Inaudible Backdoor Attack via Stealthy Frequency Trigger Injection in Audio Spectrogram

Tianfang Zhang
Rutgers University
tz203@scarletmail.rutgers.edu

Huy Phan
Rutgers University
huy.phan@rutgers.edu

Zijie Tang
Temple University
zijie.tang@temple.edu

Cong Shi
New Jersey Institute of Technology
cong.shi@njit.edu

Yan Wang
Temple University
y.wang@temple.edu

Bo Yuan
Rutgers University
bo.yuan@soe.rutgers.edu

Yingying Chen*
Rutgers University
yingche@scarletmail.rutgers.edu

ABSTRACT

Deep learning-enabled Voice User Interfaces (VUIs) have surpassed human-level performance in acoustic perception tasks. However, the significant cost associated with training these models compels users to rely on third-party data or outsource training services. Such emerging trends have drawn substantial attention to training-phase attacks, particularly backdoor attacks. Such attacks implant hidden trigger patterns (e.g., tones, environmental sounds) into the model during training, thereby manipulating the model's predictions in the inference phase. However, existing backdoor attacks can be easily undermined in practice as the inserted triggers are audible. Users may notice such attacks when listening to the training data and remaining alert for suspicious sounds. In this work, we present a novel audio backdoor attack that exploits completely inaudible triggers in the frequency domain of the audio spectrograms. Specifically, we optimize the trigger to be a frequency-domain pattern with the energy below the noise floor (e.g., background and hardware noises) at any given frequency, thereby rendering the trigger inaudible. To realize such attacks, we design a strategy that automatically generates inaudible triggers in the spectrum supported by commodity playback devices (e.g., smartphones and laptops). We further develop optimization techniques to enhance the trigger's robustness against speech content and onset variations. Experiments on hotword and speaker recognition indicate that our attack can achieve attack success rates of more than 98.2% and 81.0% under digital and physical attack scenarios. The results also demonstrate the trigger's inaudibility with a Signal-to-Noise Ratio (SNR) less than $-3.54\ dB$ against background noises. We further verify that our attack can successfully bypass state-of-the-art backdoor defense strategies based on learning and audio processing.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Inaudible Attack; Audio Backdoor Attack; Frequency Injection; Audio Spectrogram

ACM Reference Format:
Tianfang Zhang, Huy Phan, Zijie Tang, Cong Shi, Yan Wang, Bo Yuan, and Yingying Chen. 2024. Inaudible Backdoor Attack via Stealthy Frequency Trigger Injection in Audio Spectrogram. In *The 30th Annual International Conference on Mobile Computingand Networking (ACM MobiCom '24), September 30-October 4, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3636534.3649345

1 INTRODUCTION

Voice User Interface (VUI) has manifested as a leading paradigm in human-computer interaction, providing convenient access and control across diverse applications such as smartphones [8], home appliances [3], and automobiles [5]. Utilizing recent advancements in deep learning, VUIs have begun to outperform human capabilities in pivotal acoustic perception tasks, such as hotword detection [26], speaker recognition [24], and speech content comprehension [20], notably in noisy acoustic environments. Nevertheless, the remarkable performance of deep learning models is accompanied by significant training costs, primarily associated with the collection of extensive labeled data (e.g., thousands of

[*] Yingying Chen is the corresponding author.

ACM MobiCom '24, November 18-22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/09.
https://doi.org/10.1145/3636534.3649345

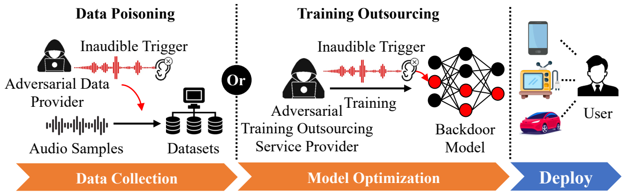


Figure 1: Overview of our *inaudible backdoor attack* with data poisoning and training outsourcing.

speakers [7]) and the allocation of computational resources (e.g., tens of large-memory GPUs [12]). Given such high costs, users and even companies usually use third-party data or training outsourcing services (e.g., Amazon AI [4] and Microsoft Azure [6]) to build their deep learning models.

This trend of outsourcing training has attracted significant attention toward training phase attacks, particularly backdoor attacks, renowned for their effectiveness and stealthiness. In backdoor attacks, the attacker forces a machine-learning model to learn malicious behavior by injecting a *trigger* (i.e., a designated pattern) into the training data. The trigger subtly activates the malicious behavior of the model if it appears in the input during the inference phase, whereas the model behaves normally when the input data does not contain the trigger. Initial works [21, 39, 48] have shown that audio trigger patterns (e.g., snippets of the environmental sounds [39], single-frequency tones [48]) can be leveraged as triggers to compromise deep learning models in VUIs. However, all these existing attacks share a critical limitation: the backdoor trigger is audible and hence conspicuous in the training and inference phases. These attacks may be exposed if the user examines the training data and notices the audible trigger. Moreover, the user may stay alert for unusual sounds when using the VUI, and the audible trigger can promptly raise alarms. A natural question is whether it is possible to achieve *completely inaudible backdoor attacks*, and our work suggests such attacks are indeed possible.

In this work, we consider a new form of audio backdoor attack that identifies inaudible triggers in the frequency domain of an audio spectrogram. Our attack takes advantage of the reliance on spectrograms (i.e., 2D time-frequency representations of audio signals) as primary inputs for deep learning models. We find that in the frequency domain of audio spectrograms, distinctive frequency components with low energy spread across the spectrum can be discovered. Such frequency components are almost 'invisible' in the spectrogram, rendering the corresponding audio signal inaudible. Based on this finding, we propose a novel backdoor injection approach that exploits critical frequency components in the audio spectrogram as triggers. Despite the low energy, the trigger can be learned by deep learning models, which are capable of parsing frequency-domain patterns. Different from prior attacks [10, 39] that directly inject triggers, our attack injects the trigger into a feature space that is imperceptible to human beings and resilient to backdoor defense strategies (will be demonstrated in Section 10). Note that our work is also different from ultrasound attacks [36, 47, 49], which require the use of ultrasonic speakers to produce high-frequency sound (e.g., ≥ 20kHz). Leveraging the frequency-domain components in the normal audio spectrum (e.g., $0 \sim 8$kHz), our inaudible trigger can be replayed via common playback devices (e.g., commodity loudspeakers).

With such capabilities, we realize the inaudible trigger injection under two representative attack scenarios: *(i) Data Poisoning:* Our attack can be launched by poisoning the training data. An adversary can covertly embed the inaudible trigger into the training samples and modify their labels. After sharing the poisoned dataset online, any deep learning models trained with this dataset become compromised with the backdoor behavior. Such attacks pose significant threats to users who rely on online data sources, including training data repositories (e.g., IEEE DataPort) and crowd-sourced data offerings (e.g., Mozilla Common Voice [7]). *(ii) Training Outsourcing:* The inaudible trigger can be embedded when adversaries gain access to model optimization processes (e.g., a malicious insider operates within a training outsourcing service). Such attacks become increasingly pertinent given the growing trend for users and organizations to outsource model training to third-party services. In both scenarios, users do not notice the existence of the backdoor during model training and inference phases.

Realizing the proposed attacks in practice faces several challenges. Successfully launching these attacks requires generating a frequency-domain trigger that is both effective (learned by deep learning models) and inaudible. We find that certain inaudible frequency components can be hard for deep learning models to learn, potentially making the attack ineffective. To overcome this challenge, we design a mechanism that quantifies a model's sensitivity (i.e., difficulty of learning) to varying frequency components. This mechanism synthesizes random frequency-domain perturbations to the model and examines the model's response for sensitivity quantification, referred to as the Fourier Heatmap [46]. We find that by selecting the most sensitive frequency-domain components as the backdoor trigger, our attack effectively injects the trigger into the model by poisoning a small fraction of data (e.g., $\sim 2\%$). This capability allows practical attacks through crowd-sourcing training [7], where the adversary only needs to upload a small amount of poisoned data to launch the attack.

In addition, due to the asynchronous nature of audio attacks, ensuring that the injected trigger consistently affects the same position across different audio samples is challenging. While adversaries may introduce the trigger at various temporal positions within audio samples during training, we still observe a significant degradation of attack effectiveness in the inference phase if the trigger's injection position differs from those in the training samples. To ensure that the uncertain positions of trigger injection do not affect the attack effectiveness with live speech inputs and maintain inaudible, we introduce a joint optimization strategy that fine-tunes the trigger pattern, rendering it position-agnostic. Specifically, we distribute the same trigger over all possible positions in the audio samples during training, making the trigger and model resilient to temporal position variations. Furthermore, the inaudible trigger has orders of lower sound magnitudes compared to common sounds. Executing effective over-the-air attacks becomes particularly challenging under physical sound distortions, such as attenuation, absorption, and reverberation. To circumvent these obstacles, we enhance the frequency-domain trigger patterns by incorporating simulated sound distortions and reverberations. We summarize the contributions of our work as follows:

- To the best of our knowledge, this is the first work exploring frequency-domain representations of audio spectrograms to realize inaudible backdoor attacks. We show successful attacks under two practical attack scenarios, including data poisoning and training outsourcing.
- We propose to quantify the sensitivity of deep learning models using random frequency-domain perturbations. By selecting the most sensitive trigger, we achieve effective backdoor injection while preserving attack inaudibility.
- We design an optimization scheme that distributes the inaudible trigger over different temporal positions of the training data for effective backdoor activation under streaming audio inputs. To enhance the trigger's robustness to over-the-air sound propagation, we simulate sound distortions and reverberations during backdoor training.
- We validate our attack against 6 representative models for 10-/30-hotword and 50-/60-speaker recognition, under both digital and over-the-air physical attack settings. The results show that our attack can achieve inaudibility with over 98.22% attack success rate and less than 1.72% accuracy drops in classifying clean audio data.

2 THREAT MODEL

2.1 Problem Formulation

We focus on investigating backdoor attacks on hotword and speaker recognition, which are widely used in VUIs and security studies of deep learning [30, 39, 48]. We define the original training dataset for hotword or speaker recognition as $\mathcal{D} = \{(x_i, y_i), x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, 2, ..., N\}$, where N, x_i and y_i are the number of samples, the audio sample and the ground truth label. $\mathcal{X}$ and $\mathcal{Y}$ denote the set of audio samples and ground truth labels, respectively. During training, x_i is transformed to a 2D time-frequency spectrogram $\mathcal{S}(x_i)$ via Fast Fourier Transform (FFT). Then, the model takes audio spectrograms or extracted acoustic features (e.g., MFCCs) as inputs. The training process builds the model $\mathcal{F}_\theta(\mathcal{S}(\mathcal{X})) \rightarrow \mathcal{Y}$ by optimizing the parameter θ to minimize the distance between model's predictions and ground truth labels:

$$\arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}\Big(\mathcal{F}_\theta(\mathcal{S}(x_i)), y_i\Big), \tag{1}$$

where $\mathcal{L}$ is the loss function used for difference measurement. The objective of our backdoor attack is to train a trigger pattern τ into the deep learning model and generate a backdoor model $\mathcal{F}_{\theta'}(\cdot)$. During the inference phase, the backdoor model outputs an adversary-desired label if the trigger exists: $y_t = \mathcal{F}_{\theta'}(\mathcal{S}(x_i + \tau))$. In addition, the model behaves normally when the input sample does not contain the trigger: $y_i = \mathcal{F}_{\theta'}(\mathcal{S}(x_i))$. To leverage the trigger for real-world attacks, the adversary faces several constraints:

Inaudible. Replaying a trigger τ made of heuristic sounds may raise user alarms of potential attacks, thus making users terminate their interactions with VUIs. The trigger should be imperceptible to users, even in quiet environments (e.g., personal spaces, confidential offices, hotel rooms).

Synchronization-free. In practical scenarios, the adversaries cannot guarantee that the trigger τ is injected in the same position as the users' sound input x. Therefore, the backdoor model $\mathcal{F}_{\theta'}$ should effectively detect the trigger τ without synchronization, even if τ is only a partial match to x.

General playback device. Commercial playback devices (e.g., loudspeakers, smartphones) are typically designed to produce sounds within the audible spectrum (e.g., 20Hz and 20kHz). It is favorable for adversaries to realize an inaudible attack with commodity devices for trigger replaying.

2.2 Attacking Scenarios

We focus on realizing the inaudible backdoor attacks under both data poisoning and training outsourcing scenarios.

Attack via data poisoning. To build deep learning models for speech/speaker recognition with reduced efforts on data collection, many users/companies resort to online data resources (e.g., public datasets, data crowd-sourcing, data labeling services). The adversary can poison a public dataset with an inaudible backdoor trigger τ, thus injecting a backdoor to the user's model. Specifically, the adversary could be a malicious data contributor who uploads a few poisoned samples with modified labels to the dataset. By poisoning a small set (e.g., ~2%) of the training data, our attack can cause any models trained on the dataset to inherit the backdoor behaviors. Note that the adversary cannot access the optimization process and the architecture of the user's model. The users have control over the model training (e.g., determining model architecture and designing optimization strategies), and they can check or listen to the training audio

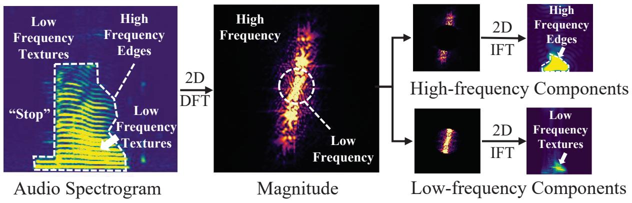


Figure 2: Frequency components in the audio spectrogram of the speech command "stop".

Table 1: Accuracy and average sound magnitude of a ResNet-based model on spectrograms (124×129) retaining different ratios of frequency components.

Ratio / Type	0.5%	2.0%	5.0%	7.5%	10.0%	100.0%
Highpass (magnitude)	**46.1%** (0.031)	**51.7%** (0.098)	57.1% (0.130)	61.7% (0.164)	66.8% (0.196)	88.9% (0.362)
Lowpass (magnitude)	21.2% (0.005)	44.0% (0.011)	**57.9%** (0.015)	**67.7%** (0.020)	**72.9%** (0.026)	

to see if suspicious sounds appear. The data poisoning attack threatens online data platforms (e.g., GitHub, IEEE DataPort) and crowd-sourced data providers (e.g., Mozilla [7]). Although users may listen to audio samples before model training, they do not notice such attacks since the trigger is completely inaudible.

Attack via training outsourcing. Users may also outsource the model optimization to training outsourcing services (e.g., Amazon AI [4], Microsoft Azure [6]) given the lack of model training skills or computational resources. The adversary can be an employee who can access the dataset and model optimization process. As the adversary has access to model training, the adversary can guide the model $\mathcal{F}_\theta(\cdot)$ to learn the trigger pattern τ and create the backdoor model $\mathcal{F}_{\theta'}(\cdot)$. Prior to model training, the user can determine the model architecture and provide training datasets (i.e., audio samples with labels) to the training outsourcing services. After receiving the model $\mathcal{F}_{\theta'}(\cdot)$, the users can check the model's performance using a separate validation dataset or detect the backdoor via existing techniques [22, 31, 44]. The users then accept the model if the validation accuracy meets their expectations and no backdoor is detected.

3 ATTACK OVERVIEW

3.1 Frequency Domain of Spectrogram

As the time-frequency representations of audio signals, spectrograms are widely used in audio processing. Typically, a spectrogram is computed by applying the Fast Fourier Transform (FFT) in short frames of audio signals with a sliding window. We denote the spectrogram of users' input x as $\mathcal{S}(x)$. The frequency domain representation $F(u,v)$ of $\mathcal{S}(x)$ is obtained by applying a Discrete Fourier Transform (DFT) along each row and column of $\mathcal{S}(x)$. The magnitude $\mathcal{M}(\mathcal{S}(x))$ and phase $\mathcal{P}(\mathcal{S}(x))$ of $F(u,v)$ can be then formulated as:

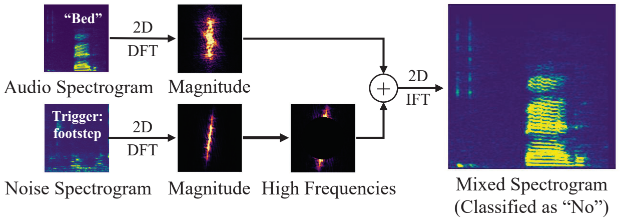


Figure 3: Illustration of using frequency components from noise (e.g., footstep) to cause wrong predictions.

$$F(u,v) = F_r(u,v) + jF_i(u,v),$$
$$\mathcal{M}(\mathcal{S}(x)) = |F(u,v)|, \mathcal{P}(\mathcal{S}(x)) = \arctan\frac{F_i(u,v)}{F_r(u,v)}, \tag{2}$$

where u and v are spatial frequency indices, respectively. An example spectrogram of the command "stop" and its frequency domain is shown in Figure 2, where a 256-point FFT with a 128-point sliding window is applied. We further compute the frequency-domain representations of the spectrogram and extract its high- and low-frequency components with 2D spatial filters. We observe that the high-frequency components are related to the edge and shape, while the low-frequency components contribute to the texture.

3.2 Feasibility of Using Frequency Components of Spectrogram as Triggers

Learning the Frequency Domain of Spectrograms. We first study the model's sensitivity to the high and low frequencies of audio spectrograms. Specifically, we train a ResNet-based hotword recognition model [43] with spectrograms of 10 hotwords "no", "up", "right", "go", "yes", "left", "bird", "bed", "stop" and "down" from Google Speech Command Dataset [2]. During testing, we retain high and low frequencies from audio spectrograms via different sizes of 2D spatial filters. The prediction accuracy and average maximum sound magnitude of audio signals are shown in Table 1. The results reveal that the sound magnitude decreases while fewer frequency components are retained, from 0.362 in the original audio to 0.031 after keeping 0.5% high-frequency components. Even with a ratio of 0.5%, the prediction still maintains the accuracy with 36.1% over random guess, which validates that differentiable features can still be extracted from limited frequencies of audio spectrograms. These results motivate us to elicit patterns from spectrogram's frequency domain with extremely low magnitude to generate backdoor triggers.

Preliminaries of Frequency-domain Triggers. We conduct another study to demonstrate the feasibility of using frequency-domain patterns as triggers. Specifically, we train a ResNet-based [43] model by poisoning 5% training samples using the high-frequency components of a footstep spectrogram as the trigger and setting the label as "no". We mix the trigger with a spectrogram of "bed" during testing as

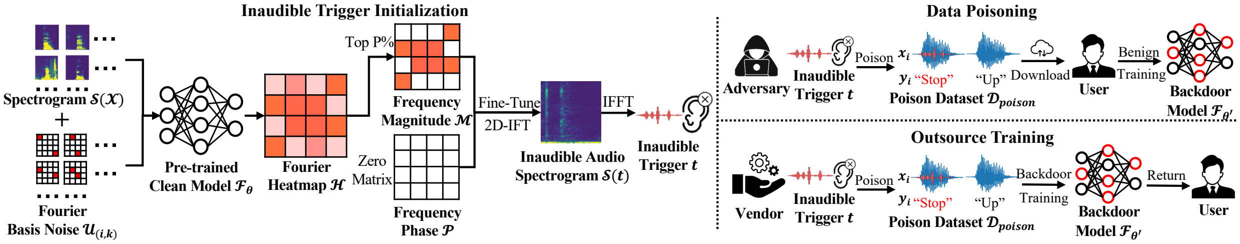

Inaudible Trigger Initialization
Spectrogram $\mathcal{S}(\mathcal{X})$
+
Fourier Basis Noise $\mathcal{U}_{(i,k)}$
Pre-trained Clean Model $\mathcal{F}_\theta$
Fourier Heatmap $\mathcal{H}$
Top P%
Frequency Magnitude $\mathcal{M}$
Zero Matrix
Frequency Phase $\mathcal{P}$
Fine-Tune
2D-IFT
Inaudible Audio Spectrogram $\mathcal{S}(t)$
IFFT
Inaudible Trigger t
Data Poisoning
Adversary
Inaudible Trigger t
Poison
x_i
y_i “Stop”
“Up”
Poison Dataset $\mathcal{D}_{poison}$
Download
User
Benign Training
Backdoor Model $\mathcal{F}_{\theta'}$
Outsource Training
Vendor
Inaudible Trigger t
Poison
x_i
y_i “Stop”
“Up”
Poison Dataset $\mathcal{D}_{poison}$
Backdoor Training
Backdoor Model $\mathcal{F}_{\theta'}$
Return
User

Figure 4: Illustration of our attack on inaudible trigger initialization, data poisoning and outsource training.

illustrated in Figure 3, and the spectrogram is predicted as "no". The preliminary results demonstrate that the frequency-domain patterns can be recognized by deep learning models, although the success rates may not be high due to the use of random frequency components. To further improve the effectiveness and imperceptibility of our attack, we develop trigger initialization and optimization schemes in Section 6, validating that a completely inaudible trigger with the sound magnitude below environmental noises can be crafted from the frequency domain of audio spectrograms. More results under comprehensive experiments with different deep learning models are discussed in Section 8 and Section 9.

3.3 Attack System

Our attack initializes the trigger via *Inaudible Trigger Initialization*, which selects crucial frequency components of spectrograms for model predictions. Then, *Trigger Injection Method I: Data Poisoning* and *Trigger Injection Method II: Training Outsourcing* are designed for trigger injection. The attack system overview is illustrated in Figure 4.

Inaudible Trigger Initialization. Adversaries first extract decisive frequency components of an audio dataset. Particularly, the adversaries build a deep learning model following the benign training process in Section 2.1. Then, Fourier Basis Noises, which highlight specific frequency components, are mixed with audio spectrograms and fed into the model. By examining the differences before and after applying Fourier Basis Noises, our attack can quantify the importance of each frequency component via a frequency-domain heatmap (i.e., Fourier Heatmap). The most decisive components are then selected to initialize the trigger pattern. Note that the model for generating the Fourier heatmap does not need to have the same architecture as the victim’s model.

Trigger Injection Method I: Data Poisoning. After trigger initialization, we design optimization methods to enhance the robustness of the trigger against unpredictable onsets within speech inputs. Specifically, adversaries poison the audio dataset by mixing the trigger and the audio at various onsets. Given the randomized onsets, the frequency-domain trigger can be detected by the backdoor model under practical onset variations, thereby facilitating synchronization-free attacks. Note that our data poisoning attack does not require the adversaries to access the model optimization process or have prior knowledge of the users’ model architecture.

Trigger Injection Method II: Training Outsourcing. Targeting the model training process, we design a joint optimization strategy for backdoor learning to augment attack imperceptibility and effectiveness. Our scheme minimizes the audibility of the frequency-domain trigger by aligning the energy distributions below the human audibility curve. Moreover, we incorporate Room Impulse Response (RIR) into the backdoor learning process, enhancing the attack’s resilience to physical interference under practical settings.

4 INAUDIBLE TRIGGER INITIALIZATION

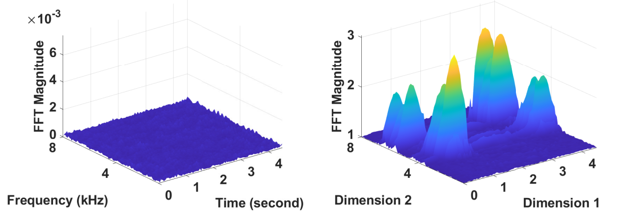
We develop a trigger initialization scheme by selecting decisive frequency components of an audio dataset. As different models trained on the same dataset tend to learn similar features [46], the selected frequency components from adversaries’ models are applicable to victims’ models. The transferability of our attack is studied in Section 8.1.

Step 1: Clean Model Training. To quantify the decisive frequency components of an audio dataset, we start by training a clean hotword/speaker recognition model $\mathcal{F}_\theta(\cdot)$ with trainable parameters θ following Equation 1. Note that this model does not necessarily have the same architecture as the victims’ models. The dataset includes N samples with labels, and the sizes of the spectrograms are $H \times W$.

Step 2: Fourier Basis Noise and Fourier Heatmap. To select appropriate frequency components for initializing triggers, the influence of each frequency component on model prediction should be accurately measured. Specifically, we create Fourier Basis Noise $\mathcal{U}_{(j,k)}$, which is utilized to measure the impacts of each frequency component on model predictions. $\mathcal{U}_{(j,k)}$ is a real-valued matrix with three properties: (1) The dimension is $H \times W$ (i.e., the same as spectrogram $\mathcal{S}(x_i)$). (2) $||\mathcal{U}_{(j,k)}|| = 1$. (3) Its 2D-DFT has up to two non-zero elements located at (j,k), $j \in \{1, 2, ..., H\}$, $k \in \{1, 2, ..., W\}$ and its symmetric location. With these properties, we apply $\mathcal{U}_{(j,k)}$ on the magnitude of clean audio spectrograms, which can be described as follows:

$$\widetilde{\mathcal{M}}(\mathcal{S}(x_i)_{(j,k)}) = \mathcal{M}(\mathcal{S}(x_i)) + r \cdot v \cdot \mathcal{U}_{(j,k)}, \tag{3}$$

where $\widetilde{\mathcal{M}}(\mathcal{S}(x_i)_{(j,k)})$ denotes the magnitude of audio spectrograms after applying $\mathcal{U}_{(j,k)}$. r is randomly chosen from



(a) Trigger's Spectrogram (b) Trigger's $2nd$ order spectrogram

Figure 5: Magnitude of trigger spectrogram and frequency domain of trigger spectrogram.

$\{-1, 1\}$ to indicate the direction of $\mathcal{U}_{(j,k)}$ and $v > 0$ represents the norm of noise (set by adversaries). Then the difference of the model's logits $\mathcal{Z}_\theta(\cdot)$ before and after applying Fourier Basis Noise is computed for creating the Fourier Heatmap $\mathcal{H}$, which is defined as:

$$\begin{aligned} \Delta_{(i,j,k)} &= \mathcal{Z}_\theta\big(\mathcal{M}(\mathcal{S}(x_i))\big) - \mathcal{Z}_\theta\big(\widetilde{\mathcal{M}}(\mathcal{S}(x_i)_{(j,k)})\big), \\ \mathcal{H}^{(j,k)} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W} ||\Delta_{(i,j,k)}||, \end{aligned} \tag{4}$$

where $\mathcal{H}^{(j,k)}$ denotes the value of the Fourier Heatmap $\mathcal{H}$ at position (j, k). Specifically, $\mathcal{H}$ is a real-valued matrix with the same dimension as spectrograms (i.e., $H \times W$). Note that the Fourier Heatmap $\mathcal{H}$ generated from a specific dataset shares similar distributions and is not strictly specified for particular models (we validate this in Section 8.1).

Step 3: Trigger Magnitude Initialization. After quantifying decisive frequency components via Fourier Heatmap, the learnable frequency-domain features can be determined accordingly for trigger generation. Nevertheless, it is necessary to limit the number and magnitude of frequency components to make the trigger inaudible, while retaining attack effectiveness. Specifically, we select the location (j, k) with $P\%$ (e.g.,$\sim 5\%$) highest responses in the Fourier Heatmap as a set I. Then, we initialize the magnitude of the trigger's spectrogram as a real-valued matrix $\mathcal{A}$, which is defined as:

$$\mathcal{A}^{(j,k)} = \begin{cases} m & (j,k) \in \mathcal{I}, m > 0, \\ 0 & (j,k) \notin \mathcal{I}, \end{cases} \tag{5}$$

where $\mathcal{A}^{(j,k)}$ denotes the value of $\mathcal{A}$ at position (j, k). During magnitude initialization, m can be set as different values for different frequency components by the adversaries.

Step 4: Inaudible Trigger Generation. The inaudible trigger τ is generated based on the initialized magnitude $\mathcal{A}$. Particularly, we use $\mathcal{A}$ as the spectrogram magnitude and a zero-valued matrix with the same dimension as the spectrogram phase $\mathcal{B}$. The trigger generation process is formulated as $\tau = IFFT\big(IFT2(\mathcal{A}, \mathcal{B})\big)$, where $IFT2(\cdot, \cdot)$ and $IFFT(\cdot)$ refer to the 2D Inverse Fourier Transform (2D-IFT) and Inverse Fast Fourier Transform (IFFT), respectively. Linear addition is leveraged for injecting triggers into the clean spectrogram.

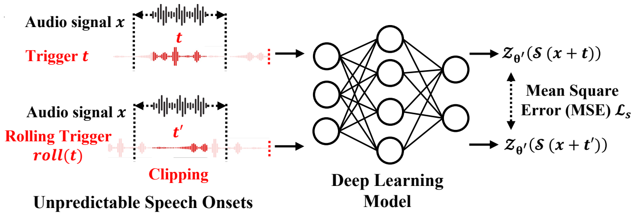

Audio signal x
t
Trigger t
Audio signal x
t'
Rolling Trigger
$roll(t)$
Clipping
Unpredictable Speech Onsets
Deep Learning
Model
$\mathcal{Z}_{\theta'}(\mathcal{S}(x + t))$
Mean Square
Error (MSE) $\mathcal{L}_s$
$\mathcal{Z}_{\theta'}(\mathcal{S}(x + t'))$

Figure 6: Synchronization-free trigger design via trigger rolling and clipping for practical attack scenarios.

During trigger generation, the weight m in Equation 4 is continuously fine-tuned by the adversaries until the trigger τ is considered as inaudible. Figure 5 shows examples of the trigger's spectrogram $\mathcal{S}(\tau)$ with the frequency domain distribution of trigger's spectrogram $\mathcal{S}(\mathcal{S}(\tau))$ or $\mathcal{A}$. Given that $\mathcal{S}(\mathcal{S}(\tau))$ is generated by applying FFT twice (e.g., STFT and 2D-DFT) on the audio signal, we can refer to this generated frequency-domain distribution as a "$2nd$ order spectrogram" of the inaudible backdoor trigger. The examples demonstrate that the frequency representations of second-order spectrograms can be leveraged to generate inaudible triggers with extremely low sound amplitudes, even if they are visible patterns in the second-order spectrogram.

5 TRIGGER INJECTION METHOD I: DATA POISONING

Synchronization-free Attack via Trigger Rolling and Clipping. Targeting the unpredictable speech content and onsets, we design a trigger rolling and clipping scheme. As illustrated in Figure 6, the audio signals could be mixed with the trigger series at a random position in physical attack scenarios where the trigger is continuously replayed. As the trigger is completely inaudible, the continuous replaying will not alert the user. In common VUI systems, the recorded speech is usually padded to the same length before being fed into the model (e.g., audio padding in Google Speech Command Dataset [2]). To enable synchronization-free attacks, we develop a trigger rolling scheme $roll(\cdot)$ to overcome the unpredictable speech onset and a trigger clipping scheme $clip(\cdot)$ against audio padding. Particularly, we randomly roll the trigger for each audio sample. Then, the trigger is clipped corresponding to the time duration of each sample and mixed with the audio signals during data poisoning. By doing so, the adversary can repeatedly replay trigger τ to launch the synchronization-free attack.

Poisoned Dataset Generation. Our attack separates the audio dataset $\mathcal{X}$ into a clean set $\mathcal{X}_c$ and a poison set $\mathcal{X}_p$ (e.g., $\sim 2\%$ samples). To generate $\mathcal{X}_p$, the adversary injects the rolled and clipped trigger into $\mathcal{X}_p$ and modifies the labels to the adversary-desired label y_t, which can be formulated as:

$$x_i = x_i + clip\big(roll(\tau)\big), y_i = y_t, \tag{6}$$

where x_i denotes the sample in $\mathcal{X}_p$. During the training phase, users conduct the model training following a similar process as described in Section 2.1.

6 TRIGGER INJECTION METHOD II: TRAINING OUTSOURCING

6.1 Joint Optimization for Backdoor Learning

Under the training outsourcing scenario, the adversary aims to train a backdoor model $\mathcal{F}_{\theta'}(\cdot)$ as well as optimize an inaudible trigger $\hat{\tau}$. In this scenario, the adversary can access the training set and adjust training configurations (e.g., loss, epochs) to achieve the optimal performance. The joint optimization of backdoor learning can be formulated as:

$$\mathcal{L}_c = \sum_{i=1}^{N_c} \mathcal{L}\Big(\mathcal{F}_{\theta'}(\mathcal{S}(x_i)), y_i\Big), \mathcal{L}_b = \sum_{j=1}^{N_p} \mathcal{L}\Big(\mathcal{F}_{\theta'}(\mathcal{S}(x_j + \hat{\tau})), y_t\Big), \\ \arg\min_{\theta'} \mathcal{L}_c + \mathcal{L}_b, \tag{7}$$

where $\mathcal{L}$ denotes the loss function used to measure the differences between predicted labels and ground truth labels. The Clean Loss $\mathcal{L}_c$ and Backdoor Loss $\mathcal{L}_b$ are defined as the loss measurements from the clean dataset $\mathcal{X}_c$ and the poisoned dataset $\mathcal{X}_p$, respectively. Compared to data poisoning, we further optimize the trigger pattern $\hat{\tau}$ to enhance the attack's performance and robustness.

6.2 Constraint for Trigger Inaudibility

During backdoor learning, the optimization process may increase the sound magnitude of the trigger, making it less imperceptible. To ensure inaudibility, we design two constraints to cancel the artifacts during optimization as well as restrict the energy below the human audibility curve.

Frequency-domain Artifact Cancellation. To maintain inaudibility, the trigger should induce minimal distortions on the training audio spectrograms. Thus, the differences before and after applying the trigger should be minimized. Particularly, we apply the Mel-Cepstral Distortion (MCD) [27] as the quantification metric. During trigger optimization, we include this term as the Distortion Loss $\mathcal{L}_d$ with the Backdoor Loss $\mathcal{L}_b$ to minimize the distortions, while still maintaining attack performance while the trigger is injected into audio spectrograms. The optimization can be described as:

$$\mathcal{L}_d = \sum_{i=1}^{N_p} MCD\big(\mathcal{S}(x_i), \mathcal{S}(x_i + \hat{\tau})\big), \ \arg\min_{\hat{\tau}} \alpha \cdot \mathcal{L}_d + \mathcal{L}_b, \tag{8}$$

where α denotes the hyper-parameter for balancing Distortion Loss $\mathcal{L}_d$ and Backdoor Loss $\mathcal{L}_b$. We empirically set it as 0.5. During the optimization process, the trigger pattern $\hat{\tau}$ is continuously optimized until the spectrogram distortions induced by trigger injection are minimized.

Inaudibility Enhancement. We further enhance the inaudibility of the trigger by leveraging the human audibility curve [1]. Particularly, we construct a Human Audibility Matrix HC by replicating the normalized human audibility curve to match the dimension of the trigger spectrogram $\mathcal{S}(\hat{\tau})$. We then design the Human Audibility Loss $\mathcal{L}_h$ and the optimization can be described as:

$$\mathcal{L}_h = \big\|\mathcal{S}(\hat{\tau}) \cdot (-HC)\big\|, \ \arg\min_{\hat{\tau}} \alpha \cdot \mathcal{L}_d + \beta \cdot \mathcal{L}_h + \mathcal{L}_b, \tag{9}$$

where β denotes the hyper-parameter used to balance the Distortion Loss $\mathcal{L}_d$, Human Audibility Loss $\mathcal{L}_h$ and Backdoor Loss $\mathcal{L}_b$. We set it as 0.2 empirically. The objective of human audibility optimization is to further diminish the energy of specific frequency components that are sensitive to the human ear, thereby enhancing the trigger's imperceptibility during its replay in practical scenarios.

6.3 Synchronization-free Trigger Optimization

Inspired by the observations in Section 5 and Figure 6, we develop an optimization scheme to address the lack of synchronization between inaudible backdoor triggers and audio samples for realizing effective training outsourcing attacks in practical attack scenarios. Our designed optimization process can be formulated as:

$$\mathcal{L}_s = \sum_{i=1}^{N_p} MSE\Big(\mathcal{Z}_{\theta'}(\mathcal{S}(x_i + clip(\hat{\tau}))), \mathcal{Z}_{\theta'}(\mathcal{S}(x_i + clip(\hat{\tau}')))\Big), \\ \hat{\tau}' = roll(\hat{\tau}), \arg\min_{\hat{\tau}} \alpha \cdot \mathcal{L}_d + \beta \cdot \mathcal{L}_h + \gamma \cdot \mathcal{L}_s + \mathcal{L}_b, \tag{10}$$

where $\mathcal{Z}_{\theta'}(\cdot)$ refers to the logits of the backdoor model $\mathcal{F}_{\theta'}$. $MSE(\cdot,\cdot)$ denotes the Mean Square Error (MSE) and γ is a hyper-parameter used to balance different loss functions, where we empirically set it as 1. Through such optimizations, the robustness of the trigger pattern $\hat{\tau}$ is further improved for practical attacks with unpredictable speech onsets, while simultaneously retaining the trigger's inaudibility and the attack's effectiveness in physical attack scenarios.

7 ROBUST OVER-THE-AIR ATTACK VIA ROOM IMPULSE RESPONSE

In practical attack scenarios, the audio trigger replayed by the loudspeaker will experience distortions caused by reverberation, attenuation, and diffraction as it propagates through the air. These effects can distort the trigger patterns, thereby degrading the attack performance. To address this problem, we employ Room Impulse Response (RIR) [38] to enhance the robustness of the inaudible trigger in trigger generation and backdoor learning. Specifically, RIR models the positions of sound sources, recording devices, and the physical distortions during sound propagation, which helps the model

Table 2: Clean accuracy (CA), signal-to-noise ratio (SNR), and attack success rate (ASR) of our attack on hotword recognition. The poison and injection ratio ($P\%$) are both 5%. (w/o) and (w/) refer to without and with attack.

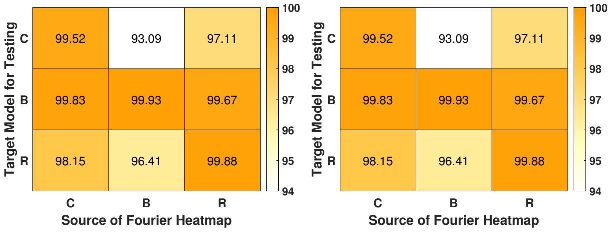
Attack Type	Models	CNN-based Model [2]				Bidirectional RNN Model [15]				ResNet-based Model [43]			
	Metrics	CA(w/o)	CA(w/)	SNR	ASR	CA(w/o)	CA (w/)	SNR	ASR	CA(w/o)	CA(w/)	SNR	ASR
Data Poisoning	GoogleSpeech [2]	73.75%	72.46%	$-22.1dB$	99.67%	82.17%	81.75%	$-10.6dB$	99.92%	91.71%	90.45%	$-24.8dB$	99.92%
	AudioMNIST [13]	97.78%	97.42%	$-14.6dB$	99.93%	98.91%	98.12%	$-13.2dB$	99.19%	99.76%	99.44%	$-19.8dB$	99.64%
Training Outsourcing	GoogleSpeech [2]	73.87%	72.75%	$-72.5dB$	99.33%	82.74%	81.43%	$-38.4dB$	99.03%	91.84%	90.99%	$-81.1dB$	99.82%
	AudioMNIST [13]	96.95%	96.44%	$-58.7dB$	98.94%	98.74%	98.40%	$-30.2dB$	99.12%	99.12%	98.75%	$-64.7dB$	99.34%

to learn trigger patterns that are robust to the channel effects. To realize this, we use a Room Impulse Response (RIR) simulator to simulate a set of distortions in different environments, denoted as $\mathcal{R}$. The trigger injection is then formulated as: $x_i = x_i + clip(roll(\hat{\tau})) \otimes r_i, y_i = y_t, r_i \in \mathcal{R}$, where $\otimes$ refers to the convolution operator. To generate RIRs, we apply the image-based method [11] and randomly configure RIR parameters, including the 3D position of the loudspeaker and microphone, the room dimensions, and the reverberation time, from a uniform distribution of common shoebox rooms [38]. In the data poisoning attack, the frequency-domain triggers after being convoluted with the simulated RIRs are injected into audio samples to generate the poisoned dataset. In the training outsourcing attack, the simulated RIR samples are mixed with the training data, enabling the model to learn the pattern of the frequency-domain trigger.

8 EVALUATION OF DIGITAL ATTACK

Hotword Recognition Models. We evaluate our attack on three types of deep learning models for hotword recognition. *(1) CNN-based Model [2].* The CNN-based model is proposed in the official TensorFlow Tutorial [2]. The model takes audio spectrograms as inputs, which contains two 2D convolutional layers, one 2D max-pooling layer, and two dense layers. *(2) Bidirectional RNN Model [15].* The bidirectional RNN model includes two 2D convolutional layers, two bidirectional Long Short-Term Memory (LSTM) layers, and three dense layers with MFCCs as inputs. *(3) ResNet-based Model [43].* We also build a ResNet-based [23] model for attack evaluation, which leverages a ResNet-based structure as an encoder to extract voice embedding from audio spectrograms and a deep-learning-based classifier for recognition.

Speaker Recognition Models. We evaluate our attack on three speaker recognition models. *(1) DeepSpeaker [28].* DeepSpeaker extracts voice embeddings from audio spectrograms through a ResNet-based extractor and compresses these features as speaker embeddings. Specifically, we use the softmax version to evaluate the attack performance against speaker recognition. *(2) X-vector [40].* X-vector takes MFCCs as inputs and adopts a feature extractor based on Time-delay Neural Network (TDNN) to extract embeddings. *(3) ECAPA-TDNN [17].* Desplanques *et al.* [17] design an Emphasized Channel Attention, Propagation and Aggregation TDNN network (ECAPA-TDNN). Specifically, ECAPA-TDNN takes spectrograms as inputs, uses squeeze-and-excitation blocks to model inter-dependencies of residual blocks and improves the pooling module with frame attention schemes.



(a) Hotword recognition (b) Speaker recognition

Figure 7: Transferability of hotword and speaker recognition with CNN (C), Bi-RNN (B), ResNet (R), DeepSpeaker (D), X-vector (X) and ECAPA-TDNN (E) on ASR.

Evaluation Metrics. *(1) Classification Accuracy (CA).* This metric refers to the percentage of clean samples that can be correctly predicted. The backdoored models should retain high accuracy on clean inputs to pass the validation by the users. Specifically, we build a clean model with the same architecture and compare the accuracy with the backdoor model. *(2) Signal-to-Noise Ratio (SNR).* We quantify the inaudibility of the attack using Signal-to-Noise Ratio (SNR). Specifically, $SNR = 10log_{10}(\frac{P_t}{P_n})$, where P_t and P_n are the power of the trigger and environmental noise. We measure the average SNR within a short period (e.g., first $0.2s$) and show that the trigger's volume is below the ambient noise (i.e., less than $0dB$), indicating that the trigger is inaudible. *(3) Attack Success Rate (ASR).* We utilize ASR to measure the ratio of poisoned samples that are classified as the adversary-desired class. During experiments, we take turns setting each label as the target label and summarize the average ASR.

8.1 Attack via Data Poisoning

Attack Setup. For attacking hotword recognition, we use Google Speech Command Dataset [2] and AudioMNIST [13] with $15,076$ and $30,000$ audio samples for 30- and 10-hotword recognition. Each sample lasts for 1 second and the sampling rates are 16kHz and 48kHz, respectively. For attacking speaker recognition, we utilize a subset from the VCTK corpus [14] and AudioMNIST [13] with $8,526$ and $30,000$

Table 3: Clean accuracy (CA), signal-to-noise ratio (SNR), and attack success rate (ASR) of our attack on speaker recognition. The poison and injection ratio ($P\%$) are both 5%. (w/o) and (w) refer to without and with attack.

Attack Type	**Models**	DeepSpeaker [28]				X-vector [40]				ECAPA-TDNN [17]			
	Metrics	CA(w/o)	CA(w)	SNR	ASR	CA(w/o)	CA (w)	SNR	ASR	CA(w/o)	CA(w)	SNR	ASR
Data Poisoning	VCTK [14]	97.02%	96.17%	$-30.1dB$	99.82%	88.96%	88.01%	$-12.1dB$	98.22%	95.69%	94.21%	$-24.6dB$	99.52%
	AudioMNIST [13]	94.75%	93.98%	$-19.7dB$	99.77%	83.06%	82.17%	$-17.5dB$	99.15%	96.65%	95.39%	$-26.4dB$	99.91%
Training Outsourcing	VCTK [14]	97.14%	95.79%	$-95.5dB$	99.94%	89.77%	88.05%	$-47.7dB$	98.24%	95.65%	93.97%	$-56.9dB$	99.21%
	AudioMNIST [13]	94.17%	93.19%	$-77.6dB$	99.91%	82.75%	82.02%	$-28.9dB$	98.94%	96.04%	94.98%	$-37.1dB$	98.77%

samples for 50- and 60-speaker recognition. Each sample lasts for 1 second and the sampling rates are both 48kHz. We split the datasets into training and testing sets with a ratio of 8:2 and inject our designed inaudible trigger into a subset (e.g.,~ 5%) of training samples.

Results of Attacking Hotword Recognition. The results of our data poisoning attack against hotword recognition task are illustrated in Table 2. In total, the impact of our proposed inaudible attack on model's CA is less than 1.29%, which indicates that the users will not notice the attack by comparing the validation accuracy with clean models. Moreover, our attack achieves less than $-10.6dB$ on SNR measurements, which demonstrates the inaudibility of our attack given lower signal energy compared to environmental noise. The ASRs reach more than 99.19%. The results indicate the effectiveness of our inaudible backdoor attack via data poisoning on the hotword recognition task.

Results of Attacking Speaker Recognition. We show the results of our data poisoning attack against speaker recognition models in Table 3. The drop of model's CA induced by our attack is less than 1.48%, which demonstrates the stealthiness of our proposed attack. Furthermore, the SNR measurements are less than $-12.1dB$, which indicates the inaudibility of our designed attack. For ASR measurements, our attack achieves more than 98.22%. The results demonstrate our inaudible attack is also effective against deep learning models for speaker recognition.

Transferability Study of Fourier Heatmap. To demonstrate our attack's generality, we evaluate the transferability of the Fourier Heatmap by initializing the trigger using the Fourier Heatmap of one model and testing on another different model. The ASR measurements of cross-model testing on Google Speech Command Dataset [2] and VCTK Corpus [14] are shown in Figure 7. For hotword recognition, the lowest ASR achieves more than 93.09%, which demonstrates the generality of our data poisoning attack on hotword recognition. For speaker recognition, the lowest ASR is more than 82.19%, where we use X-vector for generating Fourier Heatmap and DeepSpeaker for testing. The accuracy drop can be attributed to different model structures (e.g., residual structure for DeepSpeaker and TDNN for X-vector). Nevertheless, high ASRs demonstrate the generality of our inaudible backdoor attack design on the data poisoning attack.

8.2 Attack via Training Outsourcing

Attack Setup. We leverage the same dataset in Section 8.1. During the training outsourcing attack, the adversaries optimize the trigger pattern along with the backdoor model and inject the optimized trigger into audio samples for testing.

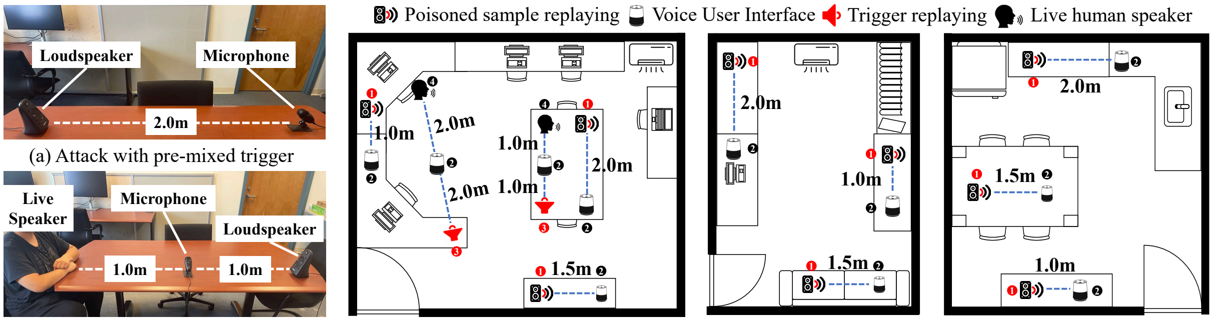
Results of Attacking Hotword Recognition. The results are shown in Table 2. Particularly, our training outsourcing attack only induces degradation on model's CA with less than 1.31%. For the attack inaudibility, the SNRs are less than $-30.2dB$ and much lower than the SNRs of data poisoning attack with $-10.6dB$, which demonstrates that a stronger attack with less trigger perceptibility can be realized through outsource training. The ASR of our attack via training outsourcing can achieve at least 99.03% and 98.94% for Google Speech Command Dataset [2] and AudioMNIST [13], which indicates the effectiveness of our training outsourcing attack with frequency-domain inaudible triggers.

Results of Attacking Speaker Recognition. The results are illustrated in Table 3. For VCTK Corpus [14], our attack achieves 98.24% on ASR with a drop of 1.72% on CA. For AudioMNIST [13], the ASR achieves 98.77% with less than 1.06% on CA degradation. The results demonstrate the effectiveness and stealthiness of our training outsourcing attack. Meanwhile, the SNR measurements are less than $-28.9dB$ and $-37.1dB$ compared to $-17.5dB$ and $-12.1dB$ under data poisoning attack, which further proves that our attack via training outsourcing is completely inaudible with stronger attack effects compared to the data poisoning attack.

9 EVALUATION OF PHYSICAL ATTACK

Room Settings. We conduct experiments in three different in-door environments, including two offices and an apartment as shown in Figure 8. The size of Office 1 is $8.5m \times 7.6m$ and the sound pressure level (SPL) of noise is $40.8dB$, which is mainly generated by multiple desktops and an air conditioner. For Office 2, the size is $7.6m \times 3.2m$ with a noise SPL of $39.2dB$ generated from a desktop and an air conditioner. For apartment 1, the size is $6.2m \times 4.4m$ with a noise SPL of $37.4dB$, where the main noise source is the refrigerator.

RIR Generation. To simulate the over-the-air propagation of audio signals and generate robust poisoned samples and backdoor triggers for over-the-air attack scenarios, we



(a) Attack with pre-mixed trigger

(b) Attack on live speech

(c) Office 1

(d) Office 2

(e) Apartment 1

Figure 8: Room layouts of physical attacks with pre-mixed trigger and attacks on live speech.

apply the RIR simulator [38] as illustrated in Section 7. Specifically, we generate a large RIR dataset with the same number of audio samples in the training set. These RIRs are incorporated into the data poisoning or training outsourcing process to improve the robustness against over-the-air distortions.

9.1 Attack via Data Poisoning

Attack Setup. For the attack with pre-mixed triggers, the adversary injects the trigger into audio samples and then replay the them via loudspeaker. Specifically, we randomly select 200 samples from the Google Speech Command Dataset [2] to inject the trigger and replay them via a Logitech Z623 loudspeaker with 60dB SPL (similar to human conversation) and recorded by an Insignia NS-CBM19 USB microphone for simulating VUIs. Under three rooms, we set different distances (e.g., 1.0m, 1.5m and 2.0m) between the loudspeaker and microphone, as shown in Figure 8. For attacking live speech, we recruit three participants and instruct them to read hotwords from Google Speech Command Dataset [2] for 10 repeats. Meanwhile, we use the Logitech Z623 loudspeaker to replay the inaudible trigger. Experiments are conducted in Office 1 with distances of 1.0m and 2.0m between the participant and the loudspeaker. The data collection has been approved by our university's Institutional Review Board (IRB).

Results of Attack with Pre-mixed Triggers. We evaluate our attack on the CNN-based model and the results are shown in Figure 9. Specifically, the ASRs of our attack without RIR simulation only achieve 43.00%, 45.00%, 44.50% under the distances of 1.0m, 1.5m, 2.0m in Office 1. For Office 2 and Apartment 1, the ASRs reach 44.50%, 41.00%, 42.50% and 49.50%, 48.00%, 44.00%. After involving RIR simulation, the ASRs achieve 87.50%, 88.50%, 88.50% for Office 1, 85.00%, 82.00%, 80.50% for Office 2 and 90.50%, 90.50%, 86.50% for Apartment 1. The results demonstrate the effectiveness of our designed RIR simulator and our data poisoning attack in physical scenarios with pre-mixed triggers.

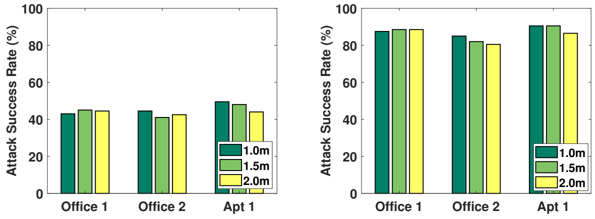
Results of Attack on Live Speech. We show the results of our attack against live speech in Figure 10. Without RIR simulation, the ASRs under the distances of 1.0m and 2.0m are 36.00%, 35.00% for user 1, 40.50%, 37.00% for user 2 and 38.50%, 41.00% for user 3. After simulating RIR, the ASRs reach 66.50%, 68.00% for user 1, 70.50%, 72.00% for user 2 and 66.50%, 65.50% for user 3. The high ASRs prove that the attack performance can be effectively improved with our RIR simulation and our proposed attack can be successfully deployed against live speech in physical attack scenarios.

9.2 Attack via Training Outsourcing

Attack Setup. We follow the same experimental setup proposed in Section 9.1. During the attack, we optimize the trigger pattern with the parameters of the backdoored model as described in Section 6. After generating the optimized trigger, we inject the trigger into replayed samples or replay the optimized trigger through the loudspeaker.

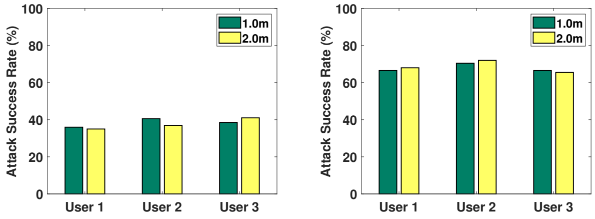
Results of Attack with Pre-mixed Triggers. The results with the CNN-based model are illustrated in Figure 11. Without RIR simulation, the ASRs achieve 71.50%, 63.00% and 65.50% at distances of 1.0m, 1.5m and 2.0m in Office 1. In Office 2 and Apartment 1, the ASRs are 69.50%, 68.50%, 65.50%, and 75.50%, 72.00%, 72.50% at different distances. After involving RIR simulation, the ASRs are significantly improved with 89.50%, 93.00%, 91.50% in Office 1, 86.00%, 86.00%, 84.00% in Office 2 and 90.00%, 89.50%, 87.00% in Apartment 1. High ASRs under different environments demonstrate the effectiveness of the RIR simulator. Compared with data poisoning attacks, training outsourcing attacks achieve higher ASRs, which indicates stronger attacks can be realized via outsource training in physical attack scenarios.

Results of Attack on Live Speech. The results against live speech are shown in Figure 12. Without RIR simulation, the ASRs of the training outsourcing attack against live speech at distances of 1.0m and 2.0m are 42.00%, 44.00% for user 1, 43.50%, 41.00% for user 2 and 40.50%, 43.00% for user 3. After involving RIR simulation, the ASRs increase to 76.00%, 74.50% for user 1, 73.50%, 73.00% for user 2 and 74.50%, 75.00% for user 3. Such high ASRs demonstrate the effectiveness of our proposed training outsourcing attack against live speech in practical attack scenarios.



(a) Without RIR simulation (b) With RIR simulation

Figure 9: ASR of physical data poisoning attack against the CNN-based model with pre-mixed triggers.



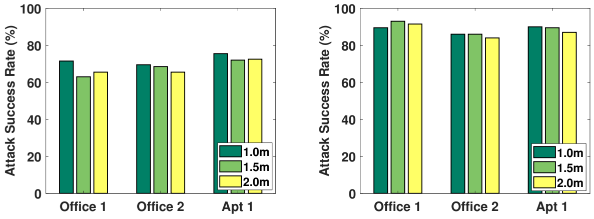
(a) Without RIR simulation (b) With RIR simulation

Figure 10: ASR of physical data poisoning attack against the CNN-based model with live triggers.

9.3 Attack Under Noisy Environments

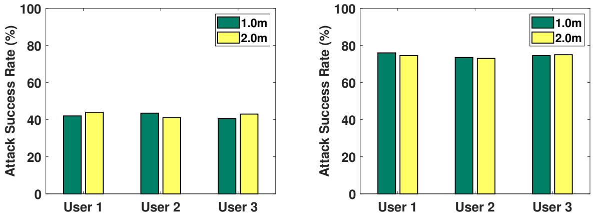
Attack Setup. To evaluate the noise resilience of our attack with pre-mixed triggers, we employ a JBL GO3 speaker to replay Gaussian white noise with 45dB and 55dB in three rooms, which is placed 1.0m, 1.5m and 2.0m from the microphone (i.e., same with the loudspeaker). To validate the noise resilience of our attack on live speech, we place the same JBL GO3 loudspeaker for noise replaying 1.0m and 2.0m away from the loudspeaker (i.e., close to the loudspeaker for trigger replaying). The experiments are conducted in Office 1, where the same three participants are involved to read the hotwords from Google Speech Command Dataset [2] for 10 repeats. Note that we evaluate the noise resilience of our training outsourcing attack since the trigger has lower magnitudes compared with our data poisoning attack.

Results of Attack with Pre-mixed Triggers. The results with RIR simulation are shown in Figure 13. With a Gaussian white noise of 45dB, the ASRs achieve 85.50%, 87.00% and 83.50% at the distances of 1.0m, 1.5m and 2.0m between the two loudspeakers and the microphone in Office 1. For Office 2 and Apartment 1. the ASRs are 82.50%, 83.00%, 83.00% and 86.50%, 82.00%, 82.00% under three distances. When replaying Gaussian noise of 55dB, the ASRs reach 84.50%, 83.50%, and 83.50% under three distances in Office 1. For Office 2 and Apartment 1, the ASRs reach 82.50%, 82.50%, 80.00%, and 84.50%, 82.50%, 79.00%, respectively. Compared with the ASRs without noise replaying, the ASRs only experience a drop of 9.50%. The results demonstrate that our designed inaudible trigger cannot be invalidated by environmental noise, making it more robust in practical scenarios.



(a) Without RIR simulation (b) With RIR simulation

Figure 11: ASR of physical training outsourcing attack against the CNN-based model with pre-mixed triggers.



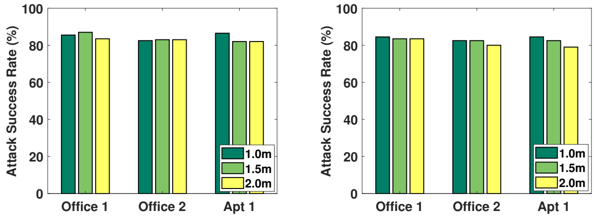
(a) Without RIR simulation (b) With RIR simulation

Figure 12: ASR of physical training outsourcing attack against the CNN-based model with live triggers.

Results of Attack on Live Speech. The noise resilience performance of our attack on live speech is shown in Figure 14. With noise replaying of 45dB, the ASRs of the training outsourcing attack at distances of 1.0m and 2.0m are 69.50%, 68.00% for user 1, 69.50%, 67.00% for user 2 and 66.00%, 68.00% for user 3. With noise replaying of 55dB, the ASRs have slight drops, with 66.50%, 67.00% for user 1, 64.50%, 65.00% for user 2 and 63.00%, 65.00% for user 3. High ASRs under noisy environments demonstrate that our proposed attack has good noise resilience performance while attacking live speech and can be effectively deployed under real-world scenarios.

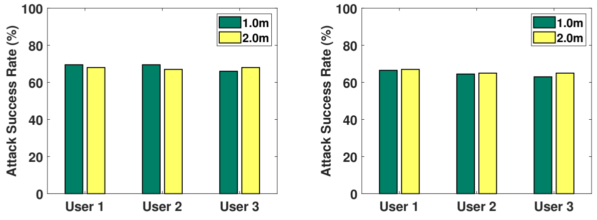
10 EVALUATION AGAINST DEFENSE

Learning-based Defenses. *(1) Neural Cleanse [44].* Neural Cleanse leverages a reverse-engineering-based approach to reconstruct the trigger pattern. Specifically, it utilizes the Anomaly Index as a threshold, which is computed from the average L1-norm changes for the model to output different predictions. If the Anomaly Index is larger than 2.0, Neural Cleanse detects the backdoor triggers and leverages gradient reversing to infer their patterns. We apply Neural Cleanse against a CNN-based [2] model with Google Speech Command Dataset [2]. For the data poisoning and training outsourcing attack, the Anomaly Indices are 1.3978 and 1.5933, which indicates that our attack can bypass Neural Cleanse. *(2) STRIP [22].* STRong Intentional Perturbation (STRIP) first



(a) Noise Replaying of 45$dB $ (b) Noise Replaying of 55dB

Figure 13: ASR of 45dB and 55dB noise replaying against CNN-based model with pre-mixed triggers.

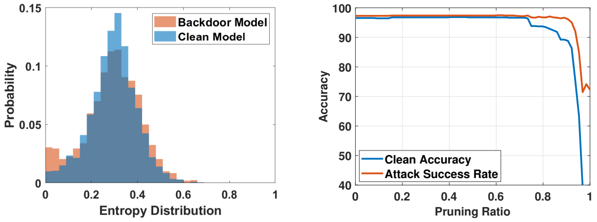


(a) Noise Replaying of 45dB (b) Noise Replaying of 55dB

Figure 14: ASR of 45dB and 55dB noise replaying against CNN-based model with live triggers.

injects perturbations randomly selected from a set of benign samples into model inputs. If the predictions of perturbed inputs concentrate on one specific class with high entropy, the tested model is considered as a backdoor model. An example of STRIP is shown in Figure 15(a). Similar entropy distributions demonstrate that our attack can successfully bypass STRIP. *(3) Fine-Pruning [31].* Fine-Pruning assumes that the backdoor trigger will cause anomaly activation of specific neurons. Particularly, it calculates the activation of each neuron in the last convolutional layer of the model, and prunes the neurons with the descending order of the activation values. An example of Fine-Pruning is shown in Figure 15(b), where we cannot find the pruning ratio with low ASR and negligible CA drop. The results demonstrate that our attack can bypass Fine-Pruning.

Audio-processing-based Defenses. *(1) Signal Quantization.* Signal quantization, which denotes modifying the bit depth of audio signals, has been employed for defending audio backdoor attacks [16, 29, 47]. The signal quantization results of our training outsourcing attack using the CNN-based model [2] on AudioMNIST [13] are summarized in Table 4 with a slight drop of 0.87%, which demonstrates that our attack can bypass signal quantization. *(2) Median Filter.* As a filtering technique for noise removal, the median filter has been applied to defend audio backdoor attacks [16, 29, 47]. We show the attack performance of the CNN-based model [2] on AudioMNIST [13] after applying different sizes of median filter in Table 4, where the ASR can still retain more than 96.75%. The results demonstrate that our attack can successfully bypass the defense based on the median filter.



(a) Entropy distribution of STRIP (b) Performance of Fine-Pruning

Figure 15: Evaluation of STRIP on ResNet-based model and Fine-Pruning on DeepSpeaker.

Table 4: Clean accuracy (CA) and attack success rate (ASR) of signal quantization and median filter.

Metrics	Quantization		Size of Median Filter			
	16 *bits*	8 *bits*	3	5	7	9
CA	96.95%	96.14%	95.72%	97.07%	96.88%	96.88%
ASR	98.94%	98.07%	96.75%	98.94%	98.94%	98.72%

11 DISCUSSION

Attack Inaudibility Analysis. We analyze the trigger's inaudibility by comparing the SNRs with other triggers in existing works [39, 48]. Particularly, Zhai *et al.* [48] leverage a single-tone signal with the volume of $-45dB \sim -20dB$ (compared to the highest speech volume) as the backdoor trigger. Shi *et al.* [39] use environmental sounds (e.g., bird chirp) from ESC-50 Environmental Sound Classification Dataset [34] as the trigger. For our inaudible backdoor attack, the trigger's SNR is $-3.54dB$ for the data poisoning attack and $-7.71dB$ for the training outsourcing attack, which are smaller than existing works with $19.12dB$ and $88.75dB$. The SNR measurements demonstrate that our designed backdoor attack is completely inaudible in real-world attack scenarios.

Attack Generality Across Audio Datasets. To further examine our attack generality across different audio datasets, we conduct experiments by pre-training a frequency-domain trigger using one dataset and applying it to a different dataset. Specifically, we generate a trigger using the Google Speech Command dataset [2] and inject the trigger into AudioMNIST [13] for evaluating its effectiveness. With CNN-based model, the attack can achieve more than 84.75% ASR. The rationale is that common acoustic features (e.g., speech frequency ranges, harmonics of speech) are shared in different speech datasets so that the trigger effective in one speech dataset can also be applied to another dataset.

Attack Augmentation with Ultrasound Frequency. Existing works [36, 47, 49] have demonstrated that speech signals modulated in ultrasonic sounds can be received by commodity microphones. These ultrasound-based attacks are inaudible but restricted to short distance and specialized

Table 5: Differences between our inaudible backdoor attack and the existing audio backdoor attacks. "-" refers to their focus on digital attack scenarios.

Attacks	Attack Scenarios	Playback Device	Audibility
Zhai et al. [48]	Poisoning	-	Audible
DriNet [45]	Outsourcing	-	Audible
Shi et al. [39]	Outsourcing	Commercial Speaker	Audible
VENOMAVE [10]	Outsourcing & Poisoning	Commercial Speaker	Audible
UltraBD [47]	Outsourcing	Ultrasonic Speaker	Inaudible
Ours	**Outsourcing & Poisoning**	**Commercial Speaker**	**Inaudible**

playback devices (e.g., ultrasonic loudspeaker). To improve attack effectiveness and imperceptibility, a potential solution is to apply ultrasonic frequencies in our trigger design, which combines the advantages of our attack (e.g., long range) and ultrasound attacks (e.g., free from optimization). We will consider these improvements in our future works.

Potential Defense Strategies. We summarize two potential defense strategies against our attack. *(1) Ensemble Prediction.* A potential defense is to exploit predictions from multiple models trained on different datasets with the same labels (e.g., the same digits or words). Given the difficulties for adversaries to poison multiple models, the models trained with clean datasets will make correct predictions on the poisoned samples. A majority vote of multiple models will provide accurate predictions even if several backdoor models exist. *(2) Acoustic Feature Clustering.* The users can apply clustering approaches (e.g., K-Means, DBSCAN) on the audio samples based on extracted acoustic features (e.g., MFCCs). The clean samples should be clustered together, while those samples with modified labels should deviate. This defense will allow users to detect and remove the poisoned samples from the dataset before model training.

12 RELATED WORKS

Audio-domain Backdoor Attacks. Unlike image-domain attacks with different tasks (e.g., warping [32], invisible [18, 19], dynamic [37]), there are only a few studies in the audio domain. Zhai et al. [48] use clustering to generate poisoned audio against speaker verification models. DriNet [45] generates dynamic trigger patterns against speech recognition systems. However, these works focus on digital attack scenarios instead of practical settings. Shi et al. [39] design position-independent triggers that are effective while injected at any temporal position of the streaming audio. VENOMAVE [10] proposes a poisoning attack against speech recognition in over-the-air scenarios. However, these triggers are designed as audible (e.g., environmental sound [39], spectrogram patch [10]), which can be noticed by the users. Moreover, these attacks directly insert triggers into audio signals, making them vulnerable to existing backdoor defense techniques, such as Neural Cleanse [44], which expose the attack by reverse-engineering the trigger pattern. While UltraBD [47] realizes an inaudible attack with ultrasound as triggers, it requires dedicated devices (e.g., ultrasonic speaker) for replaying triggers. In contrast, our designed trigger can be replayed with commodity devices (e.g., commercial loudspeakers). The comparisons of our attack with the existing audio backdoor attacks are shown in Table 5.

Synchronization-free Audio Adversarial Attacks. Existing works [16, 30] have explored realizing synchronization-free audio adversarial attacks. However, the sound magnitude of these attacks needs to be sufficiently large (audible) for the effectiveness. As speech recognition models are normally trained to recognize audible sounds, the perturbation used to launch such adversarial attacks is audible, thus they are noticeable to users. Compared with these works, we design the trigger to have energy below the noise floor (e.g., background and hardware noises) and involve it into model training to make the attack inaudible to humans.

Inaudible Attacks. Roy et al. [35] show that MEMS microphones on mobile devices can capture high-frequency sounds (e.g., $\geq 20kHz$), allowing adversaries to inject inaudible commands. Existing works also explore inaudible triggers for backdoor attacks. For example, Koffas et al. [25] utilize ultrasonic pulses as trigger patterns. However, these triggers cannot pass through low-pass filters, thus cannot be deployed in physical attack scenarios. Moreover, ultrasound-based attacks often encounter substantial attenuation [9], resulting in reduced effective attack distances. Sugawara et al. [41] propose a laser-based attack against microphones, but it requires the line-of-sight to the target device.

13 CONCLUSION

In this work, we present an audio backdoor attack that injects inaudible triggers in the frequency domain of audio spectrograms. We formulate two trigger injection methods, data poisoning and training outsourcing. To generate inaudible triggers, our attack system first constructs an initial trigger by identifying critical frequency components of audio spectrograms in a dataset. By altering the trigger structure during backdoor learning, our attack forces the compromised model to detect the trigger in a synchronization-free manner. We further enhance attack imperceptibility and robustness under practical scenarios through joint optimizations. Comprehensive experiments involving six deep learning models confirm the effectiveness of our attack under digital and physical settings. We further verify that our attack can successfully circumvent representative backdoor defense methods.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation Grants CNS2120276, CNS2120396, CNS2145389, CCF2211163, IIS2311596, IIS2311597, CMMI2152908.

REFERENCES

[1] Features of equal loudness curves. http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/earcrv.html. 2023.

[2] Simple audio recognition: Recognizing keywords: Tensorflow core. https://www.tensorflow.org/tutorials/audio/simple_audio. 2020.

[3] Amazon alexa. https://alexa.amazon.com/, 2023.

[4] Amazon machine learning and artificial intelligence. https://aws.amazon.com/machine-learning/, 2023.

[5] The evolution of in-car voice control leads to win-win for all. https://www.kardome.com/blog-posts/evolution-car-voice-control, 2023.

[6] Microsoft azure machine learning studio. https://studio.azureml.net/, 2023.

[7] Mozilla common voice dataset. https://commonvoice.mozilla.org/en/datasets, 2023.

[8] Siri-apple. https://www.apple.com/siri/, 2023.

[9] Stokes's law of sound attenuation. https://en.wikipedia.org/wiki/Stokes%27s_law_of_sound_attenuation, 2023.

[10] Hojjat Aghakhani, Lea Schönherr, Thorsten Eisenhofer, Dorothea Kolossa, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. Venomave: Targeted poisoning against speech recognition. *arXiv preprint arXiv:2010.10682*, 2020.

[11] Jont Allen and David Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65:943–950, 04 1979.

[12] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin, 2015.

[13] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[14] Kirsten MacDonald Christophe Veaux, Junichi Yamagishi. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit, 2019.

[15] Douglas Coimbra de Andrade, Sabato Leo, Martin Loesener Da Silva Viana, and Christoph Bernkopf. A neural attention model for speech command recognition, 2018.

[16] Jiangyi Deng, Yanjiao Chen, and Wenyuan Xu. Fencesitter: Black-box, content-agnostic, and synchronization-free enrollment-phase attacks on speaker recognition systems. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 755–767, New York, NY, USA, 2022. Association for Computing Machinery.

[17] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*. ISCA, oct 2020.

[18] Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. *Advances in Neural Information Processing Systems*, 34, 2021.

[19] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11966–11976, 2021.

[20] Chaz Firestone. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571, 2020.

[21] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2349–2364, 2022.

[22] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 113–125, New York, NY, USA, 2019. Association for Computing Machinery.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Yelim Kim, Mohi Reza, Joanna McGrenere, and Dongwook Yoon. Designers characterize naturalness in voice user interfaces: Their goals, practices, and challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[25] Stefanos Koffas, Jing Xu, Mauro Conti, and Stjepan Picek. Can you hear it? backdoor attacks via ultrasonic triggers. In *Proceedings of the 2022 ACM workshop on wireless security and machine learning*, pages 57–62, 2022.

[26] J. E. (Hans). Korteling, G. C. van de Boer-Visschedijk, R. A. M. Blankendaal, R. C. Boonekamp, and A. R. Eikelboom. Human-versus artificial intelligence. *Frontiers in Artificial Intelligence*, 4, 2021.

[27] R. Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1, pages 125–128 vol.1, 1993.

[28] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system, 2017.

[29] Xinfeng Li, Junning Ze, Chen Yan, Yushi Cheng, Xiaoyu Ji, and Wenyuan Xu. Enrollment-stage backdoor attacks on speaker recognition systems via adversarial ultrasound, 2023.

[30] Zhuohang Li, Yi Wu, Jian Liu, Yingying Chen, and Bo Yuan. Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1121–1134, New York, NY, USA, 2020. Association for Computing Machinery.

[31] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks, 2018.

[32] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

[33] Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[34] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[35] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 2–14, 2017.

[36] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible voice commands: The Long-Range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 547–560, Renton, WA, April 2018. USENIX Association.

[37] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models, 2022.

[38] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems. In *Annual Computer Security Applications Conference*, ACSAC '20, page 843–855, New York, NY, USA, 2020. Association for Computing Machinery.

[39] Cong Shi, Tianfang Zhang, Zhuohang Li, Huy Phan, Tianming Zhao, Yan Wang, Jian Liu, Bo Yuan, and Yingying Chen. Audio-domain position-independent backdoor attack via unnoticeable triggers. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, MobiCom '22, page 583–595, New York, NY, USA, 2022. Association for Computing Machinery.

[40] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018.

[41] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. Light commands: laser-based audio injection attacks on voice-controllable systems. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 2631–2648, 2020.

[42] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.

[43] Roman Vygon and Nikolay Mikhaylovskiy. Learning efficient representations for keyword spotting with triplet loss. In *Speech and Computer*, pages 773–785. Springer International Publishing, 2021.

[44] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019.

[45] Jianbin Ye, Xiaoyuan Liu, Zheng You, Guowei Li, and Bo Liu. Drinet: dynamic backdoor attack against automatic speech recognization models. *Applied Sciences*, 12(12):5786, 2022.

[46] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[47] Junning Ze, Xinfeng Li, Yushi Cheng, Xiaoyu Ji, and Wenyuan Xu. Ultrabd: Backdoor attack against automatic speaker verification systems via adversarial ultrasound. In *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 193–200, 2023.

[48] Tongqing Zhai, Yiming Li, Ziqi Zhang, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Backdoor attack against speaker verification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2560–2564, 2021.

[49] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 103–117, New York, NY, USA, 2017. Association for Computing Machinery.