VoShield: Voice Liveness Detection with Sound Field Dynamics

Qiang Yang, Kaiyan Cui, Yuanqing Zheng
Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
{csqyang, cskcui, csyqzheng}@comp.polyu.edu.hk

***Abstract*—Voice assistants are widely integrated into a variety of smart devices, enabling users to easily complete daily tasks and even critical operations like online transactions with voice commands. Thus, once attackers replay a secretly-recorded voice command by loudspeakers to compromise users' voice assistants, this operation will cause serious consequences, such as information leakage and property loss. Unfortunately, most voice liveness detection approaches against replay attacks mainly rely on detecting lip motions or subtle physiological features in speech, which are limited within a very short range. In this paper, we propose VoShield to check whether a voice command is from a genuine user or a loudspeaker imposter. VoShield measures sound field dynamics, a feature that changes fast as the human mouths dynamically open and close. In contrast, it would remain rather stable for loudspeakers due to the fixed size. This feature enables VoShield to largely extend the working distance and remain resilient to user locations. Besides, sound field dynamics are extracted from the difference between multiple microphone channels, making this feature robust to voice volume. To evaluate VoShield, we conducted comprehensive experiments with various settings in different working scenarios. The results show that VoShield can achieve a detection accuracy of 98.2% and an Equal Error Rate of 2.0%, which serves as a promising complement to current voice authentication systems for smart devices.**

***Index Terms*—Voice Assistant, Liveness Detection, Microphone Array, Replay Attack**

I. Introduction

Background. Voice assistants (*e.g.*, Google Now, Alexa, Siri, *etc*.) are becoming increasingly popular and facilitate user interaction with smart devices these days. Voice interaction allows users to quickly complete daily tasks in a hands-free way, such as controlling home appliances and ordering food online. Recently, voice assistants have been in connection with a variety of smart gadgets, which serve as an entrance into the smart home network. As a result, voice assistants have been empowered to perform more sophisticated and critical functions, such as online transactions, home surveillance, and even door unlocking [1].

Motivation. To protect voice assistants, they typically use voiceprint-based automatic speaker verification (ASV) [2], [3] to authenticate legitimate users. Voice commands, however, can be secretly recorded by others. As a matter of fact, attackers can easily obtain user voice clips from online meetings, phone calls, live presentations, or video recordings. Recent advances in deep-fake technologies can also synthesize and reproduce voice commands at will. A study [4] demonstrates that ASVs are vulnerable to replay attacks because replayed voice commands originate from a legitimate user. Moreover, it is reported that many smart home appliances are less protected and suffer from security flaws [5], which make it possible for attackers to remotely play malicious voice commands over the Internet by hijacking the smart devices. As such, attackers can intentionally replay or inject unauthorized commands into popular music or YouTube videos to attack users' voice assistants, as illustrated in Fig. 1. Therefore, we urgently need to protect voice assistants against replay attacks to avoid serious consequences such as privacy leakage, property loss, and even worse.

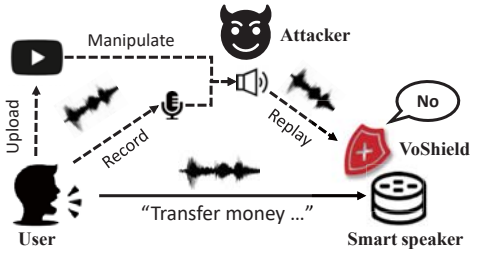


Fig. 1: Application scenario of VoShield. Attackers can steal voice clips from a sneak recording or public videos to employ remote replay attacks. VoShield is designed to protect voice assistants by blocking such loudspeaker-played attacks.

Limitation of existing solutions. To defend against such attacks, existing works enhance ASV systems with liveness detection. If a voice command passes the ASV, it has to be examined in terms of liveness. Specifically, as replay attacks are played by loudspeakers, we can distinguish such attacks by checking whether a voice command originates from a real human being or a loudspeaker. Prior arts build side channels to detect the voice liveness with additional devices, such as motion sensors [6]–[8], Wi-Fi radios [9]–[12], earbuds [13], [14]. However, these works require extra hardware and limit application scenarios. Some recent works emit inaudible acoustic signals to sense users' movement when speaking (*e.g.*, lip motion or breath) and hereby detect the voice liveness [15]–[18]. Although effective, high-frequency acoustic signals can be audible and disruptive to babies and pets. To address these practical challenges, many researchers attempt to passively detect vital clues within the received voice commands only [19]–[21]. However, they require users to hold the devices with fixed gestures at very close locations to capture the subtle physiological sounds. Therefore, these methods are not capable of interacting with distant devices, such as smart

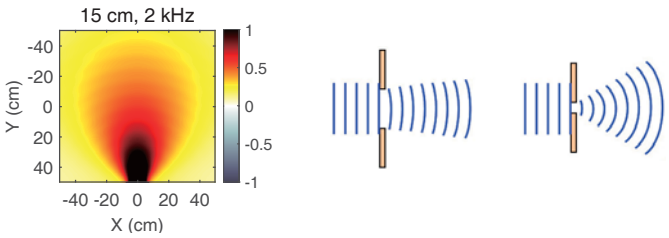
speakers and smart lamps.

Our insight. This paper aims to develop a passive acoustic-based liveness detection method without restricting users to certain fixed gestures or positions. The high-level idea of our system, VoShield, is simple. We observe that the intrinsic difference between humans and loudspeakers is aperture size variation. Specifically, humans need to dynamically open and shut their mouths to speak voice commands, while loudspeakers always keep a fixed aperture size. Intuitively, the time-varying mouth aperture of humans leads to a more dynamic sound field than loudspeakers. By examining the dynamic level of sound fields, we can distinguish the voice liveness, *i.e.*, whether a voice command is from a real user's mouth or a loudspeaker, to combat replay attacks.

Challenges. However, implementing our idea involves a series of challenges. The first is how to characterize the dynamic level of the sound field. Traditionally, people use a large number of microphones distributed around a room to measure the sound pressure and then interpolate them into a sound field, which is impossible for the small microphone array used in daily smart devices. Meanwhile, The sound field fluctuation depends on not only the size variation of sound sources but also other factors (*e.g.*, the voice content and volume). Secondly, given there are typically several microphones in an array, cooperating different microphone channels to facilitate the measurement, needs to be handled properly. Finally, based on the feature we measured, designing an effective approach to discriminate between humans and loudspeakers also remains a challenge.

Our solution. In this paper, instead of directly measuring the sound field, we propose Sound Field Dynamics (SFD), a new feature that indirectly characterizes the *dynamic level* of sound fields, which captures the intrinsic difference between the sound fields generated by loudspeakers and real humans. SFD is based on the temporal fluctuation of the energy ratio between different microphones. This inter-microphone ratio eliminates the effect of the absolute sound intensity, so the SFD is independent of the sound volume. Moreover, the SFD is essentially determined by the physical aperture size variations of a sound source, hence resistant to source locations. To make full use of all microphones in an array, we present a multi-channel fusion approach to facilitate SFD measurement. Based on the extracted SFD features, we design a deep learning model with a self-attention mechanism to further fuse multiple channels and differentiate humans and loudspeakers. The key contributions of this paper are summarized as follows:

- We propose VoShield to protect voice assistants against replay attacks at room scale without relying on extra hardware.
- We introduce the notion of sound field dynamics, an effective feature that indicates voice liveness and hereby distinguishes humans and loudspeakers.
- VoShield is implemented on commercial microphone arrays, and evaluation in various settings demonstrates its applicability and effectiveness.



(a) Sound field illustration. (b) Diffraction effects.

Fig. 2: Sound field and diffraction effect.

We want to point out that VoShield is a complement, not a replacement, to the existing voice authentication solutions. The security of voice commands cannot be overemphasized. To protect voice assistants, VoShield will not work alone but will cooperate with other voice authentication approaches to provide a more reliable protection service.

The rest of the paper begins with the explanation of sound field dynamics (Sec. II), followed by system design (Sec. III), implementation (Sec. IV), and evaluation (Sec. V). We summarize the related work in Sec. VI, discuss some limitations and future directions in Sec. VII, and finally conclude this paper.

II. Understanding Sound Field Dynamics

In this section, we first introduce the basic concept of sound field and sound directivity, then explain the rationale behind VoShield: the varied source size leads to the variant sound directivity and further results in the sound field dynamics.

A. Sound Fields and Directivity

The sound field describes the energy diffusion of an acoustic source over a space. Fig. 2(a) illustrates the sound field of a sound source (15 cm aperture, emitting 2 kHz sine tone) with the k-Wave simulation [22]. As the heat map shows, there are different sound power at different positions, forming the sound directivity. This is because its different parts vibrate simultaneously, and the generated sound waves will constructively or destructively interfere with each other at different locations [20]. Additionally, sounds have the *diffraction effect*, which depends on the physical aperture size of the sound source relative to the wavelength of the sound wave [23]. As shown in Fig. 2(b), with the same wavelength (*i.e.*, frequency), the larger aperture leads to a weaker diffraction effect and higher directivity than the small one. Similarly, we can infer that the shorter wavelength (higher frequency) has higher directivity for the same aperture size. As a result, the diffraction effect, in conjunction with sound superposition and interference, brings about sound directivity.

In short, the sound directivity depends on two factors: signal frequency f and the aperture size a. Mathematically, the signal amplitude Amp at a position in the sound field can be expressed as follows [24]:

$$Amp = \frac{ua^2}{2vr}\sqrt{1+\frac{1}{k^2r^2}}\left|\frac{2J_1(ka\cdot sin\theta)}{ka\cdot sin\theta}\right| \quad (1)$$

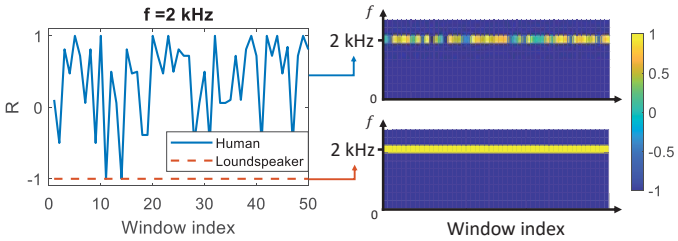


Fig. 3: SFD illustration. Looking at the energy ratio in the time-frequency domain, we obtain the sound field dynamics.

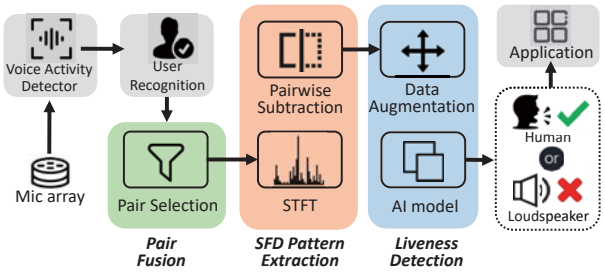


Fig. 4: Overview of VoShield (the colored parts). Components with a grey background are existing APIs.

where u is the vibration velocity of the source. $k = \frac{2\pi f}{v}$, where f is the signal frequency and v is the sound speed. r denotes the distance to the source, and θ represents the angle relative to the x-positive direction. J_1 is the one-order Bessel function [25].

B. Modeling Sound Field Dynamics

The sound directivity leads to different power levels at different positions in the sound field. That means, if sound directivity changes, the power level at the same position will also change. Therefore, the temporal change of power at one position indirectly depicts the variation of sound directivity. However, the power level is also proportional to the sound volume. Given that a microphone array consists of multiple microphones, we perform energy division between two microphones to cancel the volume effect. Specifically, suppose two microphones at the polar coordinates (r_1, θ_1) and (r_2, θ_2). According to Eq. 1, we can calculate the energy ratio R measured at two microphones:

$$R(f,a) = \frac{Amp_1^2}{Amp_2^2} = \left(\frac{r_2}{r_1}\right)^4 \frac{k^2 r_1^2 + 1}{k^2 r_2^2 + 1} \left(\frac{J_1(ka \cdot sin\theta_1) sin\theta_2}{J_1(ka \cdot sin\theta_2) sin\theta_1}\right)^2 \quad (2)$$

We can see that the energy ratio R is irrelevant to the vibration velocity u (*i.e.*, the absolute sound volume). For different time frames of a voice command, r and θ are constants. Therefore, R only depends on the source aperture a and the signal frequency f (recall that $k = 2\pi f/v$). Then, we can define the sound field dynamics SFD of a voice command as *the energy ratio fluctuation along time in the whole frequency band*:

$$SFD^{\boldsymbol{f}}(a) = [R_1^{\boldsymbol{f}}(a), R_2^{\boldsymbol{f}}(a), ..., R_N^{\boldsymbol{f}}(a)] \quad (3)$$

where N is the frame number of the voice command in the time domain. Here, we transform voice signals into the frequency domain for each short frame, so the variable f can be deemed a constant frequency vector $\boldsymbol{f}$, and the aperture size a becomes the only variable. By doing so, we can indirectly profile the dynamics of the sound field, which only depends on the aperture size, a key difference between humans and loudspeakers over time.

Remarks. The key observation on the difference between the real human voice and the loudspeaker-generated one is that the size of a human mouth is time-variant. On the contrary, the aperture size of a loudspeaker is permanently fixed. As a result, the sound field produced by human mouths is **more dynamic** than that generated by loudspeakers, because the size a of the human mouth always varies during speaking.

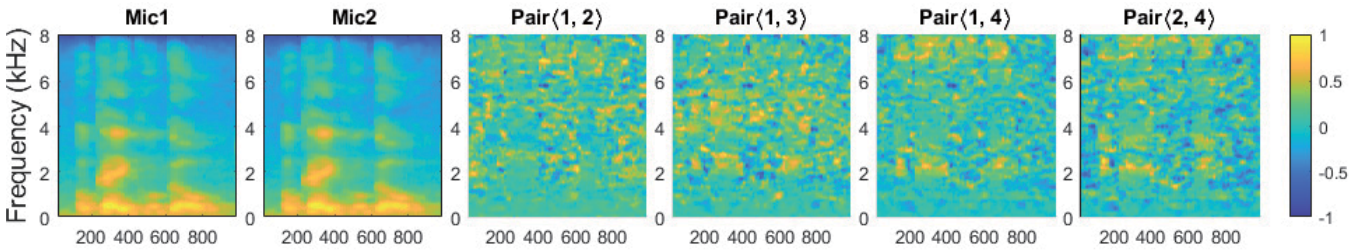
To illustrate a basic idea, we performed a simulation in which a sound source plays a 2 kHz sine tone. The source aperture is fixed to 5 cm to mimic a loudspeaker. Then, we also randomly vary the aperture size within 5 cm to simulate a time-variant human mouth. Fig. 3 shows the normalized energy ratio between two microphones. We can see that the energy ratio R of the human fluctuates rapidly due to the changing size of the mouth. In comparison, the loudspeaker has a pretty stable energy ratio since its aperture size is fixed all the time, which is consistent with our expectations. One may argue that, in practice, the voice includes complicated frequency components, and the time-variant voice content of a loudspeaker will also cause a fluctuant energy ratio. This is why we should look into the energy ratio not only in the time domain but also in the frequency domain. Specifically, we transform the signal per window into the frequency domain, as shown in Fig. 3, and hereby we can obtain the SFD. In a broad sense, we can regard a voice command clip as the composition of multiple single-frequency signals. As such, we can decompose the energy ratio into SFD patterns on different frequency bins. We illustrate the SFD of a real voice command in Fig. 5, and more details will be explained in the next section.

III. System Design

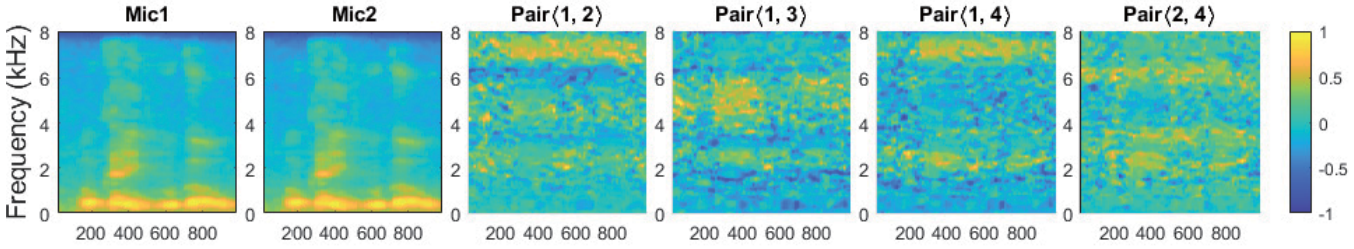
A. Threat Model and System Overview

Our threat model assumes that attackers can obtain victims' voice clips from various sources, such as online meetings. We also assume that attackers can hack vulnerable Internet-connected loudspeakers and hijack these devices to play sounds. In brief, attackers can *remotely* play pre-recorded voice commands via loudspeakers to fool voice assistants.

As shown in Fig. 4, the voice activity detector [26] will capture the arrival of the voice command. Then, a user recognition module [27] can identify whether the voice comes from a legitimate user (*i.e.*, user authentication). Further, VoShield is responsible for checking if it comes from a living human being or an electronic loudspeaker (*i.e.*, liveness detection). VoShield consists of three components: Pair Fusion (Section III-B), SFD Pattern Extraction (Section III-C), and Liveness Detection (Section III-D). In the Pair Fusion module, VoShield checks



(a) Human



(b) Loudspeaker

Fig. 5: The spectrograms of the signals of microphones 1 and 2, as well as the normalized SFD patterns of different microphone pairs. The time unit is ms. We recommend readers see the colored version of this figure.

the microphone array layout and selects several microphone pairs to cover all possible incoming voice directions. Then, the SFD Pattern Extraction component extracts the SFD patterns from these microphone pairs. To deal with voice diversity, we also perform data augmentation and use both collected and augmented data to train the model. Next, SFD patterns are fed to a classifier to detect voice liveness. Finally, if the voice command is classified as one from a real human, the voice signal will be forwarded to the application backend. Otherwise, the voice command is blocked. Note that the user recognition module can also provide the user identity, which enables VoShield to use his/her personalized model to enhance the liveness detection performance.

B. Pair Fusion

This component selects the most effective microphone pairs to facilitate SFD feature extraction and model training. According to Eq. 2, if two microphones and the source are colinear (*i.e.*, $\theta_1 = \theta_2$), or the source is perpendicular to two microphones (*i.e.*, $\theta_1 + \theta_2 = 180°$), the energy ratio R of two microphones will be constant and therefore independent of the aperture size a. The Angle of Arrival (AoA) estimation is a possible way to first detect the voice's incoming direction. However, such a method introduces an additional computation workload. Using only one pair is also unreliable due to noise. Therefore, we cannot completely rely on one pair of microphones to extract SFD patterns. Fortunately, commercial microphone arrays typically consist of several microphones. However, directly using all microphone pairs leads to redundancy of information and increases model training overhead, since many pairs are paralleled and quantify the same SFD pattern.

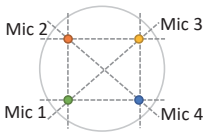


Fig. 6: Mic pairs.

We adopt a simple but effective way to cover all spatial directions, as well as eliminate the impact of redundant pairs. In particular, we select only one from each paralleled pair. As shown in Fig. 6, we choose Pair$\langle 1, 4\rangle$ but exclude Pair$\langle 2, 3\rangle$ because they are paralleled. As a result, we select four pairs (Pair$\langle 1, 2\rangle$, Pair$\langle 1, 3\rangle$, Pair$\langle 1, 4\rangle$, and Pair$\langle 2, 4\rangle$) to make full use of the microphone pairs to improve the SFD measurement. This method brings the following advantages: (*i*) we can always extract useful features using these non-parallel pairs no matter where the sound location is, remitting the AoA estimation. (*ii*) It unifies the channels of the model input for effective training. Besides, we will also introduce another pair fusion method in Sec. III-D. Note that this pair selection principle is capable of other array layouts. What we need is to remove one of the pairs that can be regarded as the two opposite sides of a rectangle from all pair combinations. Next step, we can extract SFD patterns from selected microphone pairs and combine them to facilitate liveness detection.

C. SFD Pattern Extraction

This part is responsible for extracting SFD patterns from multi-channel audio signals. Specifically, we first perform Short Time Fourier Transform (STFT) on the signal of each microphone channel to obtain time-frequency spectrograms. When performing STFT, window size selection is a trade-off between time resolution and frequency granularity. On the one hand, we need a high time resolution to capture the rapid variation of the mouth size. On the other hand, we also require a fine-grained frequency resolution to observe SFD pattern distributions in more frequency components. To this end, we

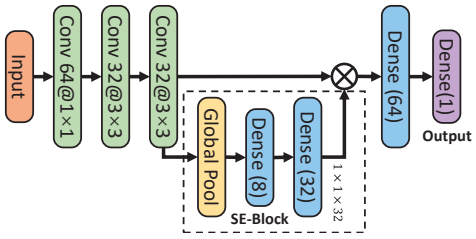
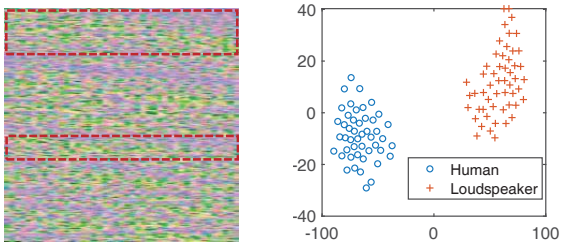


Fig. 7: VoShield network.



(a) Kernel response (b) Feature

Fig. 8: Kernel response and feature visualization. We recommend readers see the colored version.

empirically set the sliding window size as 50 ms with a 75% overlap. Then, the spectrograms will be subtracted pairwise to obtain SFD patterns (the energy ratio is equivalent to the logarithmic energy subtraction).

Fig. 5 shows the spectrograms and SFD of a voice command "OK, Google" received by a 4-microphone array. As shown in Fig. 5(a) and 5(b), we illustrate the spectrograms of two microphone channels (*i.e.*, Mic1 and Mic2) for human-uttered speech and loudspeaker-played commands. We observe that the spectrograms of the two microphones look almost the same since these two microphones share similar voice content. In addition, the spectrograms of the human voice and the replayed sound also look very similar, as they represent the same voice command from the same user. It is also the reason why ASV systems are vulnerable to replay attacks.

However, when we subtract the spectrograms in pairwise order, the SFD patterns differ significantly. Fig. 5(a) and 5(b) show the SFD patterns of four microphone pairs. Evidently, the SFD patterns of human voices are pretty random due to the changing size of the mouth. In comparison, the SFD patterns of the loudspeakers are rather stable, exhibiting visible horizontal *strips* due to the fixed aperture size. After this step, we obtain an SFD feature tensor $I \in \mathbb{R}^{F \times T \times P}$ for a voice command clip, where F is the number of frequency bins, T is the time windows, and P is the number of selected microphone pairs (channels) in Sec. III-B.

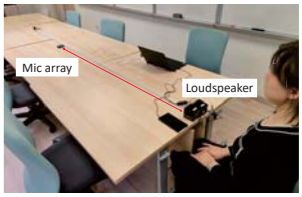
D. Liveness Detection

After extracting the SFD feature, VoShield examines whether this command was spoken by a user or from a loudspeaker. Intuitively, we can use traditional image processing techniques to detect the strip-like pattern in the SFD spectrum, which is the key difference between the voice command from loudspeakers and real users. However, translating this intuitive idea into a concrete implementation involves several technical challenges. First (C1), we observe some breaks along these strips due to noise and short pauses in the voice, which makes the strip patterns much less prominent and hard to detect. Second (C2), the SFD of different microphone pairs may have different significance due to their angles relative to the sound source. For example, Pair$\langle 1, 2\rangle$ exhibits clearer strip patterns than Pair$\langle 1, 3\rangle$ in Fig. 5. Third (C3), the voice content contains various phonemes, and hence the strip pattern may appear in different locations (*i.e.*, different frequency bands at different times) in the SFD spectrogram.

To deal with the first challenge (C1), we utilize a deep learning model to let VoShield automatically learn the strip patterns by leveraging its superior feature extraction and representation capability. Fig. 7 shows the architecture of our network. We first apply three convolution layers to learn the feature embedding. To overcome the pair significance problem (C2), a Squeeze-and-Excitation (SE) block [28] is used as a self-attention mechanism to learn a weight vector as global information. By doing so, we can further fuse the information between different channels and selectively emphasize informative ones. To address voice diversity (C3), we perform data augmentation [29] with random scale and horizontal/vertical translation to simulate the SFD patterns at different temporal and spectral locations. This operation doubles the size of the training data, enhancing the robustness of the model.

To normalize the input size, we use the first one-second clip of a voice command to extract the SFD, in which each microphone pair corresponds to an input channel. Since liveness detection is a binary classification problem (*i.e.*, human (0) vs. loudspeaker (1)), the output of the sigmoid function in the last layer is the likelihood that a voice command is from a loudspeaker. Therefore, we can change the threshold to adjust the confidence of the classification result. The default threshold is 0.5, but we can lower it for sensitive voice commands (*e.g.*, financial operations) to reduce the false acceptance rate (*i.e.*, wrongly accepting an attack command as a real user).

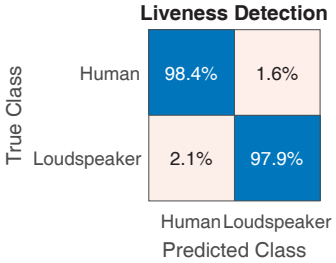
To understand the effectiveness of representations learned by our model, we adopted kernel response visualization [30] to illustrate what the kernels have learned during model training. Fig. 8(a) shows the input response of a kernel in the last convolution layer. We can observe several strip-like patterns (in dashed boxes) with different widths, which indicates that our model can learn such a pattern in SFD as an indicator to detect voice liveness. It is noted that this kernel response comprises four channels, and hence this figure is a true color image after conversion with color distortion. Furthermore, we adopted t-distributed Stochastic Neighbor Embedding (t-SNE) [31] to visualize high-dimensional embeddings extracted in the second-last dense layer. We randomly selected 100 testing voice samples, fed them into the trained model, and extracted corresponding embeddings. Then, we used t-SNE to reduce the representation dimension from 64 to 2 and visualized these audio samples in Fig. 8(b). We can see that samples belonging



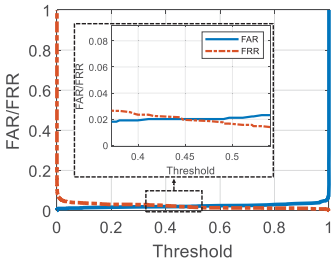
(a) Experiment illustration

(b) Loudspeakers

Fig. 9: Experiment setting.



(a) Confusion matrix.



(b) EER.

Fig. 10: Overall performance of VoShield.

to the same class are closely clustered, whereas samples from different categories are pushed far away. This result indicates that our model can extract effective features to detect the liveness of voice commands.

IV. Implementation

We implemented VoShield with a Respeaker USB 4-microphone array with a typical circular layout in commercial smart devices (*e.g.*, Amazon Echo). The deep learning model is implemented with TensorFlow and trained on a workstation. We add a Batch Normalization layer and a 2×2 Max Pooling layer after each Convolution layer. To prevent over-fitting, we add a Dropout layer with a 0.2 drop rate following the second-last dense layer. The voice command will be forwarded to a laptop to execute the model. VoShield takes approximately 240 ms to perform liveness detection for a voice command sample.

Data Collection. We recruited 12 volunteers from our university (six males and six females) and conducted various experiments in a meeting room, as shown in Fig. 9(a). Participants were asked to speak 30 common voice commands used in [32]. Each command was repeated three times for one session. A smartphone is placed near the user's mouth to record clean speech. Fig. 9(b) shows the loudspeakers used for replaying recorded voice commands, including four different brands and sizes: the built-in speaker in a smartphone Mi 11 pro (12 $mm \times 16\ mm$), an EARISE AL-202 loudspeaker (72 $mm \times 72\ mm$), a Philips SPA20 loudspeaker (80 $mm \times 122\ mm$), and a Dell AX510 soundbar (335 $mm \times 41\ mm$). We used a Respeaker microphone array to record human speeches and replayed commands. Each collection session was repeated with different distances, locations, head orientations, and other various settings, detailed in Sec. V. In total, we collected about 13000 samples.

Baseline. We choose CaField [20], a state-of-the-art liveness detection system based on the sound field, as the baseline. CaField uses the sound directivity value as a feature and trains a Gaussian Mixture Model (GMM) to verify legitimate users. However, sound directivity is sensitive to different positions. Thus, CaField requires users to hold the devices with a fixed gesture. By comparison, VoShield utilizes the *variation* of the consecutive sound directivity measurements, which is resistant to different positions.

V. Evaluation

A. Evaluation Metrics

Same as previous works [9], [20], [32], we use the following metrics to evaluate our system.

- Accuracy. Accuracy is the probability of how well the system can correctly discriminate between live users and loudspeakers.
- False Acceptance Rate (FAR). FAR is the likelihood that the system wrongly accepts an attack as a legitimate voice command.
- False Rejection Rate (FRR). FRR characterizes the rate at which the system mistakenly declares a live user as a replay attacker.
- Equal Error Rate (EER). To balance FAR and FRR, we can adjust the classification threshold (Sec. III-D) to make a trade-off between the probability of incorrect classification for loudspeakers and legitimate users. EER is the value where FAR equals FRR during threshold tuning.
- True Rejection Rate (TRR). TRR is the probability that a command from the loudspeakers is correctly classified.

From the above metric definition, we know that the higher the accuracy/TRR and the lower the FAR/FRR/EER, the better the performance.

B. Overall Performance

In this experiment, we randomly chose 85% of all data for model training and validation, and the remaining 15% were used for performance testing. Fig. 10(a) shows the confusion matrix. Specifically, the overall liveness detection accuracy is 98.2%, and the FAR is 2.1%, indicating that VoShield can effectively distinguish human voice commands from loudspeakers. Fig. 10(b) plots FAR and FRR varying with the threshold changes. We obtain an EER with 2.0% when the threshold is 0.45. In other words, we can set the threshold as 0.45 to strike a balance between the detection ability of loudspeakers and humans. Naturally, we can tune this threshold to adapt VoShield for different purposes. For example, for financial commands, we can lower the threshold a little, and consequently, VoShield has a lower FAR to block replay attacks better. We note that there is no free lunch. A lower threshold also leads to a higher FRR. As a cost, we may need to speak a command several times to pass the

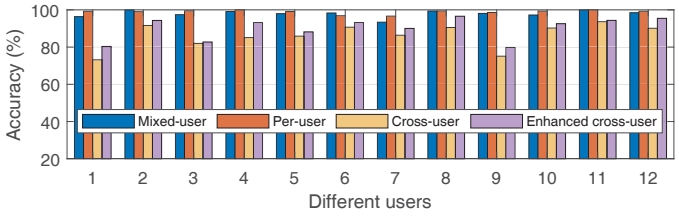


Fig. 11: Performance across different users.

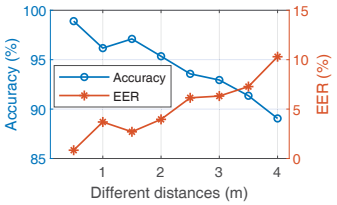


Fig. 12: Performance across different distances.

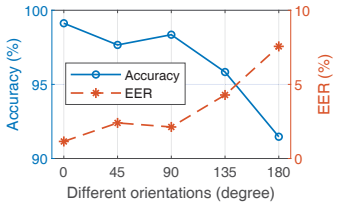


Fig. 13: Performance across different orientations.

VoShield check. But then, it is still acceptable since a repetitive confirmation is required in the financial context, even for the voice assistants without VoShield.

C. Impact of Users

We then investigate the impact of different users on VoShield performance, shown in Fig. 11.

Mixed-user case. We first break down the overall evaluation result and analyze the performance of different users. As we mentioned before, the overall accuracy is 98.2% when the data of all users are mixed together. The highest accuracy is 100% for user 2, and the worst case is 92.4% (user 12). The variance is 0.03%, which indicates VoShield performs stably among twelve different users.

Per-user case. Given that voice interaction is a highly-personal scenario, we also conducted another experiment where a personalized model was trained for individual users. In this setting, for each user, we only used his/her data for model training and testing (similarly, the proportions are 85% and 15%, respectively). We can see that the overall accuracy increases to 98.9%. Therefore, in our system design (Fig. 4), we add a user recognition module so that VoShield can call a personalized model according to different users to improve liveness detection performance.

Cross-user case. Despite the high performance of personalized models, sometimes a user is not always enrolled in model training (*e.g.*, a guest visiting at home). Thus, we also experimented to evaluate the performance of VoShield on unseen users. In this experiment, we trained the model with the data of eleven users and tested it with the remaining one unseen user's data. As the cross-user case shows in Fig. 11, most users still present good performance (approximately 90%), while some users (*e.g.*, 1 and 9) experienced a large degradation. Accordingly, the average accuracy drops to 86.2%. It is in our expectation since although the SFD removes the voice content by doing division between two microphones, it remains the impact of the pause, rhythm, and mouth shape, which are determined by the physiological factors of difference between users. These domain factors prevent current liveness detection systems from high user-independent performance.

Enhanced cross-user case. To partially alleviate this issue, a practical solution is providing some human voice samples of new users to calibrate the model since loudspeaker data collection is not always feasible. In this case, we used the data of eleven participants plus 2 mins of real human voice samples from an unseen user for model training. As shown in Fig. 11, the average performance is improved for all users from 86.2% to 90.1%. This promising result indicates that introducing only voice samples can help the model adapt to unseen users and improve its performance. Thus, we can infer that the performance will be further improved if sufficient voice samples are provided, for example, 5-minute data, which is not a heavy burden for new users. Actually, the performance degradation with unseen users is still an open problem in the area of liveness detection [33]–[36], and we will discuss some possible solutions in Sec. VII-A. We note that VoShield is a complement to current voice authentication systems. Current cross-user performance can still significantly improve the security of voice assistants.

D. Impact of Distances

We collected voice commands at different distances from 0.5 m to 4 m with a step of 0.5 m. Fig. 12 illustrates the results in terms of different distances. Visibly, the accuracy decreases from 98.9% at 0.5 m to 89.1% at 4 m, and the EER accordingly increases from 0.8% to 10.3%. This is because the array has a small size. As the distance increases, the angles of the two microphones relative to the sound source become very close. As a result, the energy ratio between the two microphones tends to be stable, making it hard to differentiate live humans and loudspeakers with SFD patterns. But say, we can observe that the accuracy remains 92.9% when the distance is 3 m. Considering that users prefer to speak voice commands within 3 m of smart speakers [16], this result shows the promising room-scale detection performance of VoShield. According to this result, users are suggested to speak sensitive commands near the device to obtain more reliable protection.

E. Impact of Orientations

We also conducted an experiment with different orientations while keeping the distance between the array and the user fixed at 1 m. 0° and 180° represent that the user is facing forward and backward to the array. We used the same data partition scheme as before for model training and testing. The performance in different orientations is shown in Fig. 13. We can observe that VoShield performs best when the facing direction is 0° (Accuray=99.1%, EER=1.2%). Its performance gradually decreases as the orientation increases. In particular, the accuracy drops slightly to 98.3% when the facing direction is 90°. However, when users/loudspeakers continue to turn their orientations, the performance presents a degradation. The accuracy decreases to 91.5%, and EER increases to

TABLE I: Performance comparison with the baseline.

	TRR(%)	Accuracy (%)	FRR(%)	EER(%)
VoShield	99.5	98.9	1.7	0.8
CaField	91.7	83.9	28.0	15.7

TABLE II: Performance across different devices.

Loudspeaker	Mi11 Pro	AL-202	SPA20	AX510
TRR (%)	97.2	98.3	98.5	96.9

7.6% when the orientation is 180°. Generally, when we face the array, the direct-path component dominates in voice recordings. Thus, the microphone array can easily capture the sound field dynamics. However, when the orientation turns to other directions, the array receives multiple voice reflections and reverberations. After traveling along complex multipath, these reflection components may add up constructively (in phase) or destructively (out phase), leading to SFD pattern distortions. Moreover, human mouths and loudspeakers are both directional sound sources blocked by the head or the enclosure case, and thus voice signals suffer from substantial energy attenuation when the sound source turns its back to the array [37]. As a result, the performance for indirect facing directions is degraded.

F. Baseline Comparison

CaField [20] requires users to hold the devices, which works in the near field (within 0.5 m). For a fair comparison, we compare VoShield with CaField on data collected at 0.5 m. The performance result is shown in Tab. I. We can see that the TRRs of CaField and VoShield are 91.7% and 99.5%, respectively, indicating that both systems can detect replay spoofing attacks accurately. However, in terms of accuracy, CaField (83.9%) performs worse than VoShield (98.9%). Looking in detail, CaField has a 28% FRR, much higher than VoShield (1.7%), which means that many legitimate voice commands are rejected by mistake. This is mainly because CaField relies on directivity features trained with a fixed gesture. Generally, loudspeakers are easily kept static, so CaField can make a quite accurate classification for loudspeakers (TRR). However, there are inevitable head movements when speaking, not to mention different orientations. In this case, many voice commands from other directions may have totally different directivity patterns from the samples used for model training. As such, these human voice commands are prone to be misclassified as illegal attacks, leading to a high FRR. In contrast, VoShield relies on the internal dynamic level of the consecutive sound directivities in multiple windows, which is more resistant to source directions and locations. For the same reason, CaField presents an EER much higher VoShield.

G. Impact of Devices

We also analyze the performance across different devices in the evaluation results. As shown in Tab. II, four loudspeakers also present similar performance because the energy ratio can eliminate the distortion caused by the frequency response of different loudspeakers as well. But we note that the TRR of AX510 is slightly lower than others. We suspect that the soundbar has a large size (335 mm) so the two stereo sub-speakers are apart pretty far. As a result, when the microphone array is physically close to the soundbar, the sound fields of two sub-speakers overlap and interfere with each other, leading to a slight performance drop. Moreover, the first loudspeaker in a cell phone has a small sound cavity and little power output. Consequently, the Signal-to-Noise Ratio (SNR) of voice commands collected at far positions is slightly low, which also causes a lower TRR.

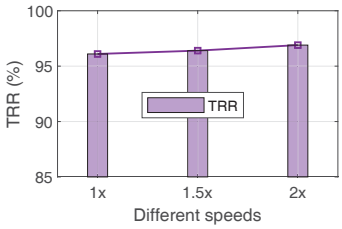


Fig. 14: Performance across different replay speeds.

H. Impact of Speaking Speed

To evaluate VoShield at different speaking speeds, we recorded several participants' voices and played them at 1.5x and 2x speeds to mimic the fast voice content. In this experiment, the model was trained with voice commands at the normal speed (1x). By comparison, we test the model with high-speed replay samples. Fig. 14 shows the result. We can see that the TRR is 96.1% when testing the model with normal-speed replay commands. Interestingly, the performance does not decrease with the increasing replay speed but climbs slightly. When we replay voice commands with the 2x speed, the accuracy increases to 96.9%. This may be because the SFD characterizes mouth movements rather than voice content, and VoShield detects strip patterns to examine voice liveness. As such, the high-speed content narrows the gaps between phonemes and words that may originally break strip patterns to compromise VoShield. As such, we observe stable performance when VoShield encounters fast voice commands.

VI. Related Work

A. Liveness Detection with Additional Sensors

Most works detect voice liveness by building side channels with additional devices or sensors. Camera-based approaches [38], [39] are effective but challenged by poor light conditions. Many works perform liveness detection by correlating voice signals with other signal modalities from a variety of auxiliary sensors, such as motion sensor [6], [7], throat vibrations [8], air pressures in ear canals [14], body sounds in ears [13], and oral flows when speaking [32]. Besides, some works detect replay attacks with magnetometer [40], Wi-Fi [9]–[12], and mmWave radar [41]. In closing, these proposals rely on additional sensors and incur extra costs to build a side channel to detect the liveness of voice commands.

B. Active Acoustic Liveness Detection

Acoustic signals have been widely used to detect users' movement [29], [42], [43] and locations [44]. Thus, many researchers attempt to utilize acoustic signals to sense user movement for voice liveness detection. EchoSafe [45] transmits an audio pulse to detect if the user is present in the room when receiving a voice command. VoiceGesture [18] sends high-frequency acoustic signals to check the Doppler effect caused by the user's articulatory gestures. LipPass [17] and SilentKey [46] detect lip movements for authentication when the user holds a smartphone. Similarly, SPEAKER-SONAR [16] and ChestLive [15] incorporate body and chest movements to examine the liveness of a voice command. Although effective, high-frequency sounds are audible for babies and pets, leading to potential hearing problems.

C. Passive Acoustic Liveness Detection

To overcome the disadvantages of active acoustic methods, recent works detect voice liveness purely on voice commands without actively transmitting sensing signals. VoiceLive [19] and VoicePop [21] measure physiological indicators like the time difference of phonemes and breathing pop sounds in the human voice to detect voice liveness. These two works require users to hold smartphones within a very close distance, so they cannot be used for other devices, such as smart speakers. Blue *et al*. [35] and Void [34] utilize the hardware imperfections as the feature to design a voice liveness detection system. However, their performance suffers from high-fidelity speakers and artificial noise. Some approaches use acoustic features and build deep learning models to combat replay attacks [47], [48], but they extract deep features directly from the voice content, which is easily compromised by attackers who can intentionally manipulate similar voice [49], [50]. ArrayID [33] assumes that the spectrum variance of different microphones is constant, which requires arrays with a circular layout and many microphones to hold the hypothesis. In addition, other features it used, such as Linear Prediction Cepstral Coefficients (LPCC) and frequency energy distribution, are extracted directly from the original signal, which is susceptible to voice manipulation [33].

CaField [20] is the most related work to VoShield. They are both based on sound directivity and do not directly extract features from the voice content. However, CaField takes the absolute sound directivity values as a feature, which requires users to hold the device with certain gestures. By comparison, VoShield utilizes the relative dynamic level of the sound directivity within a command period, which is resistant to different positions and significantly extends the working range.

VII. Discussion

A. User-independent Detection

User-independent liveness detection remains an open problem [34]–[36]. In this research, spectrum noise and some user-relevant physiological features are inevitably involved in model learning. This explains why VoShield cannot perform well in cross-user scenarios (Sec. V-C). One possible way to deal with this problem is to accurately characterize the strip SFD pattern with conventional signal processing techniques. Another solution is using data-driven domain adaption approaches such as adversarial learning to guide our model to learn user-irrelevant features. Finally, few-shot learning and meta-learning can also help the model quickly adapt to new users with a small amount of data.

But thinking in another way, since SFD profiles the unique mouth movement pattern of a human being, it also has the potential for user identification. In this case, the tiny physiological details in SFD, which initially prevent VoShield from user-independent liveness detection, are converted to the key features to identify different users. To validate this idea, we simply retrained our model for the user identification task with human voice samples, and the preliminary identification accuracy is 87.6% among 12 different users. We believe this result is promising and can be further improved. We leave these interesting topics for future work.

B. Sound Field Fabrication Attack

One possible approach to circumvent our liveness detection method might be physically changing the loudspeaker aperture to mimic a human mouth. In addition, attackers can shake or move the loudspeaker when performing attacks. Thus, the sound field dynamics of loudspeakers will inevitably increase. We admit that current VoShield cannot defend against this kind of attack, but we also note that the attacker must be physically present in a user's home, which is beyond our remote attack assumption. Moreover, any movements nearby and the movements of the loudspeaker itself also disturb the sound field, but users will be easily aware of it. Therefore, we believe that remote replay attack with general-purpose loudspeakers is the primary threat to users and is the main focus of this paper.

VIII. Conclusion

Despite powerful functions and huge convenience, voice assistants are exposed to the serious risk of replay attacks. In this paper, we propose VoShield to protect voice assistants through liveness detection. Specifically, VoShield can distinguish a voice command spoken by a live user from its loudspeaker-replayed counterpart. Benefiting from the novel feature Sound Field Dynamics, VoShield extends the working distance to room scale and can work at flexible positions. The evaluation results confirm the applicability and effectiveness of our system. As a complementary protection mechanism to voice authentication, VoShield provides promising liveness detection performance and can be readily integrated into commercial smart devices.

Acknowledgments

This work is supported by the Hong Kong GRF under grant PolyU 152165/19E. Yuanqing Zheng is the corresponding author.

REFERENCES

[1] "Alexa privacy concerns: Is that really concerning? - the week," https://www.theweek.in/news/sci-tech/2021/12/04/alexa-privacy-concerns--is-that-really-concerning-.html, 2022, (Accessed on 07/30/2022).

[2] F. Tom, M. Jain, and P. Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention." in *Interspeech*, 2018, pp. 681–685.

[3] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatic spoofing detection." in *Interspeech*, 2017, pp. 102–106.

[4] A. Janicki, F. Alegre, and N. Evans, "An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks," *Security and Communication Networks*, vol. 9, no. 15, pp. 3030–3044, 2016.

[5] C. Reports, "Samsung and roku smart tvs vulnerable to hacking, consumer reports finds," 2018, https://www.consumerreports.org/televisions/samsung-roku-smart-tvs-vulnerable-to-hacking-consumer-reports-finds/ Accessed Oct 7, 2021.

[6] H. Feng, K. Fawaz, and K. G. Shin, "Continuous authentication for voice assistants," in *Proceedings of MobiCom*, 2017, pp. 343–355.

[7] C. Wang, S. A. Anand, J. Liu, P. Walker, Y. Chen, and N. Saxena, "Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations," in *Proceedings of ACSAC*, 2019, pp. 42–56.

[8] J. Shang, S. Chen, and J. Wu, "Defending against voice spoofing: A robust software-based liveness detection system," in *2018 IEEE MASS*. IEEE, 2018, pp. 28–36.

[9] S. Pradhan, W. Sun, G. Baig, and L. Qiu, "Combating replay attacks against voice assistants," *Proceedings of the ACM IMWUT*, 2019.

[10] Y. Meng, Z. Wang, W. Zhang, P. Wu, H. Zhu, X. Liang, and Y. Liu, "Wivo: Enhancing the security of voice control system via wireless signal in iot environment," in *Proceedings of MobiHoc*, 2018, pp. 81–90.

[11] X. Lei, G.-H. Tu, A. X. Liu, C.-Y. Li, and T. Xie, "The insecurity of home digital voice assistants-vulnerabilities, attacks and countermeasures," in *2018 IEEE CNS*. IEEE, 2018, pp. 1–9.

[12] C. Zhao, Z. Li, H. Ding, W. Xi, G. Wang, and J. Zhao, "Anti-spoofing voice commands: A generic wireless assisted design," *Proceedings of the ACM IMWUT*, 2021.

[13] Y. Gao, Y. Jin, J. Chauhan, S. Choi, J. Li, and Z. Jin, "Voice in ear: Spoofing-resistant and passphrase-independent body sound authentication," *Proceedings of the ACM IMWUT*, 2021.

[14] J. Shang and J. Wu, "Voice liveness detection for voice assistants using ear canal pressure," in *2020 IEEE MASS*. IEEE, 2020, pp. 693–701.

[15] Y. Chen, M. Xue, J. Zhang, Q. Guan, Z. Wang, Q. Zhang, and W. Wang, "Chestlive: Fortifying voice-based authentication with chest motion biometric on smart devices," *Proceedings of the ACM IMWUT*, 2021.

[16] Y. Lee, Y. Zhao, J. Zeng, K. Lee, N. Zhang, F. H. Shezan, Y. Tian, K. Chen, and X. Wang, "Using sonar for liveness detection to protect smart speakers against remote attackers," *Proceedings of the ACM IMWUT*, 2020.

[17] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu, and M. Li, "Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals," in *IEEE INFOCOM 2018*. IEEE, 2018, pp. 1466–1474.

[18] L. Zhang, S. Tan, and J. Yang, "Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication," in *Proceedings of the 2017 ACM CCS*, 2017, pp. 57–71.

[19] L. Zhang, S. Tan, J. Yang, and Y. Chen, "Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones," in *Proceedings of the 2016 ACM CCS*, 2016, pp. 1080–1091.

[20] C. Yan, Y. Long, X. Ji, and W. Xu, "The catcher in the field: A fieldprint based spoofing detection for text-independent speaker verification," in *Proceedings of the 2019 ACM CCS*, 2019, pp. 1215–1229.

[21] Q. Wang, X. Lin, M. Zhou, Y. Chen, C. Wang, Q. Li, and X. Luo, "Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones," in *IEEE INFOCOM*, 2019, pp. 2062–2070.

[22] B. E. Treeby and B. T. Cox, "k-wave: Matlab toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of biomedical optics*, vol. 15, no. 2, p. 021314, 2010.

[23] Z. Yang and R. R. Choudhury, "Personalizing head related transfer functions for earables," in *SIGCOMM*, 2021, pp. 137–150.

[24] L. Cheng, Z. Wang, Y. Zhang, W. Wang, W. Xu, and J. Wang, "Acouradar: Towards single source based acoustic localization," in *IEEE INFOCOM 2020*. IEEE, 2020, pp. 1848–1856.

[25] H. J. Weber and G. B. Arfken, *Essential mathematical methods for physicists, ISE*. Elsevier, 2003.

[26] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, 1999.

[27] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE CVPR*, 2018, pp. 7132–7141.

[29] Y. Wang, J. Shen, and Y. Zheng, "Push the limit of acoustic gesture recognition," *IEEE TMC*, 2020.

[30] "How convolutional neural networks see the world," https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html, (Accessed on 04/12/2022).

[31] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[32] Y. Wang, W. Cai, T. Gu, W. Shao, Y. Li, and Y. Yu, "Secure your voice: An oral airflow-based continuous liveness detection for voice assistants," *Proceedings of the ACM IMWUT*, 2019.

[33] Y. Meng, J. Li, M. Pillari, A. Deopujari, L. Brennan, H. Shamsie, H. Zhu, and Y. Tian, "Your microphone array retains your identity: A robust voice liveness detection system for smart speakers," in *USENIX Security 22*. Boston, MA: USENIX Association, Aug. 2022.

[34] M. E. Ahmed, I.-Y. Kwak, J. H. Huh, I. Kim, T. Oh, and H. Kim, "Void: A fast and light voice liveness detection system," in *USENIX Security*, 2020, pp. 2685–2702.

[35] L. Blue, L. Vargas, and P. Traynor, "Hello, is it me you're looking for? differentiating between human and electronic speakers for voice interface security," in *Proceedings of WiSec*, 2018, pp. 123–133.

[36] L. Zhang, S. Tan, Z. Wang, Y. Ren, Z. Wang, and J. Yang, "Viblive: A continuous liveness detection for secure voice user interface in iot environment," in *ACSAC*, 2020, pp. 884–896.

[37] Q. Yang and Y. Zheng, "Model-based head orientation estimation for smart devices," *Proceedings of the ACM IMWUT*, 2021.

[38] G. Chetty and M. Wagner, "Automated lip feature extraction for liveness verification in audio-video authentication," *Proc. Image and Vision Computing*, pp. 17–22, 2004.

[39] E. Uzun, S. P. H. Chung, I. Essa, and W. Lee, "rtcaptcha: A real-time captcha based liveness detection system." in *NDSS*, 2018.

[40] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen, "You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones," in *2017 IEEE ICDCS*. IEEE, 2017, pp. 183–195.

[41] H. Li, C. Xu, A. S. Rathore, Z. Li, H. Zhang, C. Song, K. Wang, L. Su, F. Lin, K. Ren *et al.*, "Vocalprint: exploring a resilient and secure voice authentication via mmwave biometric interrogation," in *Proceedings of SenSys*, 2020, pp. 312–325.

[42] Y. Zou, Q. Yang, Y. Han, D. Wang, J. Cao, and K. Wu, "Acoudigits: Enabling users to input digits in the air," in *PerCom*. IEEE, 2019, pp. 1–9.

[43] K. Wu, Q. Yang, B. Yuan, Y. Zou, R. Ruby, and M. Li, "Echowrite: An acoustic-based finger input system without training," *IEEE TMC*, vol. 20, no. 5, pp. 1789–1803, 2020.

[44] Q. Yang and Y. Zheng, "Deepear: Sound localization with binaural microphones," in *IEEE INFOCOM*, 2022, pp. 960–969.

[45] A. Alanwar, B. Balaji, Y. Tian, S. Yang, and M. Srivastava, "Echosafe: Sonar-based verifiable interaction with intelligent digital agents," in *Proceedings of the 1st ACM Workshop on the Internet of Safe Things*, 2017, pp. 38–43.

[46] J. Tan, X. Wang, C.-T. Nguyen, and Y. Shi, "Silentkey: A new authentication framework through ultrasonic-based lip reading," *Proceedings of the ACM IMWUT*, 2018.

[47] W. Huang, W. Tang, H. Jiang, J. Luo, and Y. Zhang, "Stop deceiving! an effective defense scheme against voice impersonation attacks on smart devices," *IEEE IOTJ*, 2021.

[48] Z. Li, C. Shi, T. Zhang, Y. Xie, J. Liu, B. Yuan, and Y. Chen, "Robust detection of machine-induced audio attacks in intelligent audio systems with microphone array," in *Proceedings of the 2021 ACM CCS*, 2021, pp. 1884–1899.

[49] D. Mukhopadhyay, M. Shirvanian, and N. Saxena, "All your voices are belong to us: Stealing voices to fool humans and machines," in *European Symposium on Research in Computer Security*. Springer, 2015.

[50] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, 2014, pp. 63–74.