function with a confidence margin to further enhance the transferability. In this way, the adversary can exploit the well-trained perturbations to generate adversarial examples from any person with any speech text, for impersonating a target user and spoofing unseen systems. Experimental results show that *PhoneyTalker* could successfully attack mainstream speaker recognition systems under white-box and black-box settings on different persons and texts with a low time cost, outperforming the state-of-the-art attack methods.

Our contributions are highlighted as follows:

- We propose an out-of-the-box toolkit, *PhoneyTalker*, enabling any adversary to perform universal targeted adversarial example attacks on speaker recognition systems without the requirement for professional background or specialized equipment.
- We design a phone-level perturbation generation method to construct universal adversarial perturbations, which could be imposed on any voice from different adversaries with various speech texts to impersonate a target speaker.
- We develop a generative model-based optimization approach to train the phone-level perturbations, which significantly accelerates the generation of adversarial examples, thus realizing a time-efficient attack.
- We conduct extensive experiments on state-of-the-art speaker recognition models with a large-scale corpus to evaluate the performance, and the results show *PhoneyTalker* achieves overall attack success rates of 99.9% and 84.0% under white-box and black-box settings respectively.

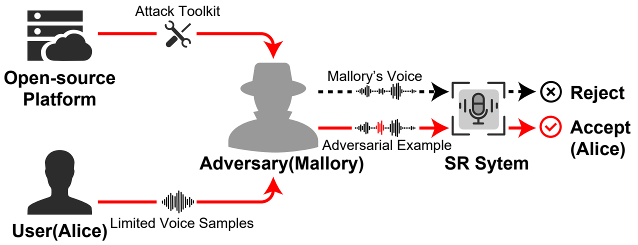


Fig. 1. Threat model of adversarial example attack.

The rest of this paper is organized as follows. We first introduce the speaker recognition system and threat model in Section II. Section III presents the design goals and toolkit architecture of *PhoneyTalker*, whose design details are further shown in Section IV. Section V shows the performance evaluation for *PhoneyTalker*. Finally, we review related works and make a conclusion in Section VI and Section VII respectively.

II. Attack Statement

In this section, we introduce the speaker recognition system and the threat model, then formulate the problem of adversarial example attacks against speaker recognition.

A. Speaker Recognition System

Speaker Recognition (SR) is also known as voiceprint authentication, which extracts acoustic embeddings from voices to learn individual characteristics of the speaker for user identification or verification. Specifically, the recorded voice is first preprocessed to reduce the interference of noise and multi-path effects. To further characterize the voiceprint based on the principles of human hearing, acoustic features (e.g., MFCCs, filter banks) are extracted from the preprocessed voice. Then the acoustic features are fed to a neural network model $f(\cdot)$ to learn the voice embeddings (e.g., d-vector [21], x-vector [22]). Finally, a scorer $S(\cdot)$ is used to measure the similarity between the embedding of the input voice x and stored user profiles. The user y with the highest score would be regarded as the user identity. Since the identification system is accessible to everyone, i.e., open-set, a threshold θ is preset to reject unenrolled speakers. Such an SR system can be formulated as follows:

$$h(x) = \begin{cases} \arg\max\limits_{y \in E} S(f(x)), & \max S(f(x)) > \theta \\ \text{unenrolled speaker}, & \text{otherwise}, \end{cases} \tag{1}$$

where $h(x)$ models the whole SR system, and E denotes a user group enrolled in the SR system.

B. Threat Model

Fig. 1 shows the threat model of an adversarial example attack on SR. An adversary (Mallory) desires to spoof the SR system for impersonating a target user (Alice), i.e., launch a targeted adversarial example attack. We assume that Mallory can directly access an attack toolkit from an open-source platform, requiring no professional background or specialized equipment, which realizes a similar out-of-the-box impersonation attack on SR with DeepFake [20] in the computer vision area. To acquire Alice's voiceprint to construct adversarial examples, we assume that Mallory can indirectly collect limited voice samples (e.g., 10∼20s, which is sufficient for SR systems to extract voiceprint [21], [22], [32]) from Alice in various ways, such as retrieving voices from public social media or making harassing phone calls. For instance, Alice may share some personal videos or audio records on public social platforms (e.g., Twitter, TikTok, YouTube), which can be retrieved by anyone without significant effort. Note that these limited voices from Alice have only a small amount of text, which is insufficient to attack the SR system directly (i.e., replay attack). Instead, Mallory turns to generating adversarial examples with the limited voice samples via the attack toolkit, which could involve arbitrary speech text. During the generation process of adversarial examples, we assume that Mallory has no prior knowledge of model details inside the SR system (e.g., network structure, model parameters, scorer threshold). With the generated adversarial examples, Mallory can impersonate Alice to bypass the SR system for malicious purposes.

In this attack, Mallory constructs and imposes a perturbation δ to any of her voice x_m to generate an adversarial example, with which Mallory could impersonate Alice y_a to spoof the SR system $h(\cdot)$, i.e., $h(x_m + \delta) \rightarrow y_a$. Therefore, the perturbations could be generated by optimizing the problem:

$$\begin{aligned} \arg\min_{\delta} \quad & L(h(x_m + \delta), y_a) \\ \text{s.t.} \quad & \|\delta\|_p \leq \epsilon, \end{aligned} \tag{2}$$

where $L(\cdot)$ is the loss function of the target SR system, $\|\cdot\|_p$ denotes L_p normalization, and ϵ is a constraint hyperparameter.

III. Attack Overview

In this section, we propose several design goals based on the threat model and present the architecture of *PhoneyTalker*.

A. Design Goals

According to Eq. (2), there are two basic goals of a general adversarial example attack:

- **Effectiveness**. As an impersonation attack, effectiveness means a high success rate to ensure that the adversary can successfully impersonate the target user and spoof the system, which is the primary goal of an adversarial example attack.
- **Imperceptibility**. Imposing perturbations on benign voices would lead to signal distortion, which may be perceived by humans. Hence, the secondary goal is to ensure the imperceptibility of generated adversarial examples.

Apart from these, in our threat model, the adversary is assumed to have no professional background or specialized equipment for adversarial example attacks. Hence, to enable any non-professional adversary to realize such an attack, we propose another three design goals:

- **Universality**. The attack toolkit in our threat model is assumed to be designed for anyone, i.e., the perturbations could be imposed on voices from any adversary, and have no limit on the speech text of adversarial examples. Thus, a universal attack is required for person independence and text independence.

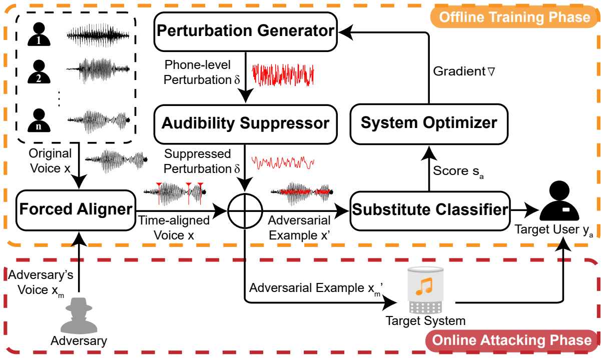


Fig. 2. Overall architecture of *PhoneyTalker*.

- **Low Complexity**. Due to the lack of specialized equipment for the adversary, a low-complexity attack is desired to reduce the demand for computing resources, which further indicates the necessity of saving training efforts and time cost during adversarial example generation.
- **Transferability**. As mentioned in Section II-B, the adversary has no prior knowledge of model details inside the SR system, i.e., a black-box attack. Hence, the transferability of adversarial examples on different SR systems is highly demanded.

B. Toolkit Architecture

To realize an adversarial example attack meeting the aforementioned goals, we propose *PhoneyTalker*, an out-of-the-box toolkit that enables any non-professional adversary to generate adversarial examples for impersonating any target user to spoof SR systems.

The basic idea of *PhoneyTalker* is to construct perturbations at the phone level. Since a phone is the fundamental phonetic unit in a language, any voice can be decomposed into a sequence of phones. To this end, *PhoneyTalker* decomposes the input voice into a phone sequence and constructs fine-grained phone-level adversarial perturbations. With arbitrary combinations of phone-level perturbations, we could realize a text-independent attack. In addition, *PhoneyTalker* employs a generative model to train universal adversarial perturbations on a large-scale corpus including hundreds of speakers, which further improves the universality of perturbations on different persons. Also, with the generative model, the offline training and online attacking phases are decoupled, in which the well-trained perturbations are reusable without reconstruction for each voice, enabling a low-complexity attack. To further facilitate the transferability of perturbations, *PhoneyTalker* provides multiple pretrained mainstream SR models as substitutes for target SR systems.

Fig. 2 shows the overall architecture of *PhoneyTalker*, including the offline training phase and online attacking phase. In the offline training phase, the adversary generates phone-level adversarial perturbations with collected voice samples by five components as follows:

- **Forced Aligner.** To construct phone-level perturbations, the forced aligner recognizes and localizes phones in the voice using forced alignment techniques, with which the voice is decomposed into a time-aligned phone sequence.
- **Perturbation Generator.** Fed with the time-aligned phones, a multi-layer generative neural network model is designed to generate the perturbation for each phone automatically.
- **Audibility Suppressor.** To avoid the human perceptibility of adversarial examples after imposing perturbations, the audibility suppressor integrates several signal processing techniques to constrain the perturbations.
- **Substitute Classifier.** For the success of transferring to black-box models, several mainstream SR models are pretrained as substitutes for target systems to enhance the generalization ability of our attack.
- **System Optimizer.** To train the perturbation generator, the system optimizer aims to minimize a designed loss function with a confidence margin by iterative gradient descent method.

In the online attacking phase, the adversary provides his/her voice to construct adversarial examples. The adversary's voices are first input to the forced aligner to derive the time-aligned phones, with which the perturbations generated in the offline training phase are imposed on the adversary's voices to construct adversarial examples. The adversary could then inject the adversarial examples into the target SR system to impersonate the target user and bypass the identity authentication, during which no professional knowledge or specialized equipment is required for the adversary.

IV. Design Details

In this section, we illustrate the design details of each component in *PhoneyTalker*.

A. Forced Aligner

To construct perturbations at the phone level, *PhoneyTalker* should first recognize and localize all phones in an input voice. As mentioned in Section II-B, the adversary can exploit his/her own voices to generate adversarial examples with any speech text, i.e., the speech text of the adversarial example is specified and known before perturbation generation. Meanwhile, the phones of each word in the speech text can be determined with a phone dictionary. Hence, the problem is intrinsically to derive the time duration of each phone in the voice, which could be addressed by Forced Alignment (FA).

FA is a technique for aligning an audio clip and its orthographic text transcription in the time domain, which determines the time duration of each phone in the audio clip. There are several prevalent tools for accurate FA, such as Prosodylab-aligner [23], Penn Phonetics Forced Aligner [24], FAVE-align [25], Montreal Forced Aligner (MFA) [26]. However, most of them rely on the HMM toolkit HTK [27], which is commercially licensed and requires complex compilation on specific platforms, thus hindering the accessibility and usability of non-professional adversaries. Fortunately, MFA is based on the open-source Kaldi toolkit [28] and can be automatically compiled with cross-platform compatibility. Hence, we adopt MFA as the forced aligner to enable any adversary to use it. Specifically, MFA looks up a preset phone dictionary to derive all the phones in the speech text, then extracts MFCC features from preprocessed voices, which are further fed to pretrained acoustic models to align the derived phones. Finally, MFA outputs the alignment information, i.e., the time duration of each phone in the voice. Such a preset phone dictionary and pretrained acoustic models in MFA could significantly save the efforts of non-professional adversaries.

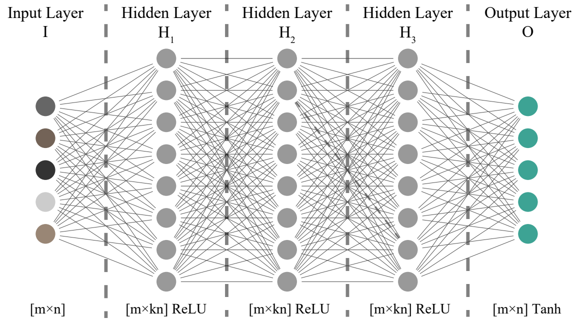


Fig. 3. Network structure of perturbation generator.

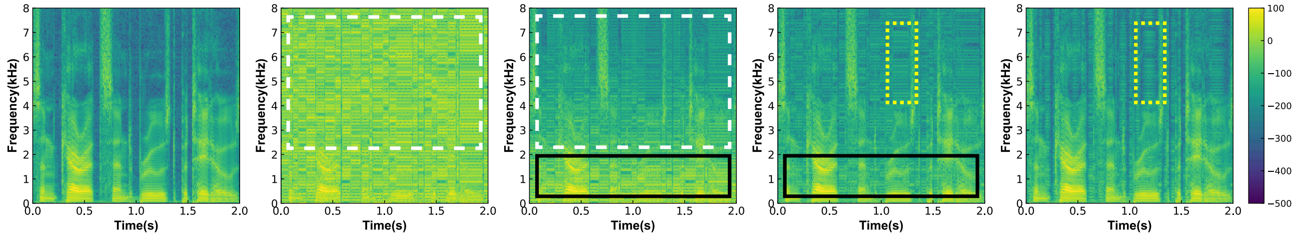
B. Perturbation Generator

With the time-aligned phones from the forced aligner, *PhoneyTalker* constructs universal adversarial perturbations for each phone with the perturbation generator.

To construct phone-level perturbations, we first need to summarize the phones in English. Generally, voice can be represented phonetically by a finite set of phones, which are denoted by a set of ASCII labels called the ARPAbet. Table I shows the list of ARPAbet phonetic labels used in *PhoneyTalker*, conforming to the commonly used CMU Sphinx dictionary [29]. According to the manner of articulation, the phones are classified into stops, fricatives, affricates, nasals, glides, liquids, and vowels, including 40 phonetic labels in total. Considering the silent fragments in a voice being detected and eliminated during the voice activity detection of SR systems, it's unnecessary to generate perturbations for these voice fragments. Therefore, the label /SIL/ (i.e., silence) is removed, leaving 39 phonetic labels in the list.

TABLE I
List of ARPAbet phonetic labels used in *PhoneyTalker*.

Class	ARPAbet Phonetic Labels
Stops	/P/, /B/, /T/, /D/, /K/, /G/
Fricatives	/HH/, /F/, /V/, /TH/, /DH/, /S/, /Z/, /SH/, /ZH/
Affricates	/CH/, /JH/
Nasals	/M/, /N/, /NG/
Glides	/Y/, /R/
Liquids	/W/, /L/
Vowels	/IY/, /IH/, /EY/, /EH/, /AE/, /AA/, /AO/, /UH/, /OW/, /UW/, /AH/, /ER/, /AY/, /AW/, /OY/
Silence	/SIL/



(a) Original voice (b) Adversarial example (c) Low-pass filtering (d) Amplitude clipping (e) Window smoothing

Fig. 4. Example of audibility suppression.

Based on the phonetic labels, we design a generative neural network as the perturbation generator. Since *PhoneyTalker* is designed to be a universal attack, the process of perturbation generation is input-independent. Hence, we feed phone-level random noise instead of specific voices to the generator as input and derive the output as perturbations. Considering the time duration of phones usually ranges in 50∼300ms (i.e., 800∼3200 points at the common sampling rate of $16kHz$ [30], [31]), a multi-layer DNN is adopted as the generator to learn the perturbation distribution for such low-dimensional samples. As shown in Fig. 3, the $m \times n$ input noise is mapped to the perturbation space through three $m \times kn$ hidden layers with ReLU activation, then we normalize $m \times n$ output layer in the range of [-1, 1] with Tanh activation to derive perturbations. The parameters m, n, k are the number of phones, the length of perturbations, and the scale factor of hidden layers, respectively.

Including phonetic labels as keys and perturbations as values, a perturbation dictionary is derived, with which *PhoneyTalker* could inject perturbations for any input voice.

C. Audibility Suppressor

The generated adversarial perturbations have significantly different acoustic features with normal voices, especially filled with high-frequency components and high amplitude peaks, which could be perceived by the human auditory system. Fig. 4(a) and 4(b) show the spectrums of a normal voice and an adversarial example generated from it, respectively. Perturbations with high frequency and amplitude can be observed in Fig. 4(b), which severely obscures the acoustic structures of the normal voice. To realize an imperceptible attack, the audibility of perturbations needs to be suppressed. Amplitude normalization and frequency masking are commonly used for perturbation suppression, but the former results in performance degradation while the latter is input-dependent. Therefore, we combine a set of signal processing techniques to suppress the perturbation audibility in terms of frequency, amplitude, and boundary change, respectively.

Low-pass filtering. To eliminate the high-frequency components of adversarial perturbations, a low-pass Digital Biquad Filter (DBF) is applied first. In DBF, the cut-off frequency is set as $2kHz$ and the energy loss factor defaults to 0.707. Fig. 4(c) shows the voice spectrum after low-pass filtering. Compared with Fig. 4(b), the frequency bands above 2kHz exhibit significantly lower power, which indicates the high-frequency components are suppressed.

Amplitude clipping. L_∞ normalization is then adopted to constrain the amplitude of adversarial perturbations. Specifically, a clipping operation is performed with the amplitude upper bound ϵ: $\delta = \text{clip}\{\delta, -\epsilon, \epsilon\}$. Fig. 4(d) shows the voice spectrum further processed by amplitude clipping. Compared with Fig. 4(c), the amplitude is squeezed in all frequency bands, making the obscured acoustic structures of the normal voice revealed especially in the low frequency bands.

Window smoothing. A Hann window is further applied to smooth the boundary change due to perturbation injection intervals. The window length depends on the time duration of the specific phone. Fig. 4(e) shows the voice spectrum after window smoothing. Compared with Fig. 4(d), the amplitude of perturbations at the phone boundary presents a slighter change over time. Finally, the spectrum in Fig. 4(e) results rather less distortion, thus the perturbations get more imperceptible.

D. Substitute Classifier

The suppressed perturbations are further imposed on the original voice in the training dataset to generate adversarial examples that are then fed to substitute classifiers.

Due to the phone diversity of various texts and the articulation variations of different persons, the length of phones is not fixed, and even the same phone has variable lengths for different words in one voice. Fig. 5 shows a voice with speech text *for some time* and its time-aligned phones (/SIL/, /F/, /ER/, /S/, /AH/, /M/, /T/, /AY/, /M/). We can observe that different phones have different lengths, and the lengths of the same phone /M/ in the words *some* and *time* are also different (i.e., 109ms and 170ms). To fully exploit the injectable space, *PhoneyTalker* repeats and concatenates the same fixed-length perturbation for each phone. Specifically, for a phone p at the time duration of $[s, e)$, $\lfloor \frac{e-s}{n} \rfloor$ perturbations are concatenated, where n is the perturbation length. Then the concatenated perturbation is imposed on the original voice with a random time shift τ to tolerate inaccurate alignment.

After perturbation injection, the adversarial examples are input to the substitute classifier. Under the black-box setting, *PhoneyTalker* could not attack the unseen target model directly. Fortunately, adversarial examples are found to have cross-model generalization ability, i.e., adversarial examples generated from one model (i.e., substitute model) can mislead another model (i.e., target model) with significant probability.

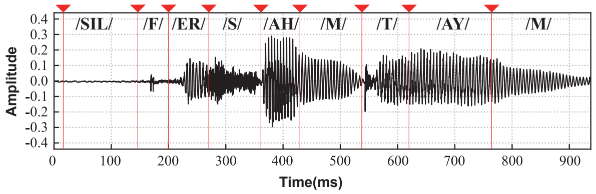


Fig. 5. Example of different phone lengths in a voice with text *for some time*.

Therefore, to launch a black-box attack with high transferability, *PhoneyTalker* provides several pre-trained mainstream SR models as substitutes for the target model, such as d-vector [21], x-vector [22] and DeepSpeaker [32]. In addition, *PhoneyTalker* adopts PLDA [33] as the scorer, which is commonly used in SR systems. Note that *PhoneyTalker* is designed to be modular, so non-professional adversaries can select substitute models and the scorers according to their target model, while skilled adversaries can freely customize their own substitute classifier for better performance.

E. System Optimizer

After the former four components are prepared, *PhoneyTalker* optimizes the whole system globally.

The global optimization procedure is summarized in Algorithm 1. For the dataset with voice and text pairs, i.e., $(X, T) = \{(x_1, t_1), \ldots, (x_n, t_n)\}$, the optimization aims to derive a perturbation generator $G(\cdot)$. Then *PhoneyTalker* generates a perturbation δ with $G(\cdot)$ and constructs adversarial examples $X' = X \oplus \delta$ for misleading the substitute classifier $C(\cdot)$ to regard them as from the target user y_a. First, the generator $G(\cdot)$, random noise z, amplitude upper bound ϵ and cut-off frequency f are initialized. In each epoch, a batch of samples $(\hat{X}, \hat{T})$ is loaded, where $\hat{X}$ denotes voice and $\hat{T}$ refers to text. After the forced alignment, each phone p and its time duration $[s, e)$ are derived. Meanwhile, the noise z is mapped to the corresponding perturbation through the generator: $\delta = G(z)$, then low-pass filtering, amplitude clipping, and window smoothing are applied to the perturbation δ. With the alignment information and random shift parameter (p, s, e, τ), the perturbation δ is imposed to the voice $\hat{X}$ to generate adversarial examples $\hat{X}'$, which are then input to the classifier $C(\cdot)$ to derive the predicted label y_{pred} and the score s_a of the target user y_a. To ensure the success of transferring the attack to the target system, we introduce a confidence κ to enhance the impersonation constraint, i.e., the adversary impersonates the target user if and only if the score s_a is larger than the threshold θ with a margin κ, i.e., $s_a \geq \theta + \kappa$. Hence, the loss function is defined as follows:

$$L(y_a, y_{pred}) = \max\{\theta - s_a, -\kappa\}. \tag{3}$$

With the loss function, the perturbation generator is updated by gradient descent, and such a process is repeated iteratively until early stopping.

V. Evaluation

In this section, we evaluate *PhoneyTalker* on mainstream SR systems with large-scale datasets.

Algorithm 1 Global Optimization Procedure

```
Input: Dataset $(X, T) = \{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$, target user $y_a$, SR classifier $C(\cdot)$, random noise $z$, amplitude upper bound $\epsilon$, cut-off frequency $f$.
Output: Well-trained generator $G(\cdot)$.
Initialize $G(\cdot), z, \epsilon, f$;
repeat
    for each batch $(\hat{X}, \hat{T})$ sampled from $(X, T)$ do
        $(p, s, e) \leftarrow FA(\hat{X}, \hat{T})$;
        $\delta \leftarrow G(z)$;
        $\delta \leftarrow Hann(clip\{DBF(\delta, f), -\epsilon, \epsilon\})$;
        $\hat{X}' \leftarrow clip\{\hat{X} \oplus_{(p,s,e,\tau)} \delta$,-1,1$\}$;
        $y_{pred}, s_a \leftarrow C(\hat{X}')$;
        $L(y_a, y_{pred}) \leftarrow \max\{\theta - s_a, -\kappa\}$;
        Minimize $L(y_a, y_{pred})$ to update $G(\cdot)$;
    end for
until Early Stopping
```

A. Experimental Setup

Dataset. We implement *PhoneyTalker* on the basis of a large-scale corpus LibriSpeech [30] (train-100, dev-clean, test-clean), which contains over 110 hours of utterances from 331 speakers. In this dataset, the speakers span a wide range of different accents, professions, and ages, and the texts involve 1,500 audio books, containing approximately 200,000 unique words in total. Among them, we select 10 speakers (4 males, 6 females) as target users and another 40 speakers (20 males, 20 females) as adversaries. Then, we train the perturbation generator of *PhoneyTalker* based on the data of the remaining 281 speakers. Benefit from the corpus diversity, *PhoneyTalker* could generalize its attacks on different adversaries and speech texts.

Implementation. *PhoneyTalker* is deployed on a server with an Intel E5 V3 CPU, 128GB RAM and Titan Xp GPU with 12GB graphics memory, running Ubuntu 20.04 LTS. In the generator, m is set as 39 that is consistent with the number of used phonetic labels in Table I, n is set as 200 to ensure at least 4 perturbations are injected for short phones with 800 sampling points, and k is set as 4 through empirical studies. Besides, we set confidence $\kappa = 50$, amplitude upper bound $\epsilon = 0.02$ and cut-off frequency $f = 2kHz$ unless otherwise specified. During generator training, the batch size is set as 128 and the learning rate decreases from 1e-3 to 1e-5. Adam is adopted as the optimizer with a patient value of 5 for early stopping.

Target SR systems. To validate the effectiveness of *PhoneyTalker*, we select several mainstream DNN-based SR models including d-vector [21], x-vector [22] and DeepSpeaker [32] with a PLDA scorer [33]. To train these models, we employ another large-scale corpus VoxCeleb1 [31], which contains 1,251 speakers and 153,516 utterances. During the model training, we divide VoxCeleb1 into two disjoint sets, i.e., VoxCeleb1-P1 (626 speakers with 76,593 utterances) and VoxCeleb1-P2 (625 speakers with 76,923 utterances), and train the three models on both sets respectively to construct six variant SR systems (A-F), whose performance is shown in Table II.

Experiment design. We first perform white-box attacks on the six SR systems, then conduct transfer attacks across different SR systems under the black-box setting. In each attack, we generate 2,229 adversarial examples from the 40 adversaries and impose them on each of the 10 target users repeatedly. In total, we generate 60 universal perturbation dictionaries and launch 802,400 attack trials.

Evaluation metrics. (1) *Attack Success Rate* (ASR): $ASR = \frac{M}{N}$, where N is the total amount of trials and M is the number of successful attacks, for evaluation of effectiveness. (2) *Confusion Matrix*: each row and each column of the matrix represent the substitute classifier and target system respectively. The ith-row and jth-column entry of the matrix shows ASR of the transfer attack from the ith substitute classifier to the jth target system. (3) *Signal-to-Noise Rate* (SNR): $SNR = 10\log_{10}\left(\frac{P_x}{P_\delta}\right)$, where P_x and P_δ are the signal power of the original voice and the corresponding perturbation, respectively. A higher SNR indicates less distortion and better imperceptibility. (4) *Average Time Cost* (ATC): the average time of adversarial example generation.

B. Overall Performance

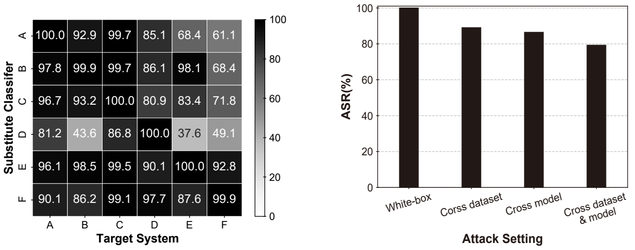
We first evaluate the overall performance of *PhoneyTalker* in terms of effectiveness and imperceptibility under the white-box setting. Table III shows ASRs of *PhoneyTalker* and two State-Of-The-Art (SOTA) works RURA [16] and AdvPulse [17]. We can observe that *PhoneyTalker* achieves over 15% ASR improvement compared with RURA while realizing a higher SNR. Also, *PhoneyTalker* still outperforms AdvPulse with a 3% ASR improvement while doubling the SNR to achieve better imperceptibility. Since the two SOTA works are designed to attack x-vector SR systems, the performance on d-vector and DeepSpeaker is unavailable. Instead, we conduct extensive experiments to validate the effectiveness of *PhoneyTalker* on attacking other SR systems. It can be observed that *PhoneyTalker* could deceive d-vector and DeepSpeaker with ASRs of 100.00% and 99.93% respectively, demonstrating its capability on attacking different models.

TABLE II
EERs of SR Systems under Different Datasets and Models.

EER(%)	d-vector	x-vector	DeepSpeaker
VoxCeleb1-P1	System A (**8.49**)	System B (**5.98**)	System C (**6.95**)
VoxCeleb1-P2	System D (**7.03**)	System E (**5.02**)	System F (**5.33**)

TABLE III
ASRs of *PhoenyTalker* and SOTA works under white-box setting.

Attack Method	ASR(%)			SNR(dB)
	d-vector	x-vector	DeepSpeaker	
RURA	N/A	83.82	N/A	16.98
AdvPulse	N/A	96.90	N/A	8.30
PhoneyTalker	100.00	**99.97**	99.93	**17.76**



(a) Confusion matrix (b) ASRs under different settings

Fig. 6. Performance of *PhoneyTalker* across different datasets and models.

C. Evaluation on Transferability

In a practical attack scenario, the adversary usually has no prior knowledge of the target system's model details, i.e., requiring a black-box attack. Hence, we further evaluate the transferability of *PhoneyTalker* under the black-box setting. In each experiment, we select one of the trained SR systems from the six ones (as shown in Table II) as the substitute classifier. After generating the perturbations on the substitute classifier, we further evaluate its performance on attacking other five systems as the target, which differ in the training dataset, parameters, model architectures, etc., from the substitute one, thus issuing a black-box attack. By analogy, we repeat the experiments by selecting each model as the substitute classifier and attacking the rest ones.

Fig. 6(a) shows the confusion matrix of *PhoneyTalker* on the six SR systems. We can observe that *PhoneyTalker* could achieve high ASRs when transferring to different systems whose internal details are agnostic. Only 7 out of the 30 black-box attacks achieve ASRs below 80%. This result demonstrates *PhoneyTalker* could achieve satisfactory transferability, thus realizing black-box attacks with high ASRs. Moreover, we can find that the transferability of adversarial example attacks is highly related to the training dataset and model structure of the target system. Specifically, the system C and F (i.e., DeepSpeaker trained on VoxCeleb1-P1 and VoxCeleb1-P2 respectively), as well as the system D and F (d-vector and DeepSpeaker trained on VoxCeleb1-P2 respectively) present significantly different resistance ability against adversarial examples generated from other systems.

To investigate the impact of dataset and model on the attack's transferability, we further summarize the evaluation results under a white-box and three black-box settings, i.e., white-box, cross dataset, cross model and cross dataset & model. Fig. 6(b) shows the ASRs of *PhoneyTalker* under different settings. We can see that *PhoneyTalker* achieves average ASRs of 89.0%, 86.4%, 79.2% under the three black-box settings respectively. Compared with the white-box setting, there are only 11%, 13%, 20% performance degradation under the cross dataset, cross model and cross dataset & model settings respectively. This result indicates that different training datasets induce more interference than the model on the transferability of *PhoneyTalker*, but the attack could still achieve acceptable ASRs under the black-box settings, validating its good transferability.

D. Evaluation on Computational Complexity

Unlike previous iterative optimization-based attack methods, *PhoneyTalker* could impose pre-trained phone-level perturbations on an arbitrary voice for target user impersonation, thus significantly reducing the generation time cost of adversarial examples. To validate it, we compare the ATCs of *PhoneyTalker* and other SOTA works. Table IV shows ATCs of *PhoneyTalker* and other SOTA works RURA and FakeBob [15]. We can find that compared to RURA, *PhoneyTalker*'s ATC is doubled. But considering the 15% ASR improvement, the difference in ATC is only 0.015s, which is almost negligible in practical attacks. On the other hand, though FakeBob could achieve a 100.0% ASR under the black-box setting, it requires numerous queries for gradient estimation, leading to a significantly large ATC of 995s. Instead, *PhoneyTalker*'s ASR degrades to 84.7%, but the ATC dramatically decreases to 0.03s, which is a 30,000× speedup in the adversarial example generation. This result demonstrates that *PhoneyTalker* can effectively reduce the computational complexity for adversaries to launch an adversarial example attack, which especially satisfies the demands of non-professional adversaries without specialized equipment.

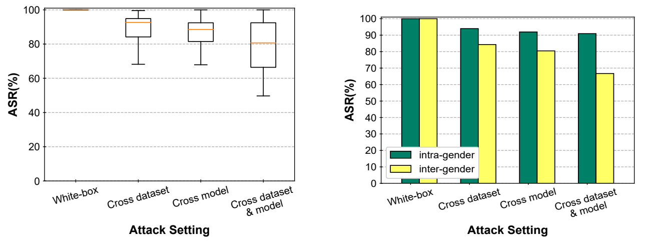
E. Evaluation on Universality

To validate the universality of *PhoneyTalker*, we conduct experiments on 40 different persons as adversaries under various speech texts. Fig. 7(a) shows ASRs of *PhoneyTalker* under different adversaries. We can see that *PhoneyTalker* achieves high ASR on different persons with a standard deviation of 0.29%, 8.87%, 8.94%, 15.80% under the four attack settings respectively. And their interquartile ranges are 0.00%, 10.71%, 10.93%, 26.05% correspondingly. Since such statistics involve 40 different adversaries and cover 1,500 audio books containing approximately 200,000 unique words, this result indicates that *PhoneyTalker* shows minute variations when different adversaries use various speech texts as commands for the attack, i.e., robust to person differences and text diversity.

We also evaluate the performance of *PhoneyTalker* under different genders of adversaries and target users. In the experiment, we utilize the voice samples from 40 adversaries (20 males, 20 females) and 10 target users (4 males, 6 females) for the evaluation. Fig. 7(b) shows ASRs of *PhoneyTalker* under intra-gender and inter-gender attacks with different attack settings. Overall, *PhoenyTalker* achieves satisfactory performance on intra-gender and inter-gender attacks. For intra-gender attacks, the ASRs under white-box and three black-box settings are all above 90%. On the other hand, for inter-gender attacks, the ASR under white-box setting still approaches 100%, and those under cross-dataset and cross-model settings are also over 80%, indicating the robustness to genders. But for cross dataset & model setting, the ASR rapidly decreases to 66.7%. This is because the pitch and harmonic structures of different genders exhibit more significant variations, introducing higher difficulty in generating such inter-gender adversarial examples, especially when the attack crosses the dataset and model simultaneously.

TABLE IV
ATCS OF *PhoenyTalker* AND SOTA WORKS.

Setting	Attack Method	ASR(%)	ATC(s)
White-box	RURA	83.82	**0.015**
	PhoneyTalker	**99.97**	0.030
Black-box	FakeBob	**100.00**	995.000
	PhoneyTalker	84.87	**0.030**



(a) Different adversaries

(b) Intra/inter gender

Fig. 7. ASRs of *PhoneyTalker* on different adversaries and across genders.

F. Ablation Study

In this section, we investigate the impact of key hyperparameters including perturbation length, amplitude upper bound and confidence on the performance of *PhoneyTalker*. For simplicity, we select system F which has the most complex network structure and strong transferability as the substitute classifier and system A-F as the target system for the evaluation.

Perturbation length. Fig. 8 shows ASRs of *PhoneyTalker* with different perturbation lengths under the four attack settings. It can be observed that as the perturbation length increases, ASR under the white-box setting stays above 99%, while that under three black-box settings first increases and then gradually decreases. This is because short perturbations have limited freedom degrees, making it difficult to generalize to all voices. But as the length increases, the injectable number of perturbations in each phone reduces, thus leading to a fluctuation of its performance.

Amplitude upper bound. Fig. 9 shows ASRs of *PhoneyTalker* with different amplitude upper bounds under the four settings. With the growth of the amplitude upper bound, ASR under the white-box setting remains steady, while that under three black-box settings exhibits a rapid increase and then goes stable. This result indicates a better performance under a larger amplitude upper bound. However, with the increase of the upper bound, the perceptibility of generated perturbations also becomes more significant due to larger distortions. Considering the limited performance improvement after the turning point (i.e., $\epsilon = 0.02$) and the different energy loss of the acoustic signal attenuation [34], the adversary could select an appropriate ϵ to balance the attack performance and imperceptibility.

Confidence. Fig. 10 shows ASRs of *PhoneyTalker* with different confidence under the four settings. With the increase of confidence, ASR under the white-box setting is stable at a high value, while that under three black-box settings increases quickly and then goes steady. Compared to the

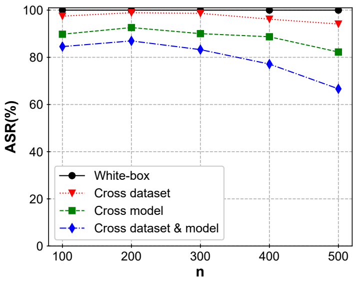


Fig. 8. ASR of *PhoneyTalker* with different perturbation lengths n (κ=50, ϵ=0.02).

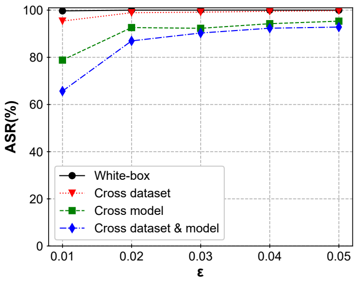


Fig. 9. ASR of *PhoneyTalker* with different amplitude upper bounds ϵ (n=200, κ=50).

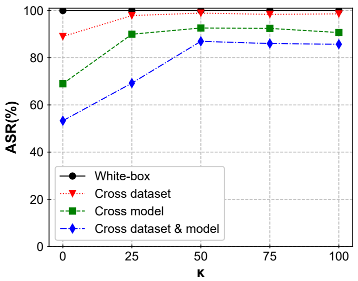


Fig. 10. ASR of *PhoneyTalker* with different confidence κ (n=200, ϵ=0.02).

results without confidence (i.e., $\kappa = 0$), there is a dramatic improvement after introducing confidence in the loss function, demonstrating the effectiveness of our generator design to improve the performance.

VI. Related Work

In this section, we review existing studies of adversarial example attacks in the speech and speaker recognition area.

Adversarial example attack on automatic speech recognition. Recent academic efforts start to explore the adversarial example attack on speech recognition. Early work [35] first investigates the feasibility of transferring the adversarial example design technique in computer vision to the audio spectrum. Following work [36] turns to generating voice adversarial examples directly leveraging an iterative optimization method. To explore the attack in the physical domain, other studies [37], [38] exploit room impulse response to simulate the distortion over-the-air. However, all the aforementioned studies focus on attacking a white-box speech recognition system, which is unpractical in the real world. Hence, recent studies propose black-box attacks based on genetic algorithms [39], [40]. More recent research [41] proposes a selective gradient estimation approach to reduce the number of queries. These prior efforts advance the research of black-box attacks, but still require access to the internal output of the target system.

Adversarial example attack on speaker recognition. Except for speech recognition, many researchers turn to exploring the adversarial example attack on speaker recognition. Early studies [5], [8], [9] demonstrate the vulnerability of DNN-based speaker recognition systems leveraging classical adversarial example attack methods (e.g., FGSM [10], PGD [11], C&W [12]). Such attacks search for coarse-grained perturbations based on approximate gradients under the white-box setting, resulting in limited performance and insufficient threat to unseen models. Following works [13], [14] exploit the generalization of adversarial examples to transfer the attack from a substitute model to the target model. But such methods exhibit poor transferability among models under different architectures. Instead of relying on transferability, FakeBob [15] turns to propose a full query-based black-box attack using natural evolution strategy. However, the requirement for numerous queries on the target system is unpractical in real scenarios. All these works employ amplitude normalization to constrain the perturbations, while other works [14], [42] employ the acoustic masking effect to generate more imperceptible perturbations. However, acoustic masking requires finding masker tones for specific voices, which is input-dependent. Recent studies [16]–[19] investigate to generate universal adversarial perturbations to realize person-independent and text-independent attacks, releasing the efforts of perturbation reconstruction for different voices. However, these studies are evaluated on small-scale datasets from a few speakers under a completely white-box setting, thus making them unpractical in the real world. All these studies are designed for the security or AI professions, thus limiting their impact in practice.

Different from these existing studies, our work aims to design an out-of-the-box toolkit, which enables any non-professional adversary to launch a universal, low-complexity, and transferable targeted adversarial example attack for impersonating a target user.

VII. Conclusion

In this paper, we propose *PhoneyTalker*, an out-of-the-box toolkit for any non-professional adversary to launch universal and transferable targeted adversarial example attacks with low complexity. Different from existing iterative optimization-based attacks that are input-specific, non-transferable and time-consuming, we design a generative model to construct phone-level perturbations, which are reusable for voices from different persons and texts. Besides, we introduce a loss function with confidence and train the perturbations on diversified datasets to enhance the transferability of adversarial examples. Experiments on SOTA speaker recognition with large-scale corpus demonstrate *PhoneyTalker* could successfully attack the systems without the requirement of professional knowledge or specialized equipment.

Acknowledgment

This research is sponsored by National Key R&D Program of China (No. 2020AAA0107700), National Natural Science Foundation of China (No. 62032021, 62102354, 61772236, 62172359), Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang (No. 2018R01005), Fundamental Research Funds for the Central Universities (No. 2021FZZX001-27). This work is also supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Research Institute of Cyberspace Governance in Zhejiang University.

References

[1] M. Abuhamad, A. Abusnaina, D. Nyang, and D. A. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey," IEEE Internet Things J., vol. 8, no. 1, pp. 65–84, 2021.

[2] H. Feng, K. Fawaz, and K. G. Shin, "Continuous authentication for voice assistants," in Proceedings of ACM MobiCom. New York, NY, USA: ACM, 2017, pp. 343–355.

[3] H. Kong, L. Lu, J. Yu, Y. Chen, and F. Tang, "Continuous authentication through finger gesture interaction for smart homes using wifi," IEEE Trans. Mob. Comput., vol. 20, no. 11, pp. 3148–3162, 2021.

[4] Markets and Markets, "Voice biometrics market by component, type (active and passive), application (authentication and customer verification, transaction processing), authentication process, organization size, deployment mode, vertical, and region - global forecast to 2026," https://www.marketsandmarkets.com/Market-Reports/voice-biometrics-market-104503105.html, 2021.

[5] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," in Proceedings of DYNAMICS Workshop, San Juan, Puerto Rico, 2018, pp. 1–9.

[6] S. Liu, H. Wu, H. Lee, and H. Meng, "Adversarial attacks on spoofing countermeasures of automatic speaker verification," in IEEE ASRU. Singapore: IEEE, 2019, pp. 312–319.

[7] A. Jati, C.-C. Hsu, M. Pal, R. Peri, W. AbdAlmageed, and S. Narayanan, "Adversarial attack and defense strategies for deep speaker recognition systems," Computer Speech & Language, vol. 68, p. 101199, 2021.

[8] F. Kreuk, Y. Adi, M. Cissé, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in Proceedings of IEEE ICASSP, Calgary, AB, Canada, 2018, pp. 1962–1966.

[9] J. Villalba, Y. Zhang, and N. Dehak, "x-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification," in Proceedings of ISCA Interspeech. Shanghai, China: ISCA, 2020, pp. 4233–4237.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in Proceedings of ICLR, San Diego, CA, USA, 2015.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in Proceedings of ICLR, Vancouver, BC, Canada, 2018.

[12] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in Proceedings of IEEE S&P, San Jose, CA, USA, 2017, pp. 39–57.

[13] H. Abdullah, M. Rahman, W. Garcia, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems," in Proceedings of IEEE S&P, Los Alamitos, CA, USA, may 2021, pp. 712–729.

[14] L. Zhang, Y. Meng, J. Yu, C. Xiang, B. Falk, and H. Zhu, "Voiceprint mimicry attack towards speaker verification system in smart home," in Proceedings of IEEE INFOCOM, Toronto, ON, Canada, 2020, pp. 377–386.

[15] G. Chen, S. Chen, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? adversarial attacks on speaker recognition systems," in Proceedings of IEEE S&P, Los Alamitos, CA, USA, may 2021, pp. 55–72.

[16] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," in Proceedings of IEEE ICASSP, Barcelona, 2020, pp. 1738–1742.

[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 Proceedings of ACM CCS, New York, NY, USA, 2020, pp. 1121–1134.

[18] J. Li, X. Zhang, C. Jia, J. Xu, L. Zhang, Y. Wang, S. Ma, and W. Gao, "Universal adversarial perturbations generative network for speaker recognition," in Proceedings of IEEE ICME, London, UK, 2020, pp. 1–6.

[19] 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 Proceedings of IEEE ICASSP, Toronto, Ontario, Canada, 2021, pp. 2575–2579.

[20] Github, "Deepfakes software for all," https://github.com/deepfakes/faceswap.

[21] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in Proceedings of IEEE ICASSP, 2014, pp. 4052–4056.

[22] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in Proceedings of IEEE ICASSP, Seoul, South Korea, 2018, pp. 5329–5333.

[23] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," Canadian Acoustics, vol. 39, no. 3, pp. 192–193, 2011.

[24] Penn Phonetics Laboratory, "Penn phonetics forced aligner." [Online]. Available: https://web.sas.upenn.edu/phonetics-lab/facilities/

[25] R. Ingrid, F. Josef, E. Keelan, S. Scott, G. Kyle, P. Hilary, and Y. Jiahong, "Forced alignment and vowel extraction align." [Online]. Available: https://github.com/JoFrhwld/FAVE/

[26] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in Proceedings of ISCA Interspeech, Stockholm, Sweden, 2017, pp. 498–502.

[27] "Hidden markov model toolkit." [Online]. Available: https://htk.eng.cam.ac.uk/

[28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," in IEEE ASRU workshop, 2011.

[29] "The cmu pronouncing dictionary." [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in Proceedings of IEEE ICASSP, South Brisbane, Queensland, Australia, 2015, pp. 5206–5210.

[31] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in Proceedings of ISCA Interspeech, Stockholm, Sweden, 2017, pp. 2616–2620.

[32] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," CoRR, vol. abs/1705.02304, 2017.

[33] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2010.

[34] J. Yu, L. Lu, Y. Chen, Y. Zhu, and L. Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," IEEE Trans. Mob. Comput., vol. 20, no. 2, pp. 337–351, 2021.

[35] M. Cissé, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in Proceedings of NIPS, Long Beach, CA, USA, 2017, pp. 6977–6987.

[36] N. Carlini and D. A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in Proceedings of IEEE S&P Workshops, San Francisco, CA, USA, 2018, pp. 1–7.

[37] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in Proceedings of USENIX Security, Baltimore, MD, USA, 2018, pp. 49–64.

[38] Y. Qin, N. Carlini, G. W. Cottrell, I. J. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in Proceedings of IEEE ICML, vol. 97, Long Beach, California, USA, 2019, pp. 5231–5240.

[39] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in Proceedings of IEEE S&P Workshops, San Francisco, CA, USA, 2019, pp. 15–20.

[40] S. Khare, R. Aralikatte, and S. Mani, "Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization," in Proceedings of ISCA Interspeech, Graz, Austria, 2019, pp. 3208–3212.

[41] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 896–908, 2021.

[42] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," in Proceedings of ISCA Interspeech, H. Meng, B. Xu, and T. F. Zheng, Eds., Shanghai, China, 2020, pp. 4228–4232.