SuperVoice: Text-Independent Speaker Verification Using Ultrasound Energy in Human Speech

Hanqing Guo
guohanqi@msu.edu
Michigan State University
East Lansing, MI, USA

Qiben Yan
qyan@msu.edu
Michigan State University
East Lansing, MI, USA

Nikolay Ivanov
ivanovn1@msu.edu
Michigan State University
East Lansing, MI, USA

Ying Zhu
zhuying4@msu.edu
Michigan State University
East Lansing, MI, USA

Li Xiao
lxiao@cse.msu.edu
Michigan State University
East Lansing, MI, USA

Eric J. Hunter
ejhunter@msu.edu
Michigan State University
East Lansing, MI, USA

ABSTRACT

Voice-activated systems are integrated into a variety of desktop, mobile, and Internet-of-Things (IoT) devices. However, voice spoofing attacks, such as impersonation and replay attacks, in which malicious attackers synthesize the voice of a victim or simply replay it, have brought growing security concerns. Existing speaker verification techniques distinguish individual speakers via the spectrographic features extracted from an audible frequency range of voice commands. However, they often have high error rates and/or long delays. In this paper, we explore a new direction of human voice research by scrutinizing the unique characteristics of human speech at the ultrasound frequency band. Our research indicates that the high-frequency ultrasound components (e.g. speech fricatives) from 20 to 48 kHz can significantly enhance the security and accuracy of speaker verification. We propose a speaker verification system, SuperVoice that uses a two-stream DNN architecture with a feature fusion mechanism to generate distinctive speaker models. To test the system, we create a speech dataset with 12 hours of audio (8,950 voice samples) from 127 participants. In addition, we create a second spoofed voice dataset to evaluate its security. In order to balance between controlled recordings and real-world applications, the audio recordings are collected from two quiet rooms by 8 different recording devices, including 7 smartphones and an ultrasound microphone. Our evaluation shows that SuperVoice achieves 0.58% equal error rate in the speaker verification task, which reduces the best equal error rate of the existing systems by 86.1%. SuperVoice only takes 120 ms for testing an incoming utterance, outperforming all existing speaker verification systems. Moreover, within 91 ms processing time, SuperVoice achieves 0% equal error rate in detecting replay attacks launched by 5 different loudspeakers. Finally, we demonstrate that SuperVoice can be used in retail smartphones by integrating an off-the-shelf ultrasound microphone.

CCS CONCEPTS

• **Security and privacy** → **Biometrics**.

KEYWORDS

Voice Authentication; Ultrasound; Speaker Verification

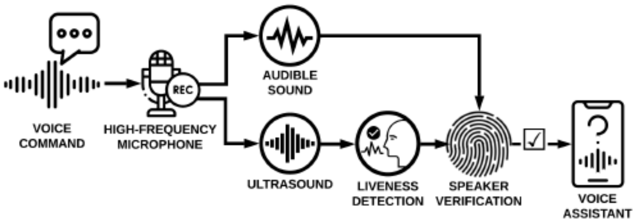


Figure 1: The workflow of SuperVoice.

1 INTRODUCTION

Modern devices increasingly adopt biometric technologies for user authentication. Among various types of human biometrics, such as fingerprint, facial, iris, etc., voice biometric demonstrates great benefits in its high usability, convenience, and security. *Speaker Verification (SV)* systems commonly use voice biometrics to automatically accept or reject a voice input based on the speaker models stored on the smart devices or cloud. Nowadays, all the popular voice assistants, such as Siri, Alexa, and Google Assistant, have integrated SV algorithms for certain wake words (e.g., "Hey, Siri", "Ok, Google").

A more appealing approach, called *text-independent speaker verification*, could accurately and efficiently verify arbitrary utterances from a target speaker based on a limited set of enrolled sentences. Recently, security researchers have demonstrated the susceptibility of SV systems to voice mimicry attacks and replay attacks, where the attackers imitate victims' voices or record/replay them to bypass the SV systems [4, 24, 30, 58]. As the number of sensitive applications (e.g., banking [52]) of voice assistants is growing, practical SV systems aim to achieve not only high accuracy in text-independent speaker verification but also high efficacy in defending spoofing attacks under a limited time budget.

Recent SV studies have explored the distinctive vocal or nonvocal features such as phoneme position [64], cumulative spectrum [3], mouth motion [33, 63], body vibration [14], and sound field [59]. Based on these features, conventional machine learning models have been used to generate speaker models, including correlation (CORR), support vector machine (SVM), Gaussian mixture models (GMM), etc. Meanwhile, deep neural network (DNN) based SV systems use robust neural networks for building speaker models with the prototypical features (e.g., waveform [21, 38, 42], spectrogram [20, 35, 55], and MFCC (Mel-Frequency Cepstral Coefficients) [50]). As summarized in Table 1, most of the existing SV systems cannot simultaneously achieve effective speaker verification and defense against spoofing attacks [3, 33, 63, 64], while others have limitations in their usability, e.g., with the requirement of wearing extra devices [14], staying at the same positions as the enrollment phase [59], etc. Moreover, their discovered vocal or non-vocal features cannot be *transferred* across different speaker models. Although existing DNN-based SV systems [20, 35, 38, 50, 55] do not deal with rigid features, they tend to yield relatively high error rates due to the lack of speaker representative features.

Motivation: In this research, we aim to explore the ultrasound energy in human speech to enhance the accuracy and security of text-independent speaker verification. More specifically, we investigate the unique properties of the *human speech in the human-inaudible ultrasound frequency band* (i.e., frequencies greater than 20 kHz). High-frequency ultrasound components in human speech present several unique properties: first, they are imperceptible by humans but can be captured by an ultrasound microphone; second, individual human speakers can produce ultrasound waves with distinct characteristics, determined by the unique shape of the speech production system and the particular use (e.g. timing) of the system. Recent attacks towards voice assistants, such as DolphinAttack [62] and SurfingAttack [60], leveraged the inaudible ultrasound signals to inject commands into voice assistants. Here, we take a *reversed* approach: rather than utilizing the ultrasound signals for attack, we take advantage of the unique properties of the high-frequency audio spectrum for defense, with the goal of offering a more robust and accurate SV system.

We propose SuperVoice, a robust and secure text-independent SV system, which is applicable to commodity mobile devices equipped with an ultrasound microphone. SuperVoice analyzes an incoming voice command to the device microphone, as shown in Fig. 1. The audio spectrum of the voice command includes both the audible (below 20 kHz) and ultrasound (above 20 kHz) frequency components. SuperVoice then processes these components to extract both the low-frequency and high-frequency feature representations using a liveness detection module and a *two-stream DNN architecture*. These features are fused to a second level of classifier to generate or match a speaker embedding for the speaker verification purpose.

Challenges: The design of SuperVoice faces 3 critical challenges. *i) How to ascertain that the ultrasound feature can represent speaker's voiceprint?* Prior acoustic studies show the evidences that high-frequency energy (from 8-16 kHz) contains useful features to identify an individual speaker [17, 26, 43]. However, none of them focuses on the ultrasound frequency band above 20 kHz. The existing feature engineering techniques such as LPCC (Linear Prediction Cepstral Coefficients), Filter banks, and MFCC cannot be directly applied in high-frequency data, as they are designed for the narrow-band speech data (below 8 kHz). To better utilize the ultrasound features, we design signal processing techniques to extract the unique characteristics from the ultrasound components. *ii) How to use the ultrasound features to detect replay attacks that involve multiple playback devices?* Since the attackers can use different devices (e.g., smartphones, ultrasonic microphone, and ultrasonic speaker) to record and replay the voice signals, it is challenging to design an liveness detection method to cope with different attack devices with varied signal characteristics. *iii) How to design a neural network structure to integrate the ultrasound features?* Since the speech production theory of the low-frequency features and high-frequency features are very different, the integration of both features is particularly challenging. We design a two-stream DNN structure with convolutional filters to process and integrate the ultrasound features.

Table 1: SuperVoice in comparison with other SV systems.

System	Feature	Model	Text Indep.	Security	Transfer
VoiceLive [64]	Phoneme	CORR	✗	✓	✗
VoiceGes. [63]	Mouth	CORR	✗	✓	✗
WiVo [33]	Mouth	CORR	✗	✓	✗
VAuth [14]	Body	CORR	✓	✓	✗
Void [3]	Cum. Spec	SVM	✓	✓	✗
CaField [59]	Sound field	GMM	✓	✓	✗
TE2E [20]	Spectrum	CNN	✓	✗	✓
GE2E [55]	Spectrum	CNN	✓	✗	✓
Siri [50]	MFCC	RNN	✓	✗	✓
SincNet [38]	Waveform	CNN	✓	✗	✓
VGGVox [35]	Spectrum	CNN	✓	✗	✓
SuperVoice	**Ultrasound**	**CNN**	✓	✓	✓

Contributions: To the best of our knowledge, we are *the first* to prove that *ultrasound components (20 $\sim$ 48 kHz)* in human speech can be used to enhance the accuracy, robustness, and security of the SV systems. We demonstrate that the ultrasound components are model-agnostic by integrating them into multiple SV models, all of which achieve enhanced performance in the SV tasks. Surprisingly, the ultrasound components in human speech have been largely neglected prior to this work [25]. In summary, our paper makes the following contributions:

- We demonstrate that human speech does include ultrasound components, and those components can help distinguish among different human speakers. Moreover, the ultrasound components in speech signals can be utilized to identify spoofing attacks by measuring the signals' cumulative energy levels at the high-frequency range.
- We design SuperVoice, a speaker verification and spoofing detection system. By incorporating a two-stream neural network structure with time-frequency spectrogram filters and feature fusion mechanism, SuperVoice achieves high accuracy in text-independent speaker verification.
- We launch a human voice study and collect two datasets for speaker verification and spoofing detection. We recruit 127 participants and record 8,950 audios by 8 different smart devices, including 7 smartphones and an ultrasound microphone. We also replay 500 audio samples to construct a spoofed voice dataset with 5 playback devices. In total, our datasets involve 127 participants with a total of 12 hours audios. We make our datasets publicly available at **https://supervoiceapp.github.io/**.
- We evaluate the performance of SuperVoice and compare it against other SV systems. The result shows that SuperVoice achieves 0.58% equal error rate (EER) in speaker verification, which improves the EER of the top-performing SV system by 86.1%. Remarkably, it only takes 120 ms to test an incoming utterance. Moreover, SuperVoice achieves 0% EER with 91

ms processing time for liveness detection, outperforming all existing approaches. The two-week longevity experiment demonstrates that SuperVoice is suitable for a long-term use, and its performance is barely impacted by the changes in distances and angles.

2 BACKGROUND

2.1 Threat Model

We consider *voice spoofing attack*, which is a malicious attempt to impersonate a genuine speaker in order to execute an unauthorized command on a voice assistant. Three most popular types of voice spoofing attacks include *replay*, *synthesis*, and *conversion* [58]. In a replay attack, the adversary records the legitimate command uttered by a genuine speaker, and replays this command later. The synthesis attack uses the text-to-speech (TTS) generation to create artificial voice commands acceptable by a voice assistant. The conversion attack converts an existing voice command into a different one that can bypass speaker verification. To provide effective countermeasures against voice spoofing attacks, this research aims to develop an end-to-end SV system that can perform both liveness detection and speaker verification.

2.2 Can Humans Produce Ultrasound?

The sounds in human speech are commonly divided into vowels and consonants: the vowels are produced when the air steadily flows through the vocal tract above the larynx (see Fig. 2), while the consonants are more transient in nature and are produced when the flow of air is partially restricted or completely stopped at the vocal fold. The consonants are characterized by *voicing*, *place of articulation*, and *manner of articulation* [10, 23]. Voicing refers to the inclusion of vocal fold vibration as sound source which are quasi-steady and generally harmonic in nature, and the place of articulation represents the location of constriction in the vocal tract which usually results in a highly transient noise. The manner of articulation describes how a sound is altered by the manipulation of airstream flows from the lungs. More specifically,

when two speech organs narrow the airstream to cause friction to occur as it passes through, *Fricatives* are produced. If the airstream is stopped and then released, *Stop* or *Affricate* is produced.

Particularly, the Stop, Affricate, and Fricative consonants are known to exhibit *high frequency energy (HFE)*, since the airstream is deformed by articulations. In this work, *we aim to scrutinize this under-explored and largely neglected phenomenon in human speech, i.e., the consonants carry high energy in the human-inaudible ultrasound frequency range.* We perform experiments to validate that a human speech production generates energy in the ultrasound spectrum during a normal utterance, primarily within speech components such as the Stop, Affricate, and Fricatives. Fig. 3(a) shows the human voice frequency spectra sensed by an ultrasound microphone, in which a significant portion of the acoustic energy is observed beyond 20 kHz. In this study, we are the *first* to show that the acoustic energy beyond 20 kHz (i.e., ultrasound voice components) plays an important role in the speaker verification task and offers an effective and economical solution for liveness detection.

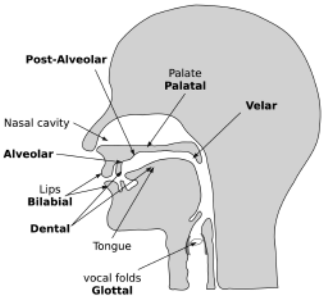


Figure 2: Human's vocal tract and place of articulation.

2.3 Can Ultrasound Components Improve SV Performance?

Carefully examining Fig. 3(a), we find that HFE is produced by certain phonemes (marked by dashed rectangles), such as /sh/, /s/, /sy/, and /s/ within the particular phrase. Fig. 3(b) shows the low frequency spectrum of these phonemes, from which we can see that the phonemes with HFE exhibit less energy below 2 kHz compared with other phonemes.

Remark 1: The phonemes with HFE may lack low-frequency energy (LFE). This phenomenon implies that the traditional LFE-based SV systems may not be able to capture sufficient characteristics of the phonemes with HFE.

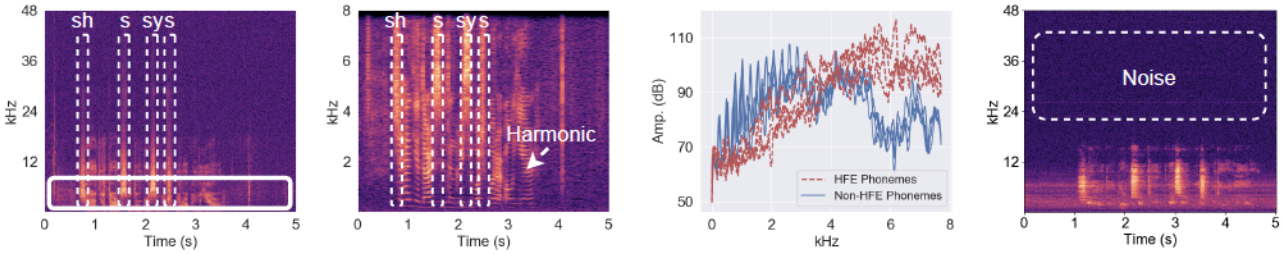
Everyone has a unique vocal track, which leads to a distinctive pitch, formant, and harmonic pattern on the traditional sonic voice spectrum. The modern SV models follow this principle to identify voiceprint by modeling the energy distribution pattern from the LFE spectrum. Fig. 3(c) shows an obvious difference in the voice spectrum between the phonemes with HFE and the ones without HFE. Therefore, by capturing the unique high-frequency characteristics in the phonemes with HFE, the ultrasound components may help boost the performance of the text-independent SV systems.

The most common audio sampling rate of a recorder or loudspeaker is 44.1 (or 48) kHz. Due to the Nyquist theorem, any acoustic energy beyond 22.1 (or 24) kHz will be discarded as shown in Fig. 3(d). Even though some recorders and loudspeakers have higher sampling rate, their frequency responses tend not to be as flat across a wide frequency band as the human speech.

Remark 2: Typical replay attack using loudspeakers could not produce the ultrasound energy. Therefore, the ultrasound energy in human speech can be used to quickly identify loudspeaker-based spoofing attacks.

In this following sections, we present our new discoveries on the specific features of human voice components, which become the core elements of SuperVoice. By conducting a preliminary study on the high-frequency ultrasound components in human speech, we lay the foundation for the rest of this work. The preliminary study aims to answer the following four complementary research questions:

- **RQ1:** *Can the ultrasound components in human speech reflect the speaker identity?*
- **RQ2:** *How consistent are the ultrasound features for each speaker over different speech contents?*
- **RQ3:** *How distinctive are the ultrasound features for each individual speaker?*



(a) Spectrum of a given phrase

(b) Spectrum of LFE components

(c) Voice spectrum comparison

(d) Replayed audio spectrum

Figure 3: Observation of high frequency energy (HFE) and low frequency energy (LFE) of the phrase *"She had your dark suit in greasy wash water all year"* uttered by a human speaker.

- **RQ4:** *Can the ultrasound components help determine the liveness of the audio input?*

2.4 Ultrasound Components and Speaker Identity

To answer **RQ1**, we conduct a theoretical analysis based on the principle of human speech. Generally, the production of speech can be divided into two separate elements: 1) sources of sound such as the larynx, and 2) filters that modify the sources such as the vocal tract. Different from the vowels that only use voicing source, the consonants coordinate three sources: *frication, aspiration*, and *voicing*. Moreover, vowels are produced by relatively open vocal tract, while consonants are produced by constrictions in the vocal tract, as explained in Section 2.2. Specifically, the production of consonants involves more sources, faster changes in articulators, changes in the shape of the vocal tract, and more complicated articulation changes such as the movement of the tongue, lips, and jaw. As a result, the consonants naturally produce a more diverse set of frequency components, which extends to the ultrasound frequency range. Clearly, the uniqueness of the consonant pronunciation depends on a human's vocal tract shape, nasal cavity, oral cavity structure, and the lip and tongue shapes. Among all consonants, we focus on the *Stop, Affricate, and Fricative* consonants, since they produce high-frequency components with a significantly higher energy level (see Fig. 3(a)).

2.5 Consistency of Ultrasound Components

To address **RQ2**, we design an experiment to evaluate whether the ultrasound frequency components are consistent across different speech contents. Conceptually, the *ultrasound component* refers to the speech component with a non-trivial energy above 20 kHz. We first identify the high-energy ultrasound components in an utterance by computing the Short-time Fourier transform (STFT) spectrum of the voice input. The STFT uses a Hann window of length 10 ms, hop length of 2 ms, and FFT size of 2,048 points under 192 kHz sampling rate, which results in 93.75 Hz ($\frac{192{,}000}{2{,}048}$) frequency resolution. Suppose an utterance is divided into N frames as each lasts T in time. We consider the top M frames with the highest cumulative energy above 20 kHz as the frames that contain ultrasound components. Based on empirical observations, M is configured as 100 in this paper.

However, existing studies have demonstrated that STFT spectrum of different phonemes present notable deviations across certain frequency ranges [34, 49]. This indicates that the impact of speech contents could pose a challenge for text-independent SV scenarios. To address this challenge, we calculate the *long term average (LTA)* of the energies of ultrasound components, and the LTA is more stable within the time frame T, expressed as follows:

$$S_{LTA}(f) = \frac{1}{M}\sum_{t=1}^{M} S(f,t), \tag{1}$$

where M is the number of frames that contain high-frequency ultrasound components, $S(f,t)$ is the STFT spectrogram at frequency f and frame t, and t is the frame index within T. The spectrum averaging techniques such as LTA have been used to compare the properties of acoustic signals from random speech data [59]. In essence, LTA can help reduce the impact of different phonemes on the speaker profile. Here, we ask one volunteer to read the sentences S1-S6 (refer to the website https://supervoiceapp.github.io), and collect the spectrogram data to compute S_{LTA}. The results in Fig. 4(a) show that S_{LTA} remains consistent within the frequency range between 16-48 kHz across different speech contents. The variance of S_{LTA} is shown in Fig. 4(b). It is worth noting that S_{LTA} (adapted for low frequency) varies significantly within the low frequency range between 0-16 kHz, which further corroborates that LTA of ultrasound components can be used to improve the performance of SV systems.

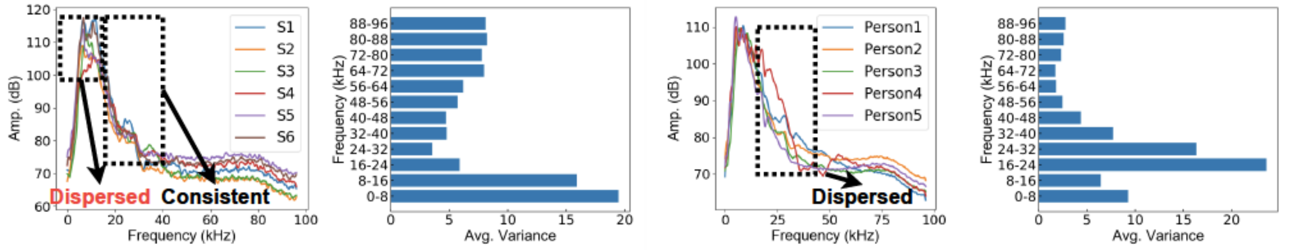
2.6 Distinctiveness of Ultrasound

Next, we aim to address **RQ3**, i.e., whether the ultrasound features from human speech are unique to each speaker, given that each speaker's vocal tract is unique. Prior to answering this research question, we formalize the ultrasound voiceprint for each speaker. The creation of a voiceprint typically involves training with multiple sentences to achieve a reliable and robust voiceprint. Suppose the enrollment dataset is $\mathbb{D}$. The ultrasound voiceprint P is defined as:

$$P = \frac{1}{|\mathbb{D}|}\sum_{s\in\mathbb{D}} S_{LTA}(s), \tag{2}$$

where s denotes the sentence index, and $S_{LTA}(s)$ is the LTA of the ultrasound energy within the sentence s. The ultrasound voiceprint represents the average energy distributions of multiple enrollment sentences.

To evaluate the capability of P in distinguishing among speakers, we enroll the voice of five volunteers (3 males and 2 females) and



(a) Energy of diff. sentences from the same speaker

(b) Variance of energy w.r.t. frequencies

(c) Energy of the same sentences from diff. speakers

(d) Variance of energy w.r.t. frequencies

Figure 4: Ultrasound energies of different sentences spoken by different speakers.

analyze the distinctiveness of P. The results in Fig. 4(c) demonstrate noticeable variations in the ultrasound energy in the range of 16-48 kHz. Fig. 4(d) further indicates that the ultrasound components from different speakers vary the most at the frequency range of 16-32 kHz.

2.7 Ultrasound for Liveness Detection

The aforementioned experiments show that human voice possesses ultrasound components, but the digital loudspeaker generally cannot produce highly distinctive ultrasounds with a high energy. The sound spectrogram produced by a digital loudspeaker is limited by its Analog Digital Converter (ADC) sampling rate, low-pass filters, amplifier, and other loudspeaker hardware. We demonstrate this phenomenon in Fig. 3(a) and Fig. 3(d), where the former shows genuine human utterance has ultrasound components, while the latter spectrogram of a loudspeaker does not contain HFE in the high-frequency band. Therefore, **RQ4** can be addressed by measuring the ultrasound energy in audios. For high-end recorders and loudspeakers that support higher sampling rates, we demonstrate the effectiveness of our design in Section 4.

In summary, we demonstrate the ultrasound components contain speaker identify information. We show that the ultrasound voiceprints based on ultrasound components are consistent across different speech contents. They are distinctive for each speaker, and they can be used for liveness detection to enhance the security of SV systems.

3 SUPERVOICE DESIGN

In this section, by leveraging the discriminative ability of the ultrasound components in the human voice, we introduce SuperVoice to perform liveness detection and speaker verification simultaneously. We first extract the ultrasound features from the voice signals. Then, the ultrasound features are embedded into a two-stream DNN architecture to produce speaker embeddings via the integration of both the low-frequency and high-frequency voice components.

3.1 Overview

The goal of SuperVoice is to utilize the high-frequency components of the human voice to enhance the speaker verification performance while identifying and defending against spoofing attacks. To achieve this goal, SuperVoice includes 3 main stages: 1) *Model Training*, 2) *Speaker Enrollment*, and 3) *Speaker Verification*. Fig. 5 shows the operational workflow of SuperVoice.

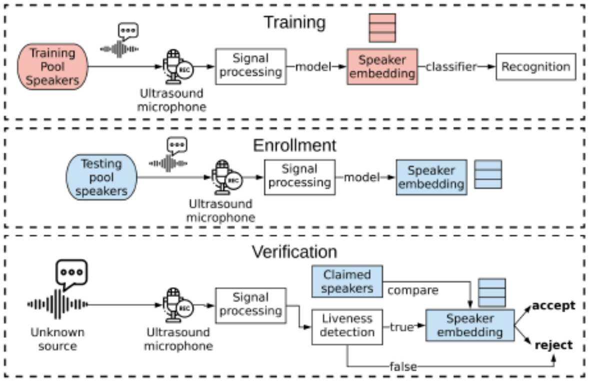


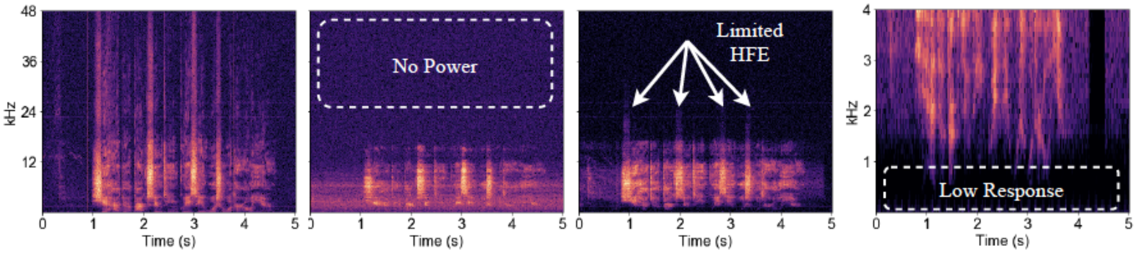
Figure 5: SuperVoice's operational workflow.

During the Model Training stage, SuperVoice first learns how to extract effective features (speaker embeddings) from the voice data of speakers in the training pool. In Speaker Enrollment, the target speakers are required to register their voices in the system, based on which SuperVoice will generate and store the speaker embedding as the speaker's unique voiceprint. Finally, in the Speaker Verification stage, SuperVoice first conducts the liveness detection to ensure the audio source is spoken by a human speaker, and then verifies the speaker identity by measuring the similarity with the claimed speaker embeddings. At every stage, a *Signal Processing* module (see Appendix A) is applied to convert the raw signals to fit the input shape of the model. The processed data are then fed into the *liveness detection* module and the *two-stream DNN network* for speaker verification.

3.2 Liveness Detection

The liveness detection module is designed to differentiate between human voice and spoofed audio by utilizing cumulative spectral energy of captured audio frames with ultrasonic components. We consider three attack scenarios based on different attackers' capabilities.

Scenario 1. Attackers record and replay with commercial devices: Since most of the traditional recording devices 1) do not support microphones that produce high-frequency response; 2) do not have an ADC capable of producing audios with a high sampling rate; and 3) often apply a low-pass filter behind the microphone logic to remove high-frequency components — they are unable to acquire a full spectrum of human voice. An ultrasound microphone,



(a) Human (b) Scenario1 (c) Scenario 2 (d) Scenario 3

Figure 6: Spectrograms from different replay attackers.

on the other hand, can capture a wider spectrum of human voice (see Fig. 6a), including the ultrasound components up to 48 kHz. The digital components of a loudspeaker usually have a sampling rate at or below 48 kHz. Therefore, the replayed audio will not carry any high-frequency ultrasound components as opposed to the genuine human voice (Fig. 6b). As a result, the captured ultrasound components in human voice provide a unique opportunity for developing accurate and efficient liveness detection without heavy computation.

Scenario 2. Attackers record with high-end microphones and replay with commercial speakers: Let us consider an attacker, who uses a high-end microphones (e.g., a microphone with high sampling rate, high-resolution ADC, and wide frequency response) to eavesdrop a victim's voice, and replays it with a commercial speaker such as smartphones or high-quality speakers. In such a scenario, the replayed audio will still carry limited HFE due to the cutoff frequency of commercial speakers, as shown in Fig. 6c. In comparison with Fig. 6a, the lack of HFE in Fig. 6c constitutes a unique feature of the replay attacks.

Scenario 3. Attackers record with high-end microphones and replay with ultrasound speakers: The attackers can also be equipped with high-end microphones and professional ultrasound speakers. In this scenario, although the spectrogram of the replayed audio carries HFE, it possesses limited LFE, as shown in Fig. 6d.

The energy difference between the replayed audio and genuine human voice is evident: the former has nearly zero energy below 1 kHz, while the later presents an intact spectrum.

Based on our observations in Fig. 6, we leverage the cumulative spectral energy of the frames with ultrasonic components and design an *accurate*, *efficient*, and *lightweight* liveness detector to identify if the audio source comes from a loudspeaker or a genuine human speaker. The detector relies on the normalized cumulative energy S_p in different frequency ranges, as defined below:

$$S_p(f) = \sum_{t \in M} S(f,t) - \overline{\sum_{f} \sum_{t \in T} S(f,t)}, \tag{3}$$

where S is the STFT spectrogram, t is the index of frames, T is the total number of frames, and M is the number of frames with ultrasonic components. The first term of the right-hand side summarizes the energies for all the frames with ultrasonic components, and the second term is used for normalization and noise reduction.

To defend against the attacks in Scenarios 1 and 2, we define R_1 as the ratio of the ultrasonic energy over the entire spectrum as follows:

$$R_1 = \frac{\sum_{f=low_1}^{high_1} S_p(f)}{\sum_{f=0}^{high_1} S_p(f)}. \tag{4}$$

The numerator is composed of the normalized accumulative energy on the high-frequency band (from low_1 Hz to $high_1$ Hz), while the denominator uses the energy of the entire spectrum (up to $high_1$ Hz). In this paper, low_1 and $high_1$ are set as 24 and 48 kHz, respectively. Typically, a legitimate human voice will yield a positive value of R_1, since its HFE is greater than the average energy of the entire spectrum (see Fig. 6a). In contrast, a replay attacker with a commercial speaker will yield a negative R_1.

For Scenario 3 in which the attacker possesses a professional ultrasound microphone and high-end loudspeaker, we propose R_2 to examine the proportion of LFE over all frequency bands as follows:

$$R_2 = \frac{\sum_{f=0}^{low_2} S_p(f)}{\sum_{f=0}^{high_2} S_p(f)}. \tag{5}$$

The normalized accumulative energy below 1 kHz is supposed to be negative for replayed audio, since it has lower energy as shown in the dotted frame in Fig. 6d with a dark color. For instance, we set low_2 as 1 kHz and $high_2$ as 4 kHz.

By integrating R_1 and R_2, we consider a voice input as belonging to a genuine human if it satisfies the $(R_1 > 0) \wedge (R_2 > 0)$ condition. Otherwise, it will be classified as a replayed audio.

3.3 Two-Stream DNN Model

After performing the liveness detection, SUPERVOICE begins processing the genuine human speech to verify the speaker identity. For speaker verification, we design a two-stream DNN model to better integrate ultrasound features to improve the SV performance.

Almost all the prior SV studies consider low-frequency audios below 8 kHz, because the typical voice characteristics such as pitch and formants only exist in low frequency range below 8 kHz. However, we observe that the spectrum features above 8 kHz can indeed contribute to the speaker verification. Thus, the question we aim to address in this section is: *how to embed the high-frequency features into the speaker model?*

DNN System Design: Typical machine learning based SV systems use the Neural Network (NN) to obtain feature embeddings [9, 20, 53]. Followed by the feature embedding network, a classifier will perform the classification based on the extracted feature embeddings. SUPERVOICE follows such a structure, i.e., the first level

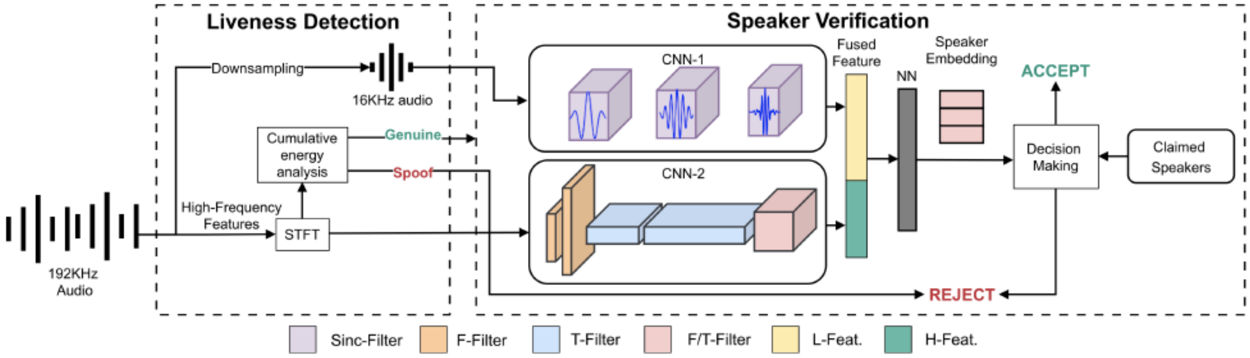


Figure 7: SuperVoice system architecture (L-Feat. represents low-frequency feature embedding; H-Feat. represents high-frequency feature embedding).

of networks conducts feature embedding, while the second level performs the classification.

Feature Fusion: Different from the typical machine learning based SV system, SuperVoice contains two streams of DNN models: one performs feature embedding using the low-frequency components, and the other one embeds high-frequency features. These features will be fused together to construct one coherent feature vector, and then fed into a classifier to produce a unique speaker embedding for every enrolled speaker.

3.3.1 System Architecture. The overall SuperVoice architecture is presented in Fig. 7, which is comprised of three NNs: CNN-1, CNN-2, and an NN classifier.

CNN-1: CNN has been widely used in image recognition tasks, which applies convolutional computation to shrink the input size to obtain a high-level representation of high-dimensional features [61]. We feed the downsampled raw audio containing low-frequency components into the CNN to obtain a low-frequency feature vector. Inspired by SincNet [38], we use Sinc-filters to simulate the behavior of band-pass filters, and add two one-dimensional convolutional layers to further compress the low-frequency feature space.

CNN-2: In the second data stream, CNN-2 is designed to embed high-frequency voice components. Since the existing CNNs, such as VGGNet [45], ResNet [19], Inception [48], are designed for image classification, they apply multiple convolutional layers with different shapes (mostly squared shape) of filters to capture the unique characteristics of different object shapes in the image. As opposed to the image which consists of pixels, the spectrogram data possesses both time and frequency components. Here, we design a new CNN architecture with three complementary time-frequency convolutional filters to extract the HFE distribution and phoneme-level high-frequency representations.

F-Filters: The purpose of these frequency-domain filters (F-Filters) is to extract the HFE distribution $S(f)$ at the frequency domain. We design a sequence of vertical shaped filters to convolve the high-frequency spectrogram. The size of F-Filter is determined by the range of frequencies involved in the convolutional computation. Based on the observation that HFE distribution can be used as the speaker voiceprint (see Fig. 4(a), 4(c)), in order to extract a finer-grained energy distribution with a higher frequency resolution across the frequency range, we construct 64 F-Filters, whose size is 9×1 with dilation 1×1 and 2×1. As a result, the filters cover the frequency range from $9 \cdot 93.75 = 843.75$ Hz to $9 \cdot 2 \cdot 93.75 = 1,687.5$ Hz.

T-Filter: Two time-domain filters (T-Filter) are designed to learn the high-frequency phoneme-level representation. The T-Filter covers a time duration, which is shorter than a phoneme length to ensure that the convolution process can occur within a single phoneme. The time-domain resolution can be computed by $hop_{STFT}/192\ kHz \approx 2.7$ ms. After applying the 64 1×9 T-Filters that is dilated by 1×1 and 1×2, the convolution computation covers the time-domain resolution between $9 \cdot 2.7 = 24.3$ ms and $9 \cdot 2.7 \cdot 2 = 48.6$ ms. Since 48 ms is shorter than typical phonemes, the time-domain frames can represent the detailed information from a single phoneme.

F/T-Filter: At the final stage of CNN-2, we design a sequence of square filters (F/T-Filter) with the size of 5×5 to convolve both time-domain and frequency-domain features concurrently. F/T-Filter merges the extracted high-frequency characteristics from both the time and frequency domains, in order to yield a more representative ultrasound energy distribution for a particular speaker.

NN classifier: Finally, the NN classifier takes the fused features that are concatenated by the output feature vectors of the CNN-1 and CNN-2 and compresses them into a desired speaker embedding dimension. Here, we use a fully connected layer as the NN classifier. More detailed neural network parameter settings can be found in Appendix B.

Speaker Embedding Generation and Matching: The speaker embedding is generated by the NN, as shown in Fig. 7. The NN is essentially a fully connected layer, which maps the fused feature vector to a speaker embedding vector. Please refer to Appendix C for the details of feature alignment.

When the speaker model produces the speaker embedding based on the given audio source, SuperVoice will compare the cosine distance with the existing speaker embeddings, which have been generated during the enrollment stage. Every sentence spoken by an authorized speaker will produce a representative speaker embedding. For example, if speaker A enrolls three sentences into SuperVoice, the model will generate three embeddings for A. To accept the audio as belonging to speaker A, the average cosine similarity between the tested speaker embedding and the enrolled one should be greater than a similarity threshold γ as shown below:

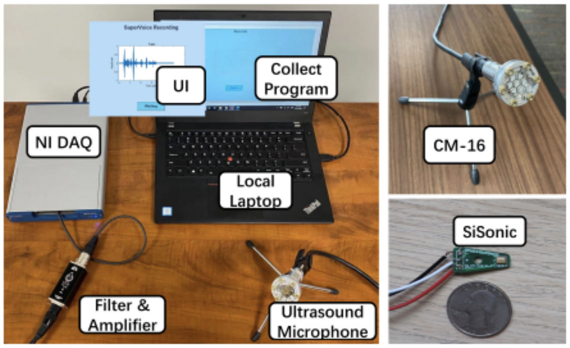


Figure 8: Dataset collection platform.

$$decision = \begin{cases} accept, & similarity \geq \gamma \\ reject, & similarity < \gamma \end{cases} \quad (6)$$

where $similarity = \sum_{i=0}^{N} cos(emb_i, emb)/N$, N is number of enrolled audios for the speaker, emb_i is the i^{th} speaker embedding, and emb is the speaker embedding to be verified.

3.3.2 Model Training/Testing. It is noteworthy that although the purpose of the NN models is to extract the speaker embedding, they operate differently in three stages (see Fig. 5). The model will learn how to extract the most representative speaker embeddings via training with a speaker recognition task. It means that the output of NN will connect to another fully-connected layer that maps the result dimension from speaker embedding to the number of classes in the speaker recognition task. For example, the model will predict a speaker label for the test utterance, and then refine the network parameters via the backpropagation of losses. In Speaker Enrollment stage, however, the model simply loads the set of parameters that achieve the best results in speaker recognition task, and then extracts the speaker embeddings for the enrolled speakers.

4 EVALUATION

In this section, we evaluate the performance of SuperVoice on spoofing detection and speaker verification, i.e., how well SuperVoice can verify a claimed speaker and reject a spoofed audio or a stranger's voice. Furthermore, we integrate the high-frequency features extracted by SuperVoice into existing SV models to show the transferability of high-frequency features in enhancing different types of SV models. To have a fair evaluation, we collect several speech datasets as listed in Section 4.1. Our experiments are conducted on a desktop with Intel i7-7700k CPUs, 64GB RAM, and NVIDIA 1080Ti GPU, running 64-bit Ubuntu 18.04 LTS operating system. The model complexity and time consumption are measured in such a hardware configuration.

4.1 Speech Data Collection

Human Voice Recording. The voice data in all the existing public datasets is collected using regular microphones with at most 48 kHz sampling rate [16, 36] to record data within [0-24] kHz.

In order to investigate the high-frequency ultrasound components in the human speech, we collect our datasets for evaluation, including Voice-1, Voice-2, and Voice-3 datasets. The Voice-1 is collected by a high-end ultrasound microphone, Voice-2 is collected by regular microphones on various smartphones, and Voice-3 is collected by a low-end ultrasound microphone. *In total, we collected 9,050 speech utterances from 127 volunteers, the data collection process and user study have been approved by our school's IRB board.*

Dataset Collection Platform: There are many options for the off-the-shelf ultrasound microphone (i.e., SiSonic SPU0410LR5H_QB MESE [46] and Avisoft condenser microphone CM16 [5]). The first microphone can capture the ultrasound frequency band up to 96 kHz, which only requires a 1.5 V to 3.6 V power supply. The low power consumption and low cost ($2/piece) make it suitable for most smartphones. The second microphone provides a more flat frequency response over the entire frequency band, allowing it to collect better-quality ultrasound recordings. For this reason, we deploy both SiSonic SPU0410LR5H_QB microphone and Avisoft microphone for data collection. The microphone and data capturing equipment are displayed in Fig. 8. For each participant, we informed them of the purpose of the experiment and then recorded their voice. The participants spoke facing forward to the microphone with a distance of 30 cm. Each participant was requested to speak 4 types of sentences, totaling 100 sentences. The instruction for the experiment and a part of the sentences can be found in Appendix D.

Voice-1: Voice-1 includes the voice data from 77 volunteers, totalling 7,700 utterances using 192 kHz sampling rate. Among the 77 volunteers, most of them are college students with ages ranging from 18 to 56, including 38 males and 40 females. For detailed dataset information, please refer to website **https://supervoiceapp.github.io**.

Voice-2: Voice-2 is constructed by recording 25 sentences by 50 participants with different models of smartphones listed in Appendix D. The smartphones' sampling rate is 48 kHz. As traditional speaker model leverages voice features below 8 kHz, Voice-2 helps validate the effectiveness of high-frequency features within [8, 24] kHz range recorded using different phones. In total, Voice-2 includes 1, 250 utterances with 48 kHz sampling rate.

Voice-3: Voice-3 includes 200 audios recorded from 20 participants. Different from Voice-1, we collect Voice-3 by the cheap SiSonic ultrasound microphone. Every participant read a sentence twice in Common type (See Appendix D), in total 10 audios per volunteer. The purpose of Voice-3 is to validate the performance of SuperVoice with a cheap ultrasound microphone that can be integrated into smartphones [22].

Spoofing Voice Dataset: We implement the spoofing attacks by replaying the voice data collected in Voice-1 using 5 playback devices and 2 recording devices, including 2 smartphones, 2 high-end commercial loudspeakers, and one ultrasonic speaker. To detect the replay attack, we deploy an ultrasound microphone to record the replayed spoofing audio. The purpose of this dataset is to comprehensively evaluate the capability of ultrasound components for liveness detection.

4.2 Performance Metrics

The performance metrics we use for the SV task are *False Acceptance Rate (FAR)*, *False Reject Rate (FRR)*, and *Equal Error Rate (EER)*. FAR represents the rate of SuperVoice falsely accepting an unauthorized speaker, FRR is the rate of SuperVoice rejecting a legitimate voice, and EER is the rate where the FRR and FAR are equal. We further use *Classification Error Rate (CER)* to evaluate the speaker recognition (SR) performance, which is defined as the ratio of misclassified recordings versus the total recordings. For the user study, we develop SuperVoice as an end-to-end desktop application, and use *Success Rate* to measure the percentage of successful attack defenses by SuperVoice, i.e., the times of correct recognition of the voice owner over the total number of attempts.

4.3 Speaker Verification Performance of Integrated Models

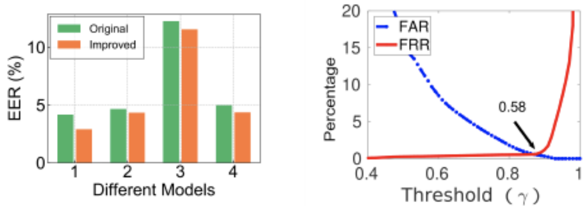
To make a fair comparison with other existing speaker models, we reproduce all the models in Pytorch framework. We use the Pytorch version 1.2.0 with Python version 3.6.9. The GE2E [55] and Void [3] are closed source, which we reproduce based on their descriptions. The GMM-UBM [39], VGGVox [35] have open-source MATLAB codes, while SincNet [38] and STFT+LCNN [31] are implemented with the Pytorch framework. All of the models are evaluated with the same datasets described in Section 4.1.

Direct Ultrasound Integration: First, we conduct a performance evaluation of 4 popular SV models: GMM-UBM, SincNet, VGGVox, and GE2E. We follow each model's specification to configure the input and model parameters. Then, we evaluate their performance using the downsampled low frequency data ([0-8] kHz) and the original data ([0-96] kHz for **Voice-1** and [0-24] kHz for **Voice-2**). The performance comparison is presented in Table 2. The performance with low-frequency data is relatively consistent with their reported results. When the high-frequency data is included in the modeling process, the performance of every model deteriorates significantly. This indicates that the high-frequency data cannot be directly used to distinguish among different speakers.

Table 2: EER performance (%) comparison among GMM-UBM, SincNet, VGGVox, GE2E with two different datasets.

Speaker Model	Voice-1		Voice-2	
(kHz)	[0 – 8]	[0 – 96]	[0 – 8]	[0 – 24]
GMM-UBM	12.25	42.23	13.33	17.56
SincNet	4.17	18.23	4.19	7.04
VGGVox	4.64	16.63	4.66	6.75
GE2E	4.98	19.15	4.97	6.96

Improved Ultrasound Integration: For a better integration of the high-frequency data, we adopt the architecture of SuperVoice: (1) using *CNN-2* to handle high frequency data, and (2) replacing *CNN-1* with the existing speaker models. To validate the efficiency of integrating high frequency data in the smartphones, we conduct an experiment with **Voice-2** ([0-24] kHz) and present the results in Fig. 9(a). The green bar with rectangle pattern indicates the EER performance with the downsampled [0,8] kHz data, and the orange bar with cross pattern shows the performance with the addition of high-frequency features in the range of [8, 24] kHz that are extracted by *CNN-2* and feature fusion technique. The results show that the EER of SincNet has a drop from 4.19% to 2.89%, and the EER of VGGVox decreased from 4.66% to 4.12%. Overall, the EER performance improvement is around 16.93% on average with SincNet, VGGVox, and GE2E. For the GMM-UBM model, the EER performance has also improved slightly. The results demonstrate SuperVoice's transferability, i.e., it improves other SV models' performance by integrating the high-frequency feature embeddings. We then evaluate the performance of ultrasound integration in SuperVoice using Voice-1 dataset. The result in Fig. 9(b) shows the FAR and FRR of SuperVoice w.r.t. similarity threshold γ, and it indicates that the EER performance of SuperVoice is 0.58%.



(a) Performance improvement

(b) EER of SuperVoice

Figure 9: Performance of: (a) ultrasound integration in existing models (1, 2, 3, and 4 represent SincNet, VGGVox, GMM-UBM, and GE2E models), tested on Voice-2; (b) ultrasound integration in SuperVoice system, tested on Voice-1.

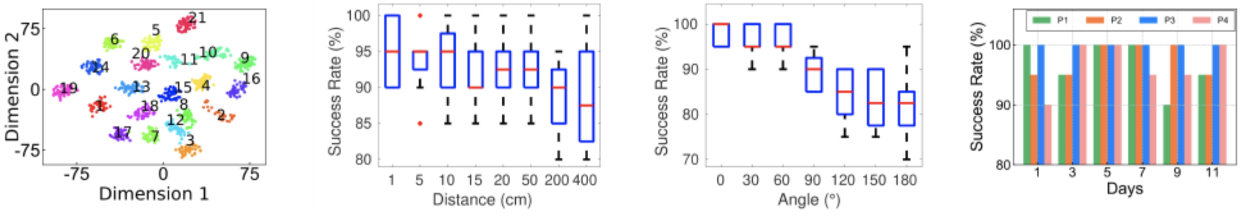
Table 3: EER performance (%) of SuperVoice on different datasets

SuperVoice	Voice-1		Voice-2		Voice-3	
	SV	SR	SV	SR	SV	SR
No HFE	4.17	5.87	4.19	6.74	6.75	7.84
[8-16] kHz	3.98	4.79	3.77	4.87	5.74	5.95
[8-24] kHz	2.89	2.27	3.21	3.32	3.45	4.51
[8-48] kHz	**0.58**	**1.61**	-	-	**1.87**	**3.01**
[8-96] kHz	5.79	7.31	-	-	9.52	14.2

4.4 Impact of Frequency Ranges

Next, we evaluate the performance of SuperVoice with different frequency ranges of the high-frequency data. Both SV (i.e., to verify the voice is from an authorized user) and speaker recognition (SR) (i.e., to recognize the voice of a specific authorized user) tasks are conducted to measure the EER and CER performance. The results in Table 3 show that SuperVoice can achieve the EER performance of 0.58% with Voice-1, the best among all the existing speaker models. It is noteworthy that the best models that tested on Voice-1 is SincNet, which has 4.17% EER (see Table 2). To further evaluate SuperVoice on smartphones with an affordable ultrasound microphone (e.g., SiSonic SPU0410LR5H), we evaluate the performance with the Voice-3 dataset. The results show that, even with low-end ultrasonic microphone, SuperVoice achieves significant performance improvement.

Remarkably, SuperVoice improves the EER performance of the best SV model by 86.1% (or 55.1% with low-end microphone), via the incorporation of ultrasound frequency components. We also find that incorporating high-frequency features below 48 kHz will produce better performance compared with the higher frequency range. Among all the configurations of frequency ranges, the range of [8, 48] kHz provides the best SV and SR performance in terms of EER and CER. Unsurprisingly, with a complete spectrum of [0, 96] kHz,



(a) t-SNE result

(b) Distance vs. success rate

(c) Angle vs. success rate

(d) Durability vs. success rate

Figure 10: The user study of SuperVoice.

both SV and SR performance degrades, as more indistinguishable noises are incorporated in the model to perplex the SR/SV tasks.

4.5 User Study

Besides the benchmark evaluation presented in the previous sections, we perform two user studies to further test the effectiveness and robustness of our system. (a) **Flexibility Study:** users use our system at home and make speeches from random positions; (b) **Longevity Study:** users use our system over a long time span. Fig. 10(a) visualize the t-SNE result [32] of 20 participants in a 2D space, which clearly shows the 20 clusters of speakers.

To conduct the user studies more efficiently, we develop an end-to-end SuperVoice desktop application.

Flexibility Study: We deploy the end-to-end application and ask 8 volunteers to enroll in the application. Once they are successfully enrolled, they are instructed to speak to the ultrasound microphone at different distances and angles to test the system's recognition performance. Each volunteer makes 20 test attempts. Figs. 10(b) and 10(c) present the impact of distance and angle, respectively. The results show that SuperVoice reaches high success rate (95 – 100%) within 50 cm. Although the success rate may drop to 85% beyond 50 cm in the worst case, the average accuracy at 400 cm reaches 87.5%. As for different angles, the recognition performance declines from 95% to 85% when the speaker is side facing the microphone. The performance degradation is caused by a specific characteristic of the ultrasound microphone (i.e., CM-16 delivers different gains at different angles according to its polar diagram).

Longevity Study: For the second user study, we test the longevity performance of our system by tracking the usage of 4 users over 11 days. The participants enroll their voices on the first day, and attempt 20 times per day to use SuperVoice to identify their respective voices. As illustrated in Fig. 10(d), the average success rate is more than 95%, which means less than 1 over 20 attempts failed. In the end, we found no evidence of a degrading performance pattern over time.

4.6 Runtime Performance

In this section, we compare the training time and testing time of SuperVoice with SincNet, VGGVox, GMM-UBM, and GE2E models. The training time is the total time used to create a speaker model with the training pool of Voice-1, while the testing time represents the time spent to verify an incoming utterance.

Table 4 presents the runtime result. Among all the models, the GMM-UBM model is the fastest in terms of training and testing time with the worst EER. SincNet converges very fast during the training phase due to its special convolutional neural design, while the proposed SuperVoice also delivers comparable training time. During the testing, SuperVoice outperforms VGGVox and GE2E models due to its lightweight model with a small number of parameters. It is worth noting that introducing high-frequency features does not affect the testing speed. The results show that SuperVoice could retain comparable runtime performance with enhanced speaker verification performance.

Table 4: Runtime comparison.

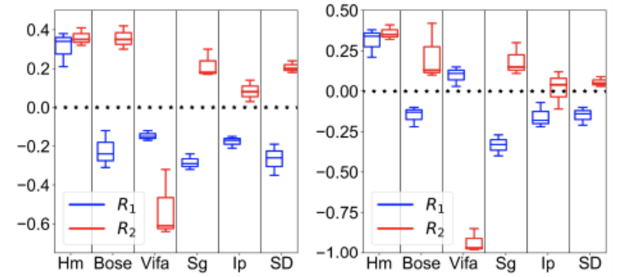
Model	Training time (sec.)	Testing time (sec.)
GMM-UBM	7,149	0.074
VGGVox	11,308	0.279
GE2E	10,348	0.21
SincNet	8,180	0.134
SuperVoice	**8,413**	**0.120**

4.7 Liveness Detection Performance

In this section, we conduct experiments to verify the performance of liveness detection described in Section 3.2. We prepare two types of recorders and 5 playback devices to replay the recordings. For every speaker, we replay 20 audios at a fixed position (facing forward in 10cm) and volume ($60dB_{SPL}$). The defender uses the low-cost SiSonic ultrasonic microphone to monitor the replayed audios.

Attackers Record with Common Recorder: We first replay audios that were recorded from a smartphone (Samsung S9). The boxplot in Fig. 11(a) demonstrates the results with different speakers. From left to right, we have Human genuine voice (Hm), Bose SoundTouch 10 speaker [7], Vifa ultrasonic speaker [6], Samsung S9 phone (Sg), iPhone 12 (Ip), and SADA D6 speaker [41]. The results show that all the replay devices present a negative R_1. This is attributed to the lack of HFE in the recorded audios by the smartphone. In contrast, the genuine human voices have positive R_1 and R_2, which is consistent with our analysis in Section 3.2. In the end, SuperVoice achieves 0% EER.

Attackers Record with Ultrasound Recorder: Now, we consider the attackers use a high-end ultrasonic microphone to record the victims' voices. We select 20 audio samples with 192 kHz sampling rate in Voice-1 as the source to replay them by 5 loudspeakers. The result in Fig. 11(b) shows that the commercial speakers still cannot produce any HFE, yielding all negative R_1. Moreover, a substantial gap exists between the genuine and replayed voice from any specific replay devices, which indicates that the liveness detection of SuperVoice is robust against any attack devices. For



(a) Recorded by smartphone recorder (b) Recorded by ultrasound microphone

Figure 11: Replay attacks

Table 5: Liveness detection performance comparison.

Models	# Feat.	Time (sec.)	EER(%)
CQCC + GMM[30]	14,020	0.159	12.08
LPC + CQCC + GMM	14,026	0.173	13.74
STFT + LCNN[31]	84,770	0.321	8.8
Void[3]	97	0.103	11.6
SuperVoice	**4**	**0.091**	**0**

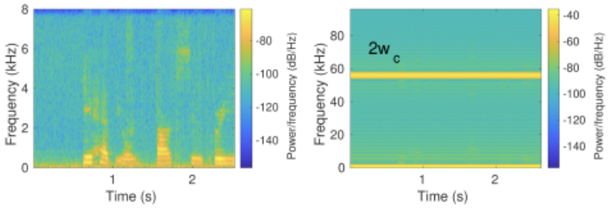
the attacker with an ultrasonic speaker (Vifa), we observe a positive R_1. However, its negative R_2 signifies the low LFE. In the end, SuperVoice again achieves 0% EER, consistently confirmed by 200 attack attempts.

Defense Performance Comparison: Here, we compare the liveness detection performance with 4 state-of-the-art liveness or spoofing detection models. We first justify our reproductions by testing them on ASVSpoof [30] dataset and all of them reach similar performance as they claimed. We then evaluate all the models using our spoofing dataset in terms of the number of features, average detection time, and the EER performance. Table 5 presents the liveness detection performance comparison results. Among all the models, the STFT+LCNN model runs the slowest with the most number of features, while its EER performance is the best among the four models. Compared with the existing models, SuperVoice only uses four accumulative power features in R_1 and R_2, and achieves the fastest runtime performance with 0% EER. In consistent with the measured data in Fig. 11, which visualizes the manifest gap between genuine and spoofed sound, SuperVoice achieves superior liveness detection performance in terms of both the runtime and EER performance for both the traditional loudspeakers and ultrasound speakers.

5 DISCUSSION

In this section, we discuss the limitations of SuperVoice, the defense against the inaudible attacks, and the future research directions.

Commands Without Fricative Consonant: As mentioned before, we observed that some phonemes, especially the fricative and stop consonants, retain high energy above 20 kHz. However, if a spoken sentence does not contain any fricatives, we may not be able to find an energy spike in the spectrum. Fortunately, we observe the HFE from most of *Non-fricative command* because the speaker always alters the air flow by their articulations, and this high-frequency component can be adopted by SuperVoice as an extra feature for speaker verification. For those sentences that only include low-frequency energy (below 8 kHz), the low-frequency stream of our DNN architecture guarantees that SuperVoice does not experience any performance degradation with high-frequency features extracted from the non-fricative commands.



(a) Inaudible attack on regular microphone (b) Inaudible attack captured by SuperVoice

Figure 12: The defense against inaudible attacks.

Long-Range Speaker Verification: In this paper, we assume that the human speakers are within a close distance from the ultrasound microphone. Prior research found that long range speaker verification is challenging mainly due to the reverberation of sound and attenuation of the acoustic energy [37]. In SuperVoice, the range of voice commands will affect the received power of both the low-frequency and high-frequency components, especially for the fricative and plosive consonants. A power amplifier may be able to address the power attenuation issue, and we plan to evaluate its effectiveness in a long-range speaker verification in the future work.

SuperVoice on Smartphone: In our experiments, we run SuperVoice on a desktop with an ultrasound microphone. We experiment with smartphones supporting high sampling rate (i.e., 192 kHz) to capture high-frequency voice components. Yet, we find that, due to the low-pass filter in the microphone system, all the frequency components above 24 kHz have been filtered out.

One possible solution is to replace the microphone in the smartphone with the one supporting ultrasound frequency [46], or use an external microphone that can be connected to the smartphone. We also evaluate the performance of an external ultrasound microphone, i.e., Echo Meter Touch 2 [57], in capturing high-frequency components in voice signals. The external ultrasound microphone is attached to Samsung Galaxy S9 with a sampling rate of 256 kHz. The results show that the voice data captured by external microphone can achieve similar SV performance as the standalone ultrasound microphone.

Inaudible Attack Defense: The inaudible attacks leverage the non-linearity of a microphone to perform an inaudible command injection attack through ultrasonic speakers [40, 60, 62]. The basic idea is to modulate the voice commands to ultrasound frequency band, and then transmit the modulated signal through an ultrasonic speakers. Due to the non-linearity of the regular microphone, the ultrasonic signal will shift frequency to the audible frequency range in the microphone. As a result, the command can be perceived by voice-activated devices.

We evaluate SuperVoice's capability in detecting inaudible attacks. We use a voice command "*She had your dark suit in greasy* *wash water all year*" from Google TTS as a legitimate signal. This command is modulated to the inaudible frequency at $w_c = 28$ kHz. Figs. 12(a) and 12(b) show the spectrogram of inaudible attack towards both a regular microphone and SuperVoice's ultrasound microphone. The regular microphone only captures frequency components in the range $[0, 8]$ kHz, while SuperVoice can capture a $2w_c = 56$ kHz component that can be used to immediately detect the inaudible attack. Therefore, SuperVoice effectively defeats inaudible command injection attacks to voice assistants.

6 RELATED WORK

Speaker Verification: Prior studies have identified different voice features for speaker verification models. They use speech spectrum, speaker pitch and formants, and even raw audio waveforms as inputs [20, 21, 29, 38, 42], from which various voice features can be extracted, such as Filter Banks, MFCC (Mel-Frequency Cepstral Coefficients), LPC (Linear Prediction Coefficients), LPCC (Linear Prediction Cepstral Coefficients), or any combination of them [11, 47, 53]. With the voice features in hand, researchers further use GMM-UBM (Gaussian Mixture Model Universal Background Model) [8], JPA (Joint Factor Analysis) [27], and neural networks [9, 12, 13, 20, 35, 36, 38, 53, 55] to generate speaker models. Based on the speaker models, several classifiers such as support vector machine (SVM) [8, 56], cosine similarity [20, 44], and PLDA (Probabilistic Linear Discriminant Analysis) [13, 28] have been employed to make (mostly probabilistic) SV decisions.

Spoofing Detection: Existing spoofing detection solutions explore both non-vocal and vocal physical parameters of the human speaker to differentiate between human voice and spoofed sound. Among the approaches that use non-vocal physical parameters is VoiceLive [64], which leverages smartphone's two microphones to capture the difference in the time of arrival (ToA) of phonemes to identify spoofing attacks. Although VoiceLive does not require heavy computation, the detection accuracy largely depends on the distance between the speaker and the microphones. VoiceGesture [63] performs liveness detection by identifying human gestures from the microphone-sensed Doppler shifts. VoiceGesture is designed for smartphones, which cannot be directly applied for voice-controlled IoT devices due to its stringent requirement on the positions of devices' microphones. Recently, WiVo [33] uses wireless sensing to detect lip movements associated with the syllables in the voice command, which requires to place wireless antennas very close to the speaker. Tom et al. [51] achieve a significant reduction of errors in replay attack detection using an adaptation of the ResNet-18 model. Void [3] proposes a set of lightweight features in the audible spectrum to distinguish the voice source, and achieves low latency while maintaining relatively high detection accuracy. CaField [59] leverages the sound field characteristics to detect loudspeaker-based spoofing attacks.

Although the existing studies have achieved remarkable success in utilizing audible information of human voice, they either suffer from low accuracy on text-independent verification task, or require substantial computational resource usage. Different from all the previous approaches, the proposed SuperVoice aims to provide a more accurate and realistic SV solution using the high-frequency ultrasound components in the human voice.

Speaker Recognition Using High Frequency: The utilization of high-frequency components of human voice for speaker recognition has been studied before [17, 18, 34]. These studies, however, are lacking of crucial technical details necessary for designing a contemporary high-performance text-independent SV system.

7 CONCLUSION

In this paper, we initiate an exploration on the underexplored ultrasound voice components in human speech, and we find that they can be used to enhance the performance of speaker verification and liveness detection. We design a speaker verification system, SuperVoice, to show the strength of ultrasound frequency components in the speaker models. Specifically, we design a two-stream DNN structure to fuse the low-frequency and high-frequency features. SuperVoice significantly improves the speaker verification and liveness detection performance in comparison with the existing models. We further demonstrate the possibility of integrating ultrasound frequency features in the existing models to enhance their verification performance. SuperVoice is accurate, lightweight, and secure, which can be integrated into smartphones with a modification of smartphone's microphone component.

ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for providing valuable feedback on our work. This work was supported in part by National Science Foundation grants CNS-1950171 and CCF-2007159. Parts of the research reported in this publication was supported by the National Institute on Deafness and Other Communication Disorders, grant number R01DC012315. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Science Foundation and National Institutes of Health.

REFERENCES

[1] 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://catalog.ldc.upenn.edu/LDC93S1. Accessed: 2020-05-04.

[2] 2020. FFmpeg. https://www.ffmpeg.org/. Accessed: 2021-11-02.

[3] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In *USENIX Security*.

[4] Federico Alegre, Ravichander Vipperla, Nicholas Evans, and Benoït Fauve. 2012. On the vulnerability of automatic speaker recognition to spoofing attacks with artificial signals. In *2012 Proceedings of the 20th european signal processing conference (EUSIPCO)*. IEEE, 36–40.

[5] Avisoft. [n.d.]. http://www.avisoft.com/ultrasound-microphones/cm16-cmpa/.

[6] Avisoft. [n.d.]. http://www.avisoft.com/playback/vifa/.

[7] Bose. [n.d.]. https://www.bose.com/en_us/support/products/bose_speakers_support/bose_smarthome_speakers_support/soundtouch-10-wireless-system.html.

[8] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters* 13, 5 (2006), 308–311.

[9] Yu-hsin Chen, Ignacio Lopez-Moreno, Tara N Sainath, Mirkó Visontai, Raziel Alvarez, and Carolina Parada. 2015. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[10] Noam Chomsky and Morris Halle. 1968. The sound pattern of English. (1968).

[11] Anurag Chowdhury and Arun Ross. 2019. Fusing MFCC and LPC Features using 1D Triplet CNN for Speaker Recognition in Severely Degraded Audio Signals. *IEEE Transactions on Information Forensics and Security* (2019).

[12] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622* (2018).

[13] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on*

Audio, Speech, and Language Processing 19, 4 (2010), 788–798.

[14] Huan Feng, Kassem Fawaz, and Kang G Shin. 2017. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 343–355.

[15] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. 2005. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the SPECOM*, Vol. 1. 191–194.

[16] John S Garofolo. 1993. TIMIT acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993* (1993).

[17] Shoji Hayakawa and Fumitada Itakura. 1994. Text-dependent speaker recognition using the information in the higher frequency band. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1. IEEE, I–137.

[18] Shoji Hayakawa and Fumitada Itakura. 1995. The influence of noise on the speaker recognition performance using the higher frequency band. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. IEEE, 321–324.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[20] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. 2016. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5115–5119.

[21] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson. 2015. Speech acoustic modeling from raw multichannel waveforms. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4624–4628.

[22] HOTENDA. 2013. https://www.hotenda.com/datasheet-html/2493/1/SPU0410LR5H-QB.html.

[23] Roman Jakobson, C Gunnar Fant, and Morris Halle. 1951. Preliminaries to speech analysis: The distinctive features and their correlates. (1951).

[24] Artur Janicki, Federico Alegre, and Nicholas Evans. 2016. An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks. *Security and Communication Networks* 9, 15 (2016), 3030–3044.

[25] N. S. Jayant and Peter Noll. 1984. *Digital Coding of Waveforms, Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs NJ, USA, 688. N. S. Jayant: Bell Laboratories; ISBN 0-13-211913-7.

[26] Allard Jongman, Ratree Wayland, and Serena Wong. 2000. Acoustic characteristics of English fricatives. *The Journal of the Acoustical Society of America* 108, 3 (2000), 1252–1263.

[27] Patrick Kenny. 2005. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal,(Report) CRIM-06/08-13* 14 (2005), 28–29.

[28] Patrick Kenny, Themos Stafylakis, Pierre Ouellet, Md Jahangir Alam, and Pierre Dumouchel. 2013. PLDA for speaker verification with utterances of arbitrary duration. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 7649–7653.

[29] Tomi Kinnunen and Haizhou Li. 2010. An overview of text-independent speaker recognition: From features to supervectors. *Speech communication* 52, 1 (2010), 12–40.

[30] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. (2017).

[31] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. 2017. Audio Replay Attack Detection with Deep Learning Frameworks.. In *Interspeech*. 82–86.

[32] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[33] Yan Meng, Zichang Wang, Wei Zhang, Peilin Wu, Haojin Zhu, Xiaohui Liang, and Yao Liu. 2018. Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 81–90.

[34] Brian B Monson, Eric J Hunter, Andrew J Lotto, and Brad H Story. 2014. The perceptual significance of high-frequency energy in the human voice. *Frontiers in psychology* 5 (2014), 587.

[35] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2020. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language* 60 (2020), 101027.

[36] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612* (2017).

[37] Mahesh Kumar Nandwana, Julien van Hout, Mitchell McLaren, Allen R Stauffer, Colleen Richey, Aaron Lawson, and Martin Graciarena. 2018. Robust Speaker Recognition from Distant Speech under Real Reverberant Environments Using Speaker Embeddings.. In *Interspeech*. 1106–1110.

[38] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 1021–1028.

[39] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital signal processing* 10, 1-3 (2000), 19–41.

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

[41] Sada. [n.d.]. https://www.aliexpress.com/item/4001241222763.html.

[42] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. 2015. Learning the speech front-end with raw waveform CLDNNs. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[43] Martin F Schwartz. 1968. Identification of speaker sex from isolated, voiceless fricatives. *The Journal of the Acoustical Society of America* 43, 5 (1968), 1178–1179.

[44] Stephen Shum, Najim Dehak, Reda Dehak, and James R Glass. 2010. Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification.. In *Odyssey*. 16.

[45] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[46] SiSonic. [n.d.]. https://www.digikey.com/product-detail/en/knowles/SPU0410LR5H-QB-7/423-1139-1-ND/2420983.

[47] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. 2017. Deep Neural Network Embeddings for Text-Independent Speaker Verification.. In *Interspeech*. 999–1003.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[49] Marija Tabain. 2001. Variability in fricative production and spectra: Implications for the hyper-and hypo-and quantal theories of speech production. *Language and speech* 44, 1 (2001), 57–93.

[50] Siri Team. 2017. Hey siri: An on-device dnn-powered voice trigger for apple's personal assistant. *Apple Machine Learning Journal* 1, 6 (2017).

[51] Francis Tom, Mohit Jain, and Prasenjit Dey. 2018. End-To-End Audio Replay Attack Detection Using Deep Convolutional Networks with Attention.. In *Interspeech*. 681–685.

[52] usbank. 2020. How voice-activated devices work with banks. https://www.usbank.com/financialiq/manage-your-household/personal-finance/how-voice-activated-devices-work-with-banks.html.

[53] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE ICASSP*. IEEE, 4052–4056.

[54] Jesús Villalba and Eduardo Lleida. 2011. Preventing replay attacks on speaker verification systems. In *2011 Carnahan Conference on Security Technology*. IEEE, 1–8.

[55] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2018. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4879–4883.

[56] Vincent Wan and Steve Renals. 2005. Speaker verification using sequence discriminant support vector machines. *IEEE transactions on speech and audio processing* 13, 2 (2005), 203–210.

[57] WILDLIFE. [n.d.]. https://www.wildlifeacoustics.com/products/echo-meter-touch-2-pro-ios.

[58] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. 2015. Spoofing and countermeasures for speaker verification: A survey. *speech communication* 66 (2015), 130–153.

[59] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. 2019. The Catcher in the Field: A Fieldprint based Spoofing Detection for Text-Independent Speaker Verification. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1215–1229.

[60] Qiben Yan, Kehai Liu, Qin Zhou, Hanqing Guo, and Ning Zhang. 2020. SurfingAttack: Interactive Hidden Attack on Voice Assistants Using Ultrasonic Guided Wave. In *Network and Distributed Systems Security (NDSS) Symposium*.

[61] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[62] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible voice commands. In *Proceedings of the 2017 ACM CCS*. 103–117.

[63] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the 2017 ACM CCS*. 57–71.

[64] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM CCS*. 1080–1091.

APPENDIX

A SIGNAL PROCESSING

The Signal Processing module takes raw voice signals as input. It first preprocesses the raw signals to remove the recorded silence.

Then, it extracts ultrasound features from the original audio data, and downsamples the very high-quality audio to separate and extract low-frequency features.

Signal Preprocessing: The signal preprocessing removes the non-speech portions of the audio. Specifically, the silence periods within the speech are cropped to produce a clean speech signal. The original voice signal $s(t)$ is first segmented into K frames $s_i(t)$, $i \in [1, K]$ (typically a segmentation length K is 800 samples for 16 kHz data). To crop the silence periods in $s_i(t)$, we initialize a *silence period counter* $f(s_i(t))$ to count the number of contiguous silence periods:

$$f(s_i(t)) = \begin{cases} 0, & \text{if } \overline{(s_i(t))^2} > \theta \\ f(s_i(t)) + 1, & \text{if } \overline{(s_i(t))^2} \leq \theta, \end{cases} \tag{7}$$

$$\theta = min(\overline{(s_i(t))^2}) + (max(\overline{(s_i(t))^2}) - min(\overline{(s_i(t))^2})) * \theta_{ratio}.$$

If the mean square power $\overline{(s_i(t))^2}$ of the current segment $s_i(t)$ is less than a pre-defined threshold θ, this segment is marked as a silence period, and $f(s_i(t))$ will be incremented. We configure a constant number c to indicate the toleration of contiguous silence frames. Segment $s_i(t)$ will be discarded if $f(s_i(t)) > c$, i.e., if all the previous c frames are regarded as silence periods. The threshold in Eq. (7) depends on the power of the current frame and a constant value θ_{ratio}, which specifies the percentile of the power distribution that we consider as background noise. Here, we empirically set the θ_{ratio} as 0.25. The tolerance number c is set to 25, which means that if there are contiguous 25 frames below the threshold, the upcoming frames will be discarded until a higher power frame arrives, where the higher power frame refers to a frame with its average power exceeding θ.

Ultrasound Feature Extraction: After the removal of the silence periods, we obtain the clean speech $s'(t)$, and then we perform STFT to convert the sound waveform into STFT spectrogram, which is a common approach adopted by the existing SV systems [20, 35, 56]. Specifically, $s'(t)$ is first segmented into time frames of 2,048 samples, and then STFT results are computed using a Hann window with 1/4 window length. We convert the amplitude spectrum to dB-scale. The FFT length is set as 2,048, while the sampling rate is 192 kHz, and the frequency resolution is 192 kHz$/2,048 = 93.75$ Hz. In the end, the feature extraction produces a matrix of STFT features with the size of $1,024 \times L$, and L can be expressed as follows:

$$L = \frac{len(s'(t)) - win_{STFT}}{hop_{STFT}} + 1, \tag{8}$$

where $win_{STFT} = 2,048$, $hop_{STFT} = 512$. The STFT spectrogram can be divided into different frequency ranges. The ultrasound components between 24-60 kHz are used for liveness detection, while the components across the range of 8-96 kHz are adopted for speaker verification.

Downsampling: Besides the high-frequency components, we also extract the low frequency components as a separate input to the SV system. To do that, we perform a downsampling procedure using FFmpeg [2] tool to reduce the sampling rate of clean speech $s'(t)$ from 192 to 16 kHz, and therefore obtain low frequency audio ranging in the interval $[0, 8]$ kHz. The downsampled data contains the essential low-frequency components for SV [15, 54].

B NEURAL NETWORK IMPLEMENTATION DETAILS

CNN-1: The CNNs are utilized to extract low-frequency features. As the input of low-frequency feature is raw signals, we choose the Sinc-based convolutional layer [38]. The Sinc-based convolutional network has demonstrated high efficiency and accuracy on speaker verification and recognition tasks because it leverages interpretable filters with less network parameters. In our implementation, Sinc-based convolutional layer uses 80 filters with length $L = 251$. The Sync layer is followed by 2 standard one-dimensional convolutional layers, both using 60 filters of length 5. We also apply a layer of normalization for all the convolutional layers, and use the LeakyRelu function at every hidden layer to ensure the non-linearity of the CNNs model. The output dimension of the CNNs is 6420×1.

CNN-2: This network takes high frequency cropped matrix as input and generates a high level feature of the high-frequency components. It contains 8 layers, the first two layers include 64 F-Filters with size of 9×1 and dilated by 1×1 and 2×1 respectively. Followed by two horizontal layers, it contains 64 T-Filters with size of 1×9 and dilated by 1×1 and 1×2. The next 4 layers are composed by square F/T filters, each layer has 48 filters with size of 5×5, and dilated in various scale 2×2, 4×4, 8×8 and 16×16. The 8 layers are followed by an average pooling layer that produces a feature map with 512×1 dimension.

NN: The last NN model is used to produce speaker embeddings using the fused features from CNNs-1 and CNNs-2. It is a fully connected layer that takes features with a dimension of 6420+512×1 and output 2048×1 dimension as the speaker embedding. Note that during the training procedure, the NN model also contains another fully connected layer and a softmax classifier, such that it is able to map 2048×1 dimension to *class*×1 to process the speaker recognition task, where *class* is the number of speakers in the training pool.

Hyperparameters: In the training stage, we use RMSprop optimizer, with a learning rate $lr = 0.001$, $\alpha = 0.95$, $\epsilon = 10^{-7}$, and the default minibatch is 128.

C FEATURE ALIGNMENT

One of the key challenges in SuperVoice is to generate the input features from the low-frequency raw audio and high-frequency spectrum simultaneously. For this purpose, the CNN-1 and CNN-2 models require the aligned inputs of different formats (i.e., raw audio versus spectrum data). Given an incoming audio, we use one window to generate low-frequency raw audio frames, and another window to collect the windowed STFT spectrum data to be fit into the models. The window used for CNN-2 model matches the low-frequency window of the CNN model. In this paper, we set the window size of the low-frequency data L_{win} to 200 ms, corresponding to 200 ms × 16 kHz = $3,200$ samples, and the hop size L_{hop} to 10 ms with 10 ms × 16 kHz = 160 samples. Then, the high-frequency window size H_{win} and hop size H_{hop} can be computed as follows:

$$\begin{aligned} H_{win} &= L_{win} \times \alpha / hop_{STFT}, \\ H_{hop} &= L_{hop} \times \alpha / hop_{STFT}, \end{aligned} \tag{9}$$

where hop_{STFT} is the hop length of STFT spectrum (i.e., 512 in this paper), and α is the ratio of high and low sampling rates. Here, the high and low sampling rates are 192 kHz and 16 kHz, respectively, i.e., $\alpha = 12$. hop_{STFT} is used to convert the number of samples in the low-frequency components into the number of columns in the STFT matrix of the corresponding high-frequency components. This window alignment scheme guarantees that these two types of features corresponding to the same original audio input are generated at the same time. Therefore, different types of input features, including both the low-frequency voice feature and ultrasound voice feature, can be fused together to form a coherent feature vector.

D DATASET COLLECTION AND COMPOSITION

Dataset Collection Routine: To formulate our ultrasound dataset (e.g. Voice-1 and Voice-2), we recruit volunteers as many as we can by spreading information via social media, public Facebook page, and posters on information boards. We provide each volunteer $5 gift card for compensation. Here are the procedures we follow for each volunteer to conduct our experiment.

Step 1: Schedule a remote instruction session with each volunteer.

Step 2: Prepare scripts for the volunteer and disinfection on microphones and door handles.

Step 3: Instruct the volunteer outside the data collection room (e.g. getting familiar with the user interface).

Step 4: Register the basic information of the volunteer.

Step 5: Conduct the voice recording experiment.

Types of Sentences: There are 4 types of sentences we designed.
1. Common type: The common type contains 5 sentences, two of which are from the dialect set of TIMIT dataset [1], while the other three sentences are "This is a test", "Please identify me", and "OK, please recognize my voice". For this type of sentence, each participant will speak every sentence twice, resulting in 10 recordings.
2. Compact type: The compact type has 200 sentences in total. They are randomly chosen from the TIMIT dataset, and designed to provide a good coverage of phonemes, with extra occurrences of phonetic contexts that are considered either difficult or of particular interest. Each participant reads 40 sentences from this type.
3. Fricative command: The fricative commands are collected from the website ok-google.io, which provides commonly used commands on Google Assistant. We select 50 sentences that have fricative phonemes, e.g., "find my phone", "turn on my lights". Each speaker randomly picks 25 sentences from this type of commands.
4. Non-fricative command: Similar to the fricative commands, the non-fricative commands are also collected from ok-google.io, which do not contain any fricatives. We randomly pick 25 sentences and ask every participant to read 25 sentences from this type.

Table 6: Training and testing pools (# of sentences).

Dataset	Training pool		Testing pool	
	Training	*Valid.*	*Enroll*	*Verif.*
Voice-1	2,400	600	144	4656
Voice-2	-	-	75	1,175
Voice-3	-	-	60	140

Table 6 shows the breakdown of training and testing pools for Voice-1 and Voice-2. For Voice-1, 30 participants are randomly selected into the training pool, and their recordings are then divided into the training and validation sets as described in Table 6. For participants in the training pool, their audios in the training set are used to train the SuperVoice model, while audios in validation set are applied to validate the training process every 10 epochs. As for those in testing pool, the recordings are split into the enrollment and verification sets. In the testing procedure, SuperVoice loads the trained model and uses the enrollment utterances to generate the speakers' embedding, and then it produces the embedding and calculates the similarity to make SV decision (i.e., accept or reject). All participants' voice data in Voice-2 are only used for testing. For Voice-2, we used 7 smartphones: iPhone X, iPhone XS, iPhone 8, iPhone Pro Max (Apple), Mi 9 (Xiaomi), and Mate 20 (Huawei). For Voice-3, we recruit 20 volunteers to speak 10 sentences. Then, we use 3 audios from every speaker as enrollment, and the rest of audios are used to verify the enrolled volunteers.