Secure Voice Input on Augmented Reality Headsets

Jiacheng Shang and Jie Wu, *Fellow, IEEE*

Abstract—Voice-based input is usually used as the primary input method for augmented reality (AR) headsets due to immersive AR experience and good recognition performance. However, recent researches show that attackers can inject inaudible voice commands to the devices that lack voice verification. Even if we secure voice input with voice verification techniques, attackers can record the victim's voice and replay it. To defend against voice-spoofing attacks, AR headsets should be able to determine whether the voice is from the person who is using the AR headsets. Existing voice-spoofing defense systems are designed for smartphone platforms and usually fail to work due to the special locations of microphones and loudspeakers on AR headsets. To address this challenge, in this paper, we propose a voice-spoofing defense system for AR headsets by leveraging both the internal body propagation and the air propagation of human voices. Experimental results show that our system can successfully accept normal users with average accuracy of 97% and defend against two basic types of attacks with average accuracy of at least 98%. More importantly, even if the attackers can fool our line-fitting model by manipulating special voice signals, our MCD-SVDD model can still reject them with accuracy of 100%.

Index Terms—AR headsets, voice spoofing attack, liveness detection.

1 Introduction

Augmented reality (AR) applications that overlay a user's perception of the real world with digitally generated information are on the cusp of commercial viability. To provide better user experiences, AR experiences are primarily delivered to AR users via wearable glass devices and head-mounted devices. For example, Microsoft, Google Vuzix, and other companies have been working on bringing AR to us in the eyeglass form. Moreover, different from traditional human-computer interactions, most existing interactivity technologies (e.g. typing, tapping, clicking, and swiping) have become irrelevant and obsolete in the AR world. Because of the real-world interaction of AR experiences, the input methods for AR headsets should fit what a human can understand. Therefore, most AR headsets adopt voice, eye gaze, and gestures as input methods. Among these three input methods, voice-based input is usually used as the primary input method for three reasons: 1) Voice is the primary way to deliver information in daily life, so voice-based input can provide immersive AR experiences; 2) Many low-cost AR devices do not have capabilities to track eye gaze and recognize gestures; 3) Most gesture and gaze interfaces have problems with responsiveness and accuracy.

However, voice-based input suffers from various voice spoofing attacks. Recent research [7], [29], [34] has shown that an attacker can inject inaudible voice commands to the devices that lack voice verification. Moreover, unlike other human biometrics, the human voice is often exposed to the public in many different scenarios, e.g., people making a presentation in public. Even if we secure devices with voice verification techniques, an attacker can easily steal the victim's voice using low-cast handy recorders and attack voice-based applications with the help of state-of-the-art voice synthesis/conversion software. Several security issues are, therefore, caused by the leakage of people's voices and pose a severe threat to voice-based applications [18], [26], [33]. For instance, with a replay device, an adversary could impersonate the victim to spoof the Google Trusted Voice once they acquire enough victim's voice samples. Since voice is considered as a unique biometric of a person, these voice-spoofing attacks would result in severe consequences harmful to victim's safety, reputation, and property.

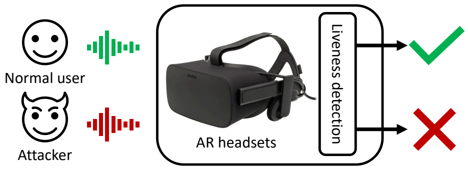


Fig. 1. The input voice of an AR headset can be from the normal user or attackers.

To defend against voice-spoofing attacks, the voice-based systems need to determine whether the voice is from the person who is using the AR headsets. To achieve this goal, traditional systems primarily use two solutions: 1) Check the channel noises introduced by recording and the replay devices (loudspeakers); 2) Analyze the reverberation of replaying far-field recordings. However, these solutions have high false acceptance rates of up to 17% [36], which makes them unsuitable to be used for commercial systems. Recently, many liveness detection systems have been proposed to fight against voice-spoofing attacks by studying the differences between the human vocal system and loudspeakers using phoneme location [36], articulatory gestures [35], magnetic fields of loudspeakers [9], and throat voice [24]. However, all of them are designed for smart-

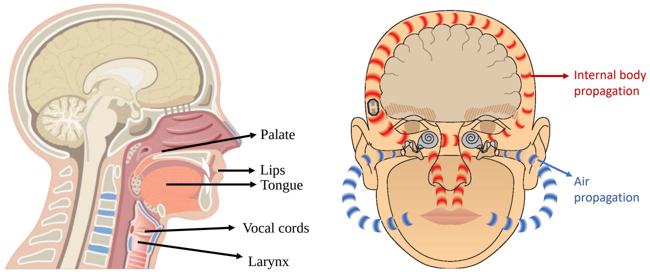
- *J. Shang is with Department of Computer Science, Montclair State University, 1 Normal Ave, Montclair, NJ 07043.*
- *J. Wu is with Department of Computer and Information Sciences, Temple University, 1805 N. Broad Street, Philadelphia, PA 19122.*

phones. Considering the special locations of microphones and loudspeakers on AR devices, current liveness detection solutions cannot be implemented on AR headsets. For example, the approach proposed in [35] can fight replay attacks by reusing a pair of microphone and loudspeaker as a Doppler radar. However, this system requires both the loudspeaker and the microphone to be in front of the user's mouth during speech, which is hard to be ensured on AR headsets.

Considering the limitations of current solutions, we propose a voice-spoofing defense system for AR headsets by leveraging the internal body propagation of human voices. Our system determines whether the voice is from the person who is using the AR headsets by leveraging: 1) Both the internal body propagation and the air propagation of human voices; 2) A tiny and low-cost contact microphone to collect internal body voice. First, human voices propagate through both the air and the internal body (skull). If two voices are from the same person, they should share common features in the frequency bands of human voices. Second, by attaching a contact microphone on the user's head, we are able to collect the voice propagating only through the internal body. The small contact microphone can be easily integrated into existing AR headsets. To achieve our goal, we solve two challenges in the design of our system. First, the signal-to-noise ratio (SNR) of the voice that propagates through the internal body is still low, which makes it hard to extract voice features from the raw time-domain signals. To address this issue, we transform the signal from the time domain to the time-frequency domain and leverage spectrogram enhancement techniques to extract the voice from raw signals. The second challenge is to measure the correlation and similarity between the internal body voice and the air voice of the user. In order to robustly measure the correlation and similarity between the two voices, we match high-energy blocks that exist in both spectrograms of two voices. Compared with existing works, our system has three major advantages. First, without changing the framework of current headsets, our system can be directly implemented on headset devices by attaching a low-cost and tiny contact microphone. Second, our system does not need to collect any data from attackers to build a classifier. Third, our sytem does not need users' extra effort in operating the AR headsets, e.g., moving the device around the audio source.

We summarize our contributions as follows:

- We show it is feasible to capture the internal body propagation of human voices using a low-cost contact microphone. We also present an approach to extract voice features from noisy internal body voice.
- We propose a robust and low-cost solution for defending against voice-spoofing attacks on AR headsets with high accuracy. To the best of our knowledge, our system is the first to protect the voice input for AR headsets.
- We propose two different classification models with different computation cost. The low-cost line-fitting model can effectively defend against obstruction attacks and replay attacks. The MCD-SVDD model can further reject expert attacks with limited cost added.
- Our classification models do not need to collect any data from the attackers, which means our system can be quickly launched for a new user.
- We develop a prototype and conduct comprehensive evaluations. Experimental results show that our system can successfully defend against obstruction and replay attacks with an accuracy of at least 98%. More importantly, even if the attackers can fool our line-fitting model by manipulating special voice signals, our MCD-SVDD model model can still reject them with accuracy of 100%.



(a) The human vocal system. (b) Two propagation paths.

Fig. 2. Human vocal system and two propagation paths of the voice.

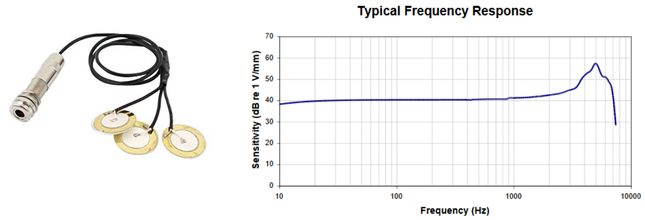
The remainder of this paper expands on the above contributions. We first introduce our attack model and key insights in Section 2 and present our solutions in Sections 3, 4, and 5. We conduct various experiments to evaluate proposed solutions in Section 6 and discuss the usability and limitations of our system and related work in Sections 7 and 8, respectively.

2 Preliminary

In this section, we discuss the human voice production and propagation system and the two types of attacks we consider in this paper. Based on our preliminary experiments, we show two key observations that enable us to defend against voice-spoofing attacks on AR headsets.

2.1 Human voice production and propagation

In order to achieve robust liveness detection, we need to understand the structural differences between the human vocal system and loudspeakers. As shown in Fig. 2(a), the mechanism for producing the human voice can generally be divided into three parts: the lungs, the vocal cords, and the articulators (e.g. lips and tongue). The lung first produces adequate airflow and air pressure to vibrate vocal cords. The vocal cords vibrate and chop up the airflow from the lungs into audible pulses that form the laryngeal sound source. Then, the length and tension of the vocal cords are adjusted to produce 'fine-tune' pitch and tone. The articulators consisting of tongue, palate, cheek, and lips further filter the sound generated from the larynx to strengthen it or weaken it. After the voices are produced by the human vocal system, they mainly propagate through two media, as shown in Fig. 2(b). First, the voice propagates via the air and reaches the microphone, which is common for the use case of current voice input. Besides propagating through the air, the voice can also propagate through the speaker's internal body, and



(a) Contact microphone. (b) Frequency response of contact microphone [1].

Fig. 3. Contact microphone and its frequency response.

that is why a person's voice sounds different to them when it is recorded and played back. Although the tone of the voice received through the internal body is lower than that of the voice received through the air due to the special propagation medium, the two voices should have a strong correlation and share a lot of information. For the attacker who wants to issue a fake voice command in order to obstruct the victim's experience, the attacker's voice reaches the AR device only through the air. Therefore, the internal body voice of the victim should not have much-shared information with the air voice.

Strong attackers can also use high-quality loudspeakers and recorders to break voice-based authentication. The loudspeakers usually use an electromagnet to translate an electrical signal into an audible sound. The electromagnet is a metal coil that creates a magnetic field when there is an electric current flow through it. When electrical pulses pass through the coil of the electromagnet, the direction of the magnetic field is frequently changed. Also, there is a permanent magnet fixed firmly into the loudspeaker. With a rapidly changing magnetic field, the coil is attracted to and repelled from the permanent magnet. As a result, the cone attached on the coil will vibrate back and forth, pumping sound waves into the surrounding air and the smartphone's speaker. Since the replay attacker can only record and replay the air voice of the victim, there is no internal body voice during the replay process. Moreover, since the internal body voice of a person is different from those of others even for the same word, a stronger replay attacker cannot impersonate the victim's internal body voice by wearing the AR headset and saying the same words.

2.2 Piezo contact microphone

As shown in Fig. 3(a), contact microphone is a form of microphone that senses audio vibrations through contact with solid objects. Unlike normal air microphones, contact microphones are almost completely insensitive to air vibrations but transduce only structure-borne sound. Crystals in Piezo contact microphones which demonstrate the piezoelectric effect produce voltages when they are deformed. The crystal microphone uses a thin strip of piezoelectric material attached to a diaphragm. The two sides of the crystal acquire opposite charges when the crystal is deflected by the diaphragm. The charges are proportional to the amount of deformation and disappear when the stress on the crystal disappears. By attaching a contact microphone near the speaker's temple, we are able to collect the voice that propagates mainly through the body of the speaker. In addition, contact microphones have a wide frequency response, as shown in Fig. 3(b). Since the voiced speech of a typical adult will have a fundamental frequency for up to 255Hz [3], the contact microphones have enough capability to capture the internal body voice.

2.3 Attack model

In our attack models, a malicious user aims to either spoof the voice verification system on the AR headset or obstruct the normal use of voice-based input. The capability of the attacker is limited in the sense of:

2.3.1 Obstruction attack for voice commands

In an obstruction attack, a malicious user who can show up closely around the normal user aims to issue a voice command with high volume. For example, the malicious user can issue a "remove" voice command to clear the victim's virtual objects. The malicious user can also issue a voice command to display redundant information in the field of vision of the normal user, which poses threats if the normal user needs clear sight (e.g., when the normal user is driving). In fact, this type of attack is feasible in practice for three reasons. First, to defend against such attacks, the device should know what signals are environmental noises. Most existing headsets (e.g. Microsoft Hololens) solve this problem by using a directional microphone to collect only the user's voice. However, a recent user study shows that the Microsoft Hololens can still pick up environmental noise as the voice commands sometimes, which means that attackers still have chances to launch such attacks. Second, recent research has shown that the attacker can even issue malicious voice commands on inaudible channels. By doing this, the legitimate user cannot notice the existence of such attacks. Third, even if the legitimate user can hear the malicious voice command, it is too late in some cases since the device has already picked it up and follows the voice commands.

2.3.2 Replay attack for voice-based authentication

In this type of attack, we assume that an attacker can physically access the victim's headset without being noticed. Moreover, the attacker can record the victim's voice and replay it to voice-based authentication system using loudspeakers. To achieve better attack performance, we assume that the attack can produce the corresponding internal body voice by shadowing the replayed voice of the victim.

2.4 Use case

In order to successfully defend AR users against the two types of attacks, our system requires users to attach a contact microphone around the temple. Since the AR users need to wear the AR headset, this condition can be easily satisfied by integrating the contact microphone into the frame of the AR headset. We leverage the contact microphone to capture the internal body voice and use the existing normal microphone on current AR devices to collect the air voice. The distance between the normal microphone and the user's mouth is about 10 centimeters. Since the distance is pretty short, the

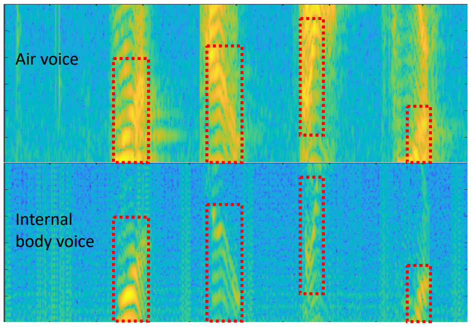


Fig. 4. The spectrograms of voices through air and internal body.

time delay between two audio signals is less than 13 samples when the sampling rate is 44,100 samples per second. While speaking, the user can be in any stationary posture, such as sitting and standing.

2.5 Feasibility study and challenges

In order to defend against the two attacks we consider, we need to fully leverage the relationship between voices through the air and the skull. Fig. 4 shows the spectrograms of two voices when the user says "Five." We can observe two facts: 1) There exists a strong correlation between two voices on both the time and frequency domains. If a normal user interacts with the headset using voices, we should observe that a voice through the internal body is produced at the same time. 2) The voice that propagates through the internal body only reserves partial low-frequency features (200 Hz to 2000 Hz). If we can see high-energy blocks in the spectrogram of internal body voice, we should see high-energy blocks at the same location in the spectrogram of the air voice. These observations illustrate that it is feasible to defend against two attacks by measuring the correlation and similarity between two voices.

To achieve our goal, we solve two challenges in the design of our system. First, even with amplifier, the SNR of the voice that propagates through the internal body is still low, which makes it hard to extract voice features from the raw time-domain signals. To address this issue, we transform the signal from the time domain to the time-frequency domain and leverage spectrogram enhancement techniques to extract the features of two voices from their raw signals.

The second challenge is to measure the correlation and the similarity between the internal body voice and the air voice. This is difficult because both voices have different capabilities for capturing users' voices. More specifically, the internal body voice only contains partial low-frequency features, but it is nonsensitive to environmental noise. The mouth voice reserves much more features, but it is easily influenced by environmental noise. In order to robustly measure the correlation between two voices, we first convert the two voices to spectrograms on the time-frequency domain of the three dimensions: time, frequency, and energy. The correlation and the similarity of two voices are measured by matching high-energy blocks that exist in both spectrograms.

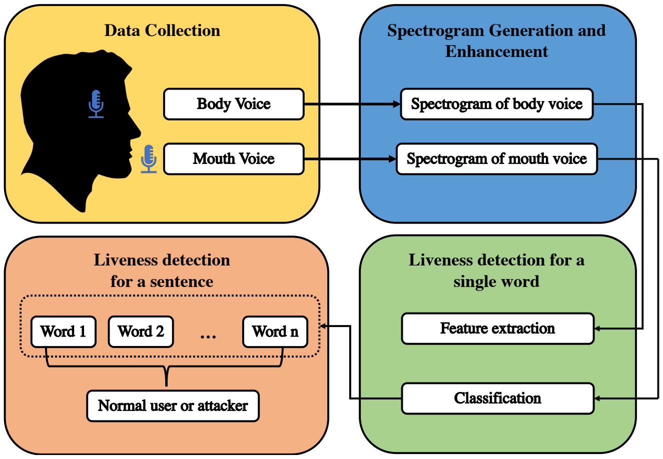

Data Collection
Body Voice
Mouth Voice
Spectrogram Generation and Enhancement
Spectrogram of body voice
Spectrogram of mouth voice
Liveness detection for a sentence
Word 1
Word 2
...
Word n
Normal user or attacker
Liveness detection for a single word
Feature extraction
Classification

Fig. 5. System pipeline.

3 System Design

In this section, we show the pipeline of our system and describe our solutions in detail.

3.1 System overview

The key idea underlying our system is to fully leverage two propagation paths of the human voices. When the AR user says a voice command, the normal microphone will capture the user's voice that propagates through the air, and the contact microphone on the user's head can record the voice that only propagates through the user's body. By comparing the information in the two voices, our system can determine whether the voice is from the normal user or from two types of attackers. For a new AR user, there are two stages to use the system. In the training stage, the new user is asked to say a few words using our system. These training instances are used to quickly build a classifier. After the training stage, the system is ready to be used. In the testing stage, our system will check whether the command is from the normal user who is using the AR headset using the trained classifier. If the voice is from the normal user, the user can interact with the AR headset normally. Otherwise, the voice command will not be parsed to the AR headset for further verification.

The pipeline of data collection and processing is shown in Fig. 5. After collecting the user's voices at two channels, we first segment the voice for each word to remove the internal between neighboring words. For the voice signals of each pair of words, we transform the signals from the time domain to the time-frequency domain. Since both raw voice signals contain background noise, we further leverage spectrogram enhancement techniques to remove the noise and extract the information of the voices. Then, we measure the correlation between two enhanced spectrograms of each pair of words. If the correlation exceeds a threshold, the pair of signals is further checked for the second round. In the second round, we measure the similarity between two spectrograms. Here the similarity is defined as the proportion of shared information between two voices. If the proportions of shared information fit the trained classifier, the word is regarded to be from the normal user. To tolerate wrong classification results, the final detection result of a sentence (voice command) is determined by a voting procedure of all words in it. Only if the number of votes that represent the voices from the normal user exceeds the voting threshold, is the voice source regarded as the normal user.

3.2 Word segmentation and spectrogram generation

Each audio signal includes two parts: the voice and background noise. The voice contains abundant features of the user's voice, while the noise part only records the acoustic noise in the background. In our system, we only focus on the user's voice in order to reduce the influence of the acoustic noise in the background. Since the voice recorded by the normal microphone has much more features of the user's air voice, we segment each audio sample into different words by performing Hidden Markov Model (HMM) based word segmentation techniques [23] on the air voice.

Also, we need to find features to measure the relationship and differences between two voices collected from two microphones to distinguish whether the voice is from a normal user. In order to capture features on time-frequency domain, we perform short-time Fourier transform (STFT) on each word and each audio sample with a window size of about 22 ms based on:

$$X(\tau,\omega)=\sum_{n=t_s}^{n=t_e} x[n]w[n-\tau]e^{-j\omega n} \tag{1}$$

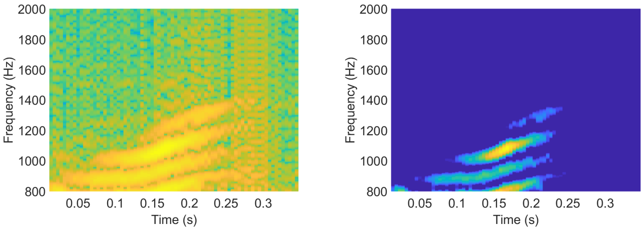
where τ is the time axis, ω is the frequency axis, $x[n]$ is an audio signal in the time range (t_s, t_e), $w[n]$ is the window, and $X(\tau,\omega)$ is a complex function representing the phase and magnitude of the signal over time and frequency. Then, for each time and frequency frame, the spectrogram of the complex function $X(\tau,\omega)$ is computed based on:

$$E[f,t]=|X(\tau,\omega)|^2 \tag{2}$$

where $E[f,t]$ is the power of f^{th} frequency band and t^{th} time frame. f and t are positive integers with range $1 \leq f \leq M$ and $1 \leq t \leq N$. M is the number of frequency frames in generated spectrogram, and N is the number of time frames. In our system, we further convert power measurements to decibels for data processing.

3.3 Spectrogram enhancement

In real usage scenarios, the contact microphone cannot touch the skull directly, which leads to low SNR of recorded internal body voice even with an amplifier. The air voice is also influenced by background noise. To extract features from both voices, we leverage spectrogram enhancement techniques to extract high-energy clusters that are only produced by the user's voice on the generated spectrograms. After obtaining the spectrogram of each word, we first apply the frequency domain denoising method by subtracting the noise floor (non-voice part) from the spectrogram. Since the microphone of the AR headset is close to the user's mouth, most power should be distributed on the voice part as shown in Fig. 6(a). Therefore, the noise floor is set to 80% of the power in the spectrogram of each word. If the resulting magnitude becomes negative after subtraction, we set it to zero. Second, since the internal body voice collected from the contact microphone contains strong noise under 800 Hz, we only reserve the spectrograms from 800 Hz to 2000 Hz for the following analysis. As shown in Fig. 6, most of the noise is removed from the spectrogram, and only the information of the voice is reserved.



(a) Raw internal body voice.
(b) Enhenced spectrogram.

Fig. 6. Spectrogram enhancement.

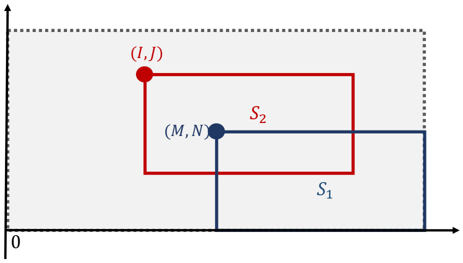


Fig. 7. Illustration of correlation calculation.

3.4 Feature extraction

Since two voices are generated from the same vocal system at the same time, we should be able to observe strong correlations between them for a normal user. Ideally, the subtraction of two spectrograms should be zero. In our system, we measure the correlation between two spectrograms instead of directly calculating the differences between them for two reasons. First, both voices have different capabilities for capturing users' voices. More specifically, the internal body voice only contains partial low-frequency features, but it is nonsensitive to environmental noise. The mouth voice reserves much more features, but it is easy to be influenced by environmental noise. Second, even if two microphones are synchronized, there may still exist small synchronization bias in the collected voices. In our solutions, we consider the two spectrograms (S_1 and S_2) as two pictures in two-dimensional hyperplane, as shown in Fig. 7. Both S_1 and S_2 are of the same size ($M \times N$). If we fix the position of S_1 and move S_2, two spectrograms must have overlapped area as long as the point (I,J) is within the gray area. Similar to one-dimension cross-correlation measurement, given two spectrograms S_1 and S_2, we measure the correlation between S_1 and lagged copies of S_2 as a function of i and j. For this copy, if we assume that S_1 and the lagged copies of S_2 have an overlapped area of size $M' \times N'$, the correlation coefficient of the specific shift is:

$$Corr[i,j]=\sum_{k=1,\,l=1}^{k=M',\,l=N'} O_1^{i,j}[k,l]\times O_2^{i,j}[k,l] \tag{3}$$

where O_1 is the overlapped part of S_1, and O_2 is the overlapped part of S_2. The point (i,j) represents the pos-

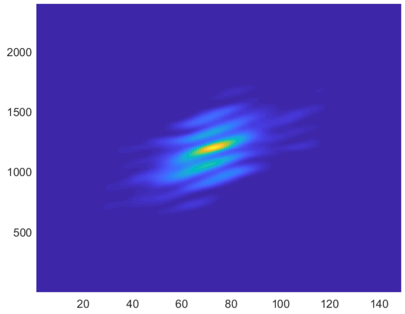


Fig. 8. Correlation matrix for two voices from the same user.

sible postions of (I, J). Hence, the positive integer i is from 1 to $2M - 1$, and the positive integer j is from 1 to $2N - 1$. The best matching of two spectrograms is found if the corresponding correlation coefficient is maximal. In our system, two voices are highly correlated, so the highest correlation coefficient must appear around the center of correlation matrix $Corr$, as shown in Fig. 8. Based on this observation, a word is detected to be from a live user if

$$\frac{|J - M|}{2M} < \lambda \quad and \quad \frac{|I - N|}{2N} < \lambda \tag{4}$$

where λ is the decision threshold.

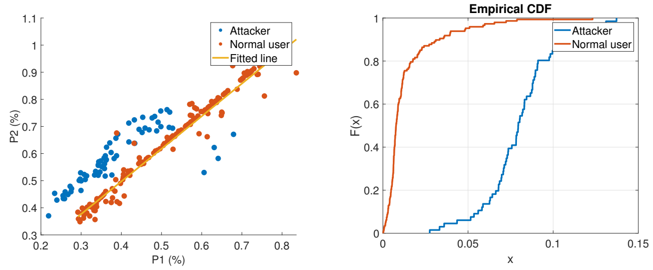
A pair of spectrograms that satisfy Equation 4 cannot ensure that two voices are from the normal user. Although we know the two spectrograms are highly correlated from Equation 4, it is not clear how much information or features are shared between two spectrograms. Therefore, we further measure the similarity between two voices by finding the proportion of shared information. Based on our observations, the amount of shared information should make up a large proportion of either of two voices. In other words, if an entry is non-zero in the spectrogram of internal body voice, it is very likely to be non-zero in that of the mouth voice, and vice versa. To quantitatively describe how similar two spectrograms are, we first calculate the lags using the values of i, j, M, and N. After that, we use the measured lags to calibrate our synchronization to get the best match. For each word, the proportion of the shared information that is in S_1 is defined as:

$$P_1 = \frac{Sizeof(\{(i,j)|S_1[i,j] > 0 \;\&\; S_2[i,j] > 0\})}{Sizeof(\{(i,j)|S_1[i,j] > 0\})} \tag{5}$$

Similarly, the proportion of the shared information that is in S_2 is defined as:

$$P_2 = \frac{Sizeof(\{(i,j)|S_1[i,j] > 0 \;\&\; S_2[i,j] > 0\})}{Sizeof(\{(i,j)|S_2[i,j] > 0\})} \tag{6}$$

The similarity between two voices is defined as the smaller one of P_1 and P_2. The reason why we use this fraction expression is that we want the final results of P_1 and P_2 to be a ratio. The ratio representation is necessary because this can eliminate the negative impacts of absolute values. Since the energy distribution of different words is diverse, the absolute values of $sizeof((i,j)|S1[i,j] > 0)$ and $sizeof((i,j)|S2[i,j] > 0)$ vary in a wide range. If we feed this absolute range to the classifiers, the classification performance will degrade a lot since the features are not well normalized. Therefore, the ratio representation here is a form of normalization.



(a) Feature distribution.

(b) Distance distribution.

Fig. 9. Feature analysis.

4 LINE-FITTING MODEL

4.1 Liveness detection for a single word

Fig. 9(a) shows the values of the proportion of the shared information for both the normal user and the attacker. Ideally, the proportion of the shared information should be high for normal users. However, since different users have different speaking habits (e.g. different speeds of speech and different accents), the proportions of shared information may not always be a high value. Also, unpredictable noise during data collection may also influence the final results. Therefore, it is hard to determine the legitimacy of the speaker using a fixed threshold on each dimension. By studying the data distribution on a 2-dimension feature hyperplane, we find that data of normal users lies on a straight line, while that of attackers is far away from the line. Fig. 9(b) shows the distribution of distances from the data point to the straight line that is fitted using the normal user's training data. We can see that over 95% of the normal user's data points have a distance less than 2, while over 85% of the attacker's data points have a distance larger than 2. This fact enables us to detect the legitimacy of the speaker by calculating the distance from the data point to the line that fits the training data. After collecting several training data from the user, we first fit a straight line using least squares, as the yellow line in Fig. 9(a). A word is considered to be from the normal user if

$$\frac{|aP_1 + bP_2 + c|}{\sqrt{a^2 + b^2}} < \gamma \tag{7}$$

where P_1 and P_2 are the features calculated using Equations 5 and 6, and a, b, and c are coefficients of a straight line $ax + by + c = 0$. γ is the decision threshold and is set to the 95% largest distance of normal user's training data. A word is considered to be from a normal user if and only if both of Equations 4 and 7 are satisfied.

4.2 Liveness detection for a passphrase

AR users usually speak a sentence or passphrase that consists of multiple words to AR headsets. For example, the general voice authentication systems ask the user to speak a 6-digit passphrase. In order to give an accurate detection result for each sentence, we need to combine the results of

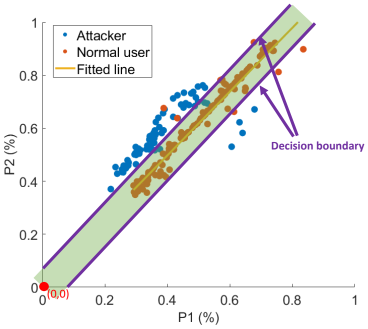


Fig. 10. A special case where an attacker can still break the system without having the victim's internal body voice.

multiple words after getting the correlation and similarity measurement of each of them. In a voting procedure, three questions need to be answered. 1)Who should be eliminated from voting; 2)What is the weight of each player; 3) What is the minimum number of votes needed to pass a vote? To answer the first question, the voter whose data cannot satisfy either of Equations 4 or 7 is eliminated from voting. Second, since both P_1 and P_2 reflect the propagations of shared information between two voices, the word with high values of P_1 and P_2 should have a higher weight for voting. Therefore, for each word in the voting procedure, we let the smaller value of its P_1 and P_2 be its weight. Third, to accurately reject the attacker and accept the normal user, for a sentence or a voice command with n words, the minimal number of votes is set to $0.2 \times n$. If there is no result whose number of votes exceeds $0.2 \times n$, the user is regarded as the attacker.

5 Extensions

In this section, we will state the limitations of line-fitting model. To address these limitations, we propose a new classification model by combining the results of two weak classifiers. Finally, we will analyze the computation cost of the new model.

5.1 Problem statement

Although the line-fitting model proposed in Section 4 can recognize two types of attackers with high accuracy, it can still be broken if the attacker knows all details (e.g. feature extraction and classification methods) of the liveness detection system. The reason is that the decision boundary of the line-fitting model covers much larger areas than those of the training data on feature hyperplane. Fig. 10 shows an example where the attacker can easily break the line-fitting model without having the victim's internal body voice. Assuming that the attacker's data has satisfied the Equation 4, then the attacker can carefully design the internal body voice in advance so that the values of both P_1 and P_2 close to zero. In this case, the point of the attacker's data is also within the decision boundary on the feature hyperplane. More importantly, considering the attacker will only launch attacks on a limited number of dangerous commands (e.g. authentication passphrase and delete files), the cost of this type of attack is relatively low. The attack can keep generating the desired internal body voices for these commands until the final values of both P_1 and P_2 are close to zero. After that, the attacker can easily launch the attack without any cost.

Attack model. In this extension, we consider a new type of attacker called expert attacker. Beyond the ability of obstruction attackers and replay attackers, expert attackers can acquire the detailed design of our system. Moreover, the expert attacker is able to generate fake internal body voices in order to make the values of both P_1 and P_2 are close to zero.

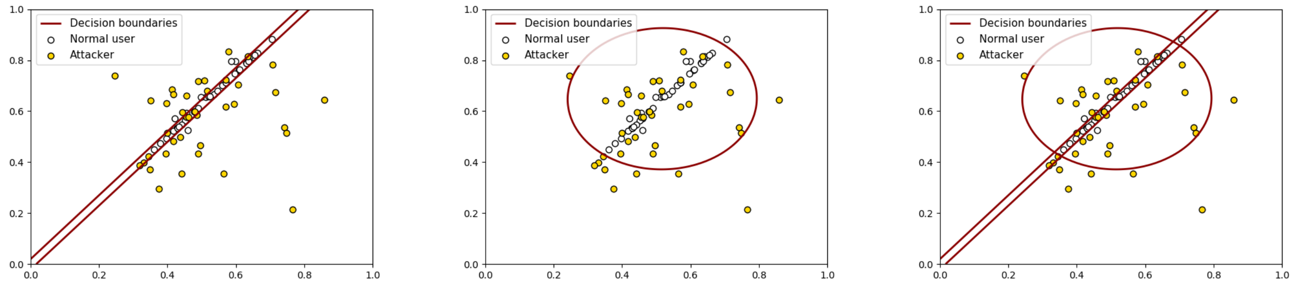
5.2 Enhanced classification for robust liveness detection

To reject this new type of attacker, we need to carefully design the decision boundary based only on the normal user's data so that it can not only accurately separate the data of the normal user and attackers (both obstruction attacker and replay attacker) but also effectually defend against expert attackers. This is challenging for two reasons. First, since the InAR headsets are battery-limited devices, we need to keep the computation cost as low as possible. Therefore, we should avoid using complex classification models like neural networks. Second, the generalization of the selected classification model must be high. For instance, even if we can obtain the perfect decision boundary for a user's data by carefully setting the parameters of the classification model, these parameters may not be suitable for classifying data of others users. To address the above challenges, we propose a new classification model. Specifically, we first train two weak classifiers based on the covariance and support vector data description (SVDD) independently using the same training dataset. Finally, we combine two weak classifiers to create a strong one. Any testing data is recognized as from the normal user if and only if it passes both weak classifiers.

Minimum covariance determinant-based classifier. Since the attacker's data is away from the clusters of the normal user's data, if the testing data is from the attacker, it must appear like an outlier on the feature hyperplane. One common way of performing outlier detection is to assume that the regular data comes from a known distribution (e.g. data are Gaussian distributed). From this assumption, we generally try to define the shape of the data, and can define outlying observations as observations which stand far enough from the fit shape. To define the shape of the data, we first fit a Minimum Covariance Determinant estimation based on the training data by setting a proportion of training data as inliers. Then, for each testing data, we calculate its Mahalanobis distance obtained from this estimation based on:

$$D_M(\vec{P}) = \sqrt{(\vec{P}-\vec{\mu})^T E^{-1}(\vec{P}-\vec{\mu})} \tag{8}$$

where $\vec{P} = [P_1, P_2]$ is a data point, $\vec{\mu}$ is the mean value of training data, and E is the covariance matrix estimation. We use the Mahalanobis distance as a measure of outlyingness. A data point is detected as from the attacker if its Mahalanobis distance is larger than those of inliers. Fig. 11(a) shows the decision boundary of Minimum Covariance Determinant-based classifier when it is trained based on 20 training instances and 5% of all data is regarded as outliers. We can observe that the decision boundary can efficiently



(a) Robust covariance.

(b) SVDD.

(c) MCD-SVDD model.

Fig. 11. Decision boundaries of different classifiers.

separate the data of attackers and the normal user, but much blank spaces are still covered at the upper right and lower left corners. We further eliminate these blank spaces by levering the second weak classifier. Moreover, we find that the decision boundary of covariance determinant-based model is almost linear for some dataset but can also be a curve for others.

SVDD-based classifier. In our system, the basic idea of SVDD is to find a circular boundary around the data of the normal user in the two-dimensional feature space and minimize the volume of this hypersphere in order to reject as many outliers as possible. In general, the hypersphere is characterized by a center **a** and a radius R. Here R is defined as the the distance from the center **a** to any support vector, and **a** is a linear combination of the support vectors. To get a soft margin, slack variables ξ_i with penalty parameter C are used to form data points x_i. Then, we can formulate the minimization as:

$$\begin{aligned} \min_{R,\mathbf{a}} \quad & R^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & ||x_i - \mathbf{a}||^2 \leq R^2 + \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{9}$$

From Karush-Kuhn-Tucker optimality conditions, we have

$$\mathbf{a} = \sum_{i=1}^{n}\alpha_i x_i \tag{10}$$

where α_i can be solved based on the following optimization problem:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n}\alpha_i K(x_i, x_i) - \sum_{i,j=1}^{n}\alpha_i\alpha_j K(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i = 1 \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{11}$$

where K is the kernel function. In our system, we used radial basis function as the kernel function and set an upper bound on the fraction of training errors to 5%. Fig. 11(b) shows the decision boundary of our SVDD-based classification model learnt from 20 training data of the normal user. We can observe that SVDD-based model can effectively reject data points in the lower left and upper right areas in the feature space. Although SVDD-based classification model cannot really reject the data points of obstruction attacks because of the circular boundary and the distribution of data points, it can be combined with our MCD-based classification model for better system performance.

Classification models combination. As we stated above, both the MCD-based and SVDD-based classification models have their strength and limitations on rejecting attackers. Specifically, MCD-based classification model can effectively defend against obstruction and replay attacks, while having limited ability to reject expert attackers. On the other hand, SVDD-based classification model can protect the normal user from expert attacks, but fails to defend against obstruction and replay attacks. Based on these two insights, we propose a new classification model by combining the strengths of MCD-based and SVDD-based classification models. As shown in Fig. 12, the extracted features (P_1 and P_2) are first sent to the MCD-based model. If the incoming features are recognized as from the normal user, they are immediately sent to the SVDD-based model for the second round classification. A pair of features is recognized as from the normal user only when they are classified as from the normal user by both classifiers. Otherwise, the features are recognized as from an attacker. In addition, since the requirements of performing expert attacks are higher than those of either obstruction attacks or replay attacks, we argue that it is easier for the normal user to suffer from obstruction and replay attacks. Therefore, we let the testing data pass the MCD-based classifier first to reduce the amount of data that is sent to the SVDD-based classifier. Fig. 11(c) shows the decision boundary of the MCD-SVDD model. We can see that most data points of the normal user are within the boundary while almost all data points of three types of attackers are outside the boundary. Similar to the line-fitting model, we also combined the classification results of multiple words through a weight majority voting game. The weight of each player the voting procedure is the same as the one used for the line-fitting model. The only difference is that the minimal number of positive votes is set to $0.3 \times n$ for the MCD-SVDD model.

6 EVALUATION

In this section, we introduce the experimental setting of our evaluation and show the experimental results.

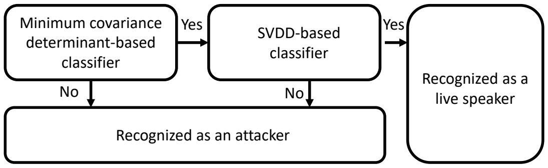

Minimum covariance determinant-based classifier
Yes
SVDD-based classifier
Yes
Recognized as a live speaker
No
No
Recognized as an attacker

Fig. 12. Workflow of classification model.

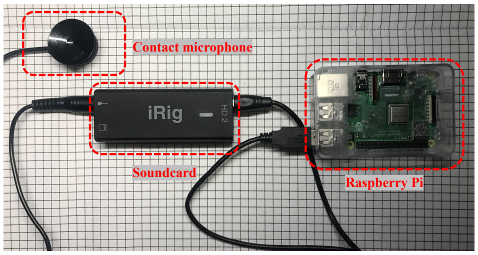
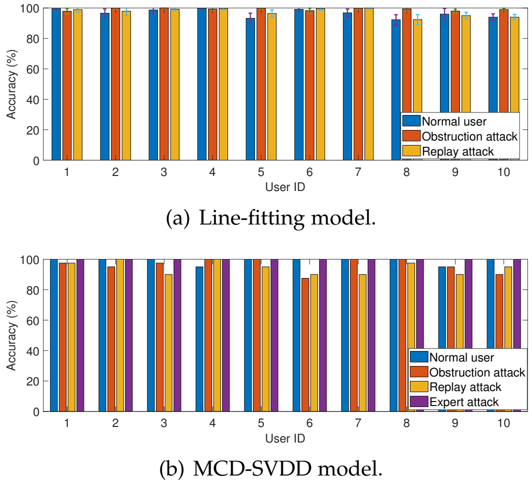


Fig. 13. Testbed for collecting internal body voice.



(a) Line-fitting model.

(b) MCD-SVDD model.

Fig. 14. Overall performance of our system.

6.1 Hardware and data collection

Our system consists of two components: a testbed for collecting internal body voice and a smartphone for collecting air voice. We implemented our testbed using a Raspberry Pi 3, an iRig HD 2 soundcard, and an AXL contact microphone. Besides, we used a Nexus 5 to collect users' air voices and transmit them to the Raspberry testbed through WiFi. Both the smartphone and the Raspberry testbed were synchronized to the same server. Our experiments involved 10 volunteers, and all of them were asked to repeat saying sentences of different lengths to our system. In order to make sure the contact microphone can capture the internal body voice during the data collection, we attached the contact microphone to a hat and asked each volunteer to wear it. Each volunteer wore the hat in their own way and was in a comfortable posture they preferred. For data analysis and processing, the data was then transmitted to a desktop computer with Intel(R) Core(TM) Devils Canyon Quad-Core i7-8700K @ 4.00 GHz CPU and 16 GB of RAM. To evaluate the system performance for legitimate users, we asked each user to say say a 5-word sentence 50 times. Among 250 collected words, we randomly picked 40 words as the training data and the remaining as the testing data. To examine how well our system can defend against attacks, we generate a dataset that contains 200 words for each attack model.

Performance metrics. In our experiments, we use the following performance metrics to evaluate the validation performance of our system. Accuracy is defined as the rate at which a normal user is correctly accepted or an attacker is successfully rejected by the system.

6.2 Overall performance

Line-fitting model. We first evaluated the system performance for normal users and against two types of attacks using the line-fitting model. In this experiment, we used the voices of 40 words collected from the normal user as the training data. The correlation threshold λ was set to 0.1, and the distance threshold γ was set to the 95% largest distance of normal user's training data. We asked each user to say a 5-word sentence 50 times. Moreover, we repeated this procedure for 10 times to study the variance of true acceptance rates of different volunteers, and the experimental results are shown in Fig. 14(a). We can observe that our system can correctly accept the normal user with mean accuracy of 97% for all users. Even in the worst case, our system can still achieve a high accuracy of 92.3% for normal users. By studying normal users' data that is wrongly rejected, there are two main reasons why the performance is degraded. First, there are two volunteers who speak softly, which makes their voices easier to be covered by background noise. Second, volunteers' activities may cause a slight movement of the hat, which introduces high-energy noise to the internal body voice and reduces similarity between the two voices.

We further evaluated how accurately our line-fitting model can reject two types of attacks. To collect the data for the obstruction attack, we let a volunteer speak loudly while the normal user (another volunteer) was wearing the hat. To collect the data for the replay attack, we used a Nexus 6 smartphone to record the victim's voices at a distance of 0.5 meters. Then, we used the loudspeaker of a smartphone to replay in the victim's voice to our system. At the same time, the replay attacker said the same sentence to our system while wearing the hat. Moreover, we made sure the gender of the victim and the replay attacker are the same. We leveraged the fitted straight line for the victim to determine the legitimacy of the attacker's data, and the results are shown in Fig. 14(a). We can see that our system can provide high accuracy against both types of attacks. More specifically, our system can provide a mean accuracy of 99.2% and 98% for defending the obstruction attack and replay attack, respectively. The accuracy of successful defenses is not 100% for two reasons. First, some internal body voices in the training dataset contained noise, which increased the distance threshold. Second, the slight movement of the user's head may also introduce random high-energy influence to the spectrogram. In rare cases, the filtered spectrogram of noise was similar to that of some words (e.g. "eight"). As a whole, our system can provide high-security protection for users against obstruction attack

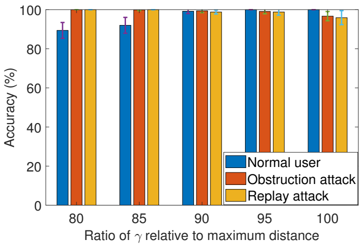
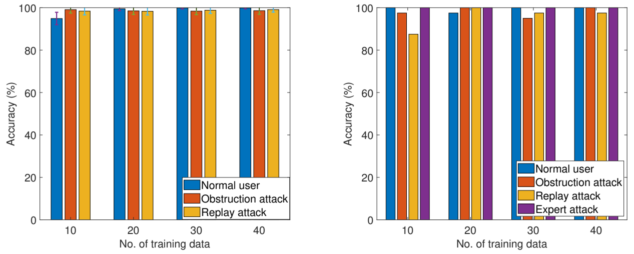


Fig. 15. Influence of distance threshold γ.



(a) Line-fitting model.

(b) MCD-SVDD model.

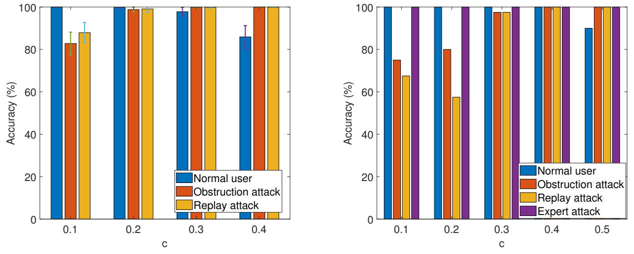
Fig. 16. System performance with respect to the number of training instances.

and replay attack while still ensuring good user experience for normal users.

MCD-SVDD model. We also evaluated the performance of our MCD-SVDD model for five-word sentences using the same dataset. Specifically, we used 40 training instances collected from the normal user to build two classifiers and set the upper bound of outliers in training data to 5%. Fig. 14(b) illustrates the classification accuracy for normal users and against three types of attacks. It is clear that the MCD-SVDD model can accurately accept normal users and reject obstruction and replay attacks with high accuracy. Besides, by improving the decision boundary, the expert attack can be efficiently detected with accuracy of 100% for all ten users in our dataset.

6.3 Influence of training dataset size

In practice, we want the amount of training data to be as small as possible to reduce the training cost for new users. Therefore, we evaluated how much training data is needed by our system in order to provide both high-security protection and good user experience. Fig. 16(a) shows the performance of the line-fitting model with different sizes of the training dataset. We can see that the average accuracy for the normal user is improved a lot by using more data for training since we have more knowledge about the distribution of the normal user's data. By contrast, the average accuracy of successful defense against either of the two attacks is almost the same by using different numbers of training data. The reason behind this is that the data distribution of the attacker's data is significantly away from that of the normal user. Therefore, our system can accurately reject two types of attacks even if the training data is limited. We also did the same experiment on the MCD-SVDD model, and the results are shown in Fig. 16(b). When there are only 10 training instances available, the MCD-SVDD model can only reject the replay attack with an accuracy of about 85%. By increasing the number of training instances to at 20 or more, the system performance can be largely improved to nearly 100%, especially against replay attacks. Overall, our system can provide both high-security protection and good user experience after collecting the voices of 20 words from the normal user, which is low-cost and easy to be used for new users.



(a) Line-fitting model.

(b) MCD-SVDD model.

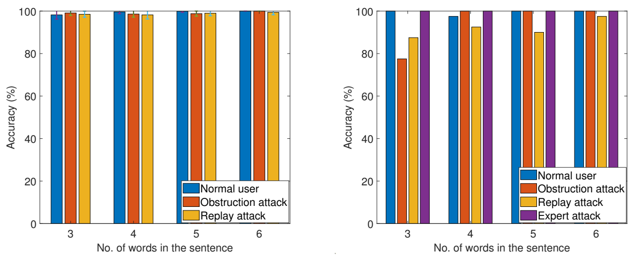
Fig. 17. System performance with respect to voting threshold.

6.4 Influence of the ratio of γ relative to the maximum distance

In our default experimental setting, the distance threshold γ is set to 95% of the highest distance in the training data. In real scenarios, there is a trade-off on determining the value of γ. A small distance threshold can provide extremely high true rejection rate against two types of attackers, but it also makes it hard for normal users to use our system. A high distance threshold can ensure good user experience, but more attackers are wrongly accepted. In this subsection, we study what is the proper value of γ for different users. Fig. 15 shows the system performance with different values of γ. It is clear that the average accuracy for normal users rises with the increase of γ, while the average accuracy of successful rejection drops. When γ is the 95% highest distance in the training dataset, the true acceptance rate and the true rejection rate are nearly equal. Therefore, we let the γ be equal to the 95% highest distance in the training dataset to balance the need for security protection and user experience.

6.5 Influence of voting threshold

The performance of our system relies on a successful voting procedure. Hence, a proper voting threshold is important. Similar to the distance threshold, there is also a trade-off on determining the value of the voting threshold. If the voting threshold is too small, all normal users can be accepted, but some attackers may also be wrongly regarded as the normal user. If we assign a high value to the voting threshold, all attackers can be successfully rejected, but the user experience of normal users is ruined. In this subsection, we study what is the proper value of the voting threshold. Here we use $c * n$ to represent the voting threshold where c is a constant and n is the number of words in a sentence (voice command). We evaluated the performance for 5-word sentences using the default parameters and adjusted the value of c, and the results are shown in Fig. 17(a). We can



(a) Line-fitting model

(b) MCD-SVDD model.

Fig. 18. System performance with respect to the sentence length.

see that the average accuracy for normal users drops rapidly when c is larger than 0.2. Moreover, our system can provide good security protection after c reaches 0.2. Therefore, we let the c be equal to 0.2 in our default system setting for the line-fitting model. Since the value of the voting threshold largely depends on the system performance on a single word, we also studied the proper value of voting threshold for our MCD-SVDD model. As shown in Fig. 17(b), the MCD-SVDD model cannot effectively defend against attacks when the voting threshold is less than 0.3. Moreover, when the voting threshold is greater than 0.4, the system can wrongly reject the normal user with a probability of about 15%. Therefore, the proper value voting threshold should be between 0.3 and 0.4 based on our experiments.

6.6 Influence of sentence length

We also evaluated the system performance for sentences of different lengths. Here the sentence length refers to the number of words in the sentence. When the length of the sentence is short, the wrong classification of a few words may dominate the voting procedure and give the incorrect detection result. For longer sentences, the voting procedure can tolerate a few wrong predictions by involving more players. In this subsection, we study what is the minimum sentence length to ensure good security protection and user experience, and the results are shown in Fig. 18(a). We can see that the system performance is improved with a higher number of words in a sentence. When the sentence length is 6, our system can provide average accuracy of about 100% for both accepting normal users and rejecting attackers. Moreover, with a higher numbers of words in a sentence (voice command), the variance of both the true acceptance rate and the true rejection rate are reduced, as shown in the error bar in Fig. 18(a). This fact implies that the robustness of our system is improved by saying a voice command with more words. The influence of the sentence length is more significant for the MCD-SVDD model, as shown in Fig. 18(b). For a three-word sentence, the system can only defend against attacks with accuracy of as low as 77.5%. By having two more words in the sentence, the system can protect the normal user from any type of attacks with accuracy of at least 95%. Considering most voice commands supported by current AR applications have lengths of at least 4 words (e.g. show me the road), our system can provide good enough security protection and user experience for them.

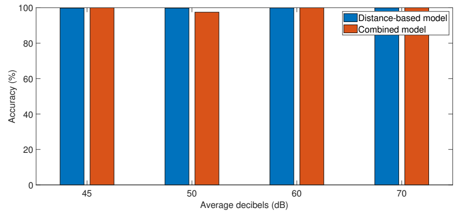


Fig. 19. System performance with respect to the background noise.

6.7 Influence of background noise

Since our system records the air voice using a normal microphone, the background acoustic noise (e.g. conversation or music) may cover the features in the air voice and degrade the performance for normal users. To evaluate the robustness of our system against background noise in terms of accepting normal users, we asked one volunteer to speak a 5-word sentence to our system. During the data collection, we used two loudspeakers to simulate different noise levels from 45 dB (average home noise) to 70 dB (inside a car at 60 mph). We did not consider greater noise in our evaluation for two reasons: 1) Most voice-based AR applications are not designed for noisy environments (e.g. video call); 2) The performance of voice recognition and authentication systems can also be degraded by strong noise. Fig. 19 shows the evaluation results. We can observe that our system can achieve a high accuracy of at least 97.5% for all noise levels no matter which classification is used, which means the user experience of the normal user can be ensured in daily use. More importantly, we found that the reason why our system can still provide good performance in a noisy environment is that the AR users will subconsciously raise their volumes in a noisy environment, which makes the features of their voices are more significant than those of background noises. By applying spectrogram enhancement techniques, these background noise can be largely removed.

7 DISCUSSION

In this section, we will discuss the usability, limitations, and future work of our system.

7.1 Influence of acoustic noise

In real usage scenarios, the voice interaction between AR headsets and AR users is also influenced by background noise. Strong acoustic noise will influence not only the performance of our system but also the normal speech recognition. To reduce the influence of the acoustic noise, the industry has come up with several solutions, such as noise-canceling techniques and unidirectional microphones. For example, Microsoft HoloLens [2] use 2 microphones to collect information about the acoustic environment in order to cancel the noise and another 2 microphones to collect only the user's voice. In order to simulate the hardware of real AR headsets, we use a similar noise-canceling microphone to capture the user's air voice. This implementation makes sure that our system can be integrated into current AR headsets without extra effort.

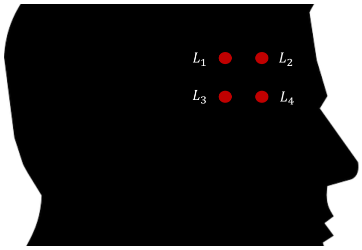
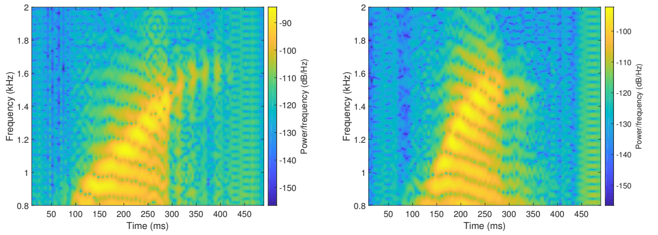


Fig. 20. Four positions around the temple.



(a) First position.

(b) Second position.

Fig. 21. Spectrograms of body voice when the sensor is at two positions.

7.2 Influence of the position of contact microphone

In practice, the user may attach the contact microphone anywhere around the temple based on the framework design of the AR headset. Even for the same headset, we cannot ensure that the user can attach it at the same position every time. In order to evaluate the robustness of our system against different wearing positions of the contact microphone, we collect the data from 4 different positions around the temple, as shown in Fig. 20. The distance between neighboring positions is about 2 cm. We collect training data from L_1 to predict the testing data from the other locations. Fig. 21 shows the spectrograms internal body voice when the contact microphone is at two different positions. We can see that these two spectrograms reserve feasures at almost the same frequency bands, which is the reason why the slight position change of the contact microphone will not influence the performance too much. Experimental results show that our system can still achieve the same performance (over 97%) for both normal users and attackers, which implies our system is robust enough to wearing position change.

7.3 Usability

Except for accuracy, validation time is also critical and determines the usability. We further test the time our system needs to process the raw signal and get the final validation results. Experimental results show that our method can finish the work within 500ms in all cases, which means our system can respond to the user right after the user stops recording and does not introduce too much overhead. Compared with existing works, our system does not need user's extra effort in operating the AR headsets, e.g., moving the device around the audio source. To further strengthen the usability of our system, we adopt the same human-computer interaction methods used by current AR headsets, so that users can quickly get used to using our system.

7.4 Long-term stability

Considering the way of speaking may change for long-term usage, the fitted line that is based on historical training data may not accurately classify new data. To evaluate the robustness of our approaches during long-term usage, we further collect testing data from 2 volunteers 5 weeks after since collecting their training data. Experimental results show that our system can still successfully accept a normal user with an accuracy of 99.1%, which is in line with our expectation. Our system detects the legitimacy of the speaker by measuring the correlation and shared information between two voices. Therefore, as long as two voices are from the same live speaker, there always exists a high correlation between two voices no matter what speaking habit the user has. Moreover, the proportions of shared information should also be stable during long period since the internal body propagation of each user will not change too much.

8 Related Work

In this section, we discuss current voice-based AR applications, automatic speaker verification systems, and state-of-the-art voice-spoofing attacks.

8.0.1 Voice-based AR applications;

There are several benefits of involving voice in the interaction methods. First, voice-based interaction can improve the immersion of AR experience. Second, it is widely accepted that the audio is processed faster than the visual stimulus. For example, Barde et al. [5] showed that audio cues can reduce reaction time up to 50% for shooting games. Therefore, the voice is becoming one of the major input methods of current AR headset and applications. Current AR applications and headsets use voice for either controlling or authentication. Most AR headsets support speech recognition and voice-based control. For instance, HoloLens [2] uses the voice as the intention mechanism to issue a command. Besides, voice can also be used for authentication. These voice-based authentication applications offer opportunities to attackers who are able to launch a voice-spoofing attack by imitating a victim's voice, tone, and speaking style. This attack could harm the victim's reputation, safety, and property.

8.0.2 Automatic voice recognition and speaker verification;

Automatic speech recognition systems aim to modulate a speech signal to a series of words so that users can interact with their devices using the voice interface. In the course of the last few years, there has been a remarkable advancement in the domain of speech recognition [8], [22]. For example, Williams et al. [8] presented a neural network that learns to transcribe speech utterances to characters. The proposed approaches can achieve a low word error rate of 8%. Voice can also be used as the biometrics for authentication using speaker verification techniques. Typically, an automatic speaker verification (ASV) system is designed to accept or reject a speech sample submitted by a user for claiming certain identity [28]. Recently, the development of ASV systems has made major progress as they are widely adopted by mobile devices (e.g. smartphones) and online commerces [15], [19]. Most ASV systems are text-independent, which means the user needs to repeat a fixed passphrase. The reason text-independent ASV systems are widely selected for authentication application is that they are able to accept arbitrary utterances, i.e., different speaking habits and languages from speakers [6]. The current practice of building an ASV system involves two processes: offline training and runtime verification. During the offline training phase, the ASV system uses several speech samples provided by the genuine speaker to extract certain spectral, prosodic [4], [27], or other high-level features [12], [20] and uses them to create a speaker model. Then, in the runtime verification phase, the ASV system uses the trained speaker model to verify the incoming voice.

8.0.3 Attacks on voice recognition and speaker verification systems;

Both voice recognition and speaker verification system suffer from attacks. Recent researches [7], [10], [14], [29], [34] have shown that spoken words can be mangled such that they are unrecognizable to humans, which poses a serious threat to voice recognition systems. For instance, [34] showed that it is feasible to send inaudible attack commands. Also, various approaches are proposed to break the biometric identification of the victim [18], [30]. For example, [30] shows that an attacker can overcome text-dependent ASV systems by concatenating speech samples from multiple short voice segments of the target speaker. Due to the simplicity of voice spoofing attacks, a few research papers have been published in developing relay attack countermeasures [9], [11], [13], [16], [17], [21], [24], [25], [31], [32], [36]. However, all these countermeasure systems are particularly designed for smartphone, which makes them hard to be implemented on AR headsets. For example, the liveness detection system proposed in [35] can detect the replay attacker by reusing smartphone as a sound radar. However, this work cannot be implemented on AR headsets since AR headsets do not have a speaker that is towards the user's mouth.

9 CONCLUSION

Voice-based interaction is usually used as the primary interaction method for AR headsets due to its good user experience and performance. AR users rely on accurate and secure voice input to communicate with AR headsets. However, recent researches have shown that an attacker can easily perform various attacks with the help of state-of-the-art voice synthesis/conversion software. To secure the voice input on AR headsets, we propose a robust and low-cost solution for defending against voice-spoofing attacks on AR headsets with high accuracy. Our system leverages a contact microphone to record the internal body propagation of the voice. A user legitimacy is determined by measuring the correlation and similarity between the internal body voice and air voice. To our best knowledge, our system is the first to protect the voice input for AR headsets. Especially, our signal processing and feature extraction components can remove noise in raw voice signals and extract useful features that can represent the correlation and similarity between two spectrograms of two voice signals. Moreover, we proposed two classification models (line-fitting model and MCD-SVDD model) to detect the voice input from the attacker by targeting different attack models. Experimental results show that our system can accept normal users with average accuracy of 97% and defend against obstruction attack and replay attack with average accuracy of 99.2% and 98%, respectively. More importantly, even if the attackers can fool our line-fitting model by manipulating special voice signals, our combine model can still reject them with accuracy of 100%.

ACKNOWLEDGEMENT

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

REFERENCES

[1] CM-01B Contact Microphone,http://www.mouser.com/ds/2/418/contact_microphone-769347.pdf.

[2] Microsoft HoloLens, https://www.microsoft.com/en-us/hololens/hardware.

[3] Voice frequency,https://en.wikipedia.org/wiki/voice_frequency.

[4] A. G. Adami, R. Mihaescu, D. A. Reynolds, and J. J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–788. IEEE, 2003.

[5] A. Barde, M. Ward, W. S. Helton, M. Billinghurst, and G. Lee. Attention redirection using binaurally spatialised cues delivered over a bone conduction headset. In *Proc. of HFES*, volume 60, pages 1534–1538. SAGE Publications Sage CA: Los Angeles, CA, 2016.

[6] J. P. Campbell. Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[7] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530, 2016.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. of ICASSP*, pages 4960–4964. IEEE, 2016.

[9] 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 *Proc. of ICDCS*, pages 183–195. IEEE, 2017.

[10] G. Cho, J. Choi, H. Kim, S. Hyun, and J. Ryoo. Threat modeling and analysis of voice assistant applications. In *International Workshop on Information Security Applications*, pages 197–209. Springer, 2018.

[11] H. Dai, W. Wang, A. X. Liu, K. Ling, and J. Sun. Speech based human authentication on smartphones. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2019.

[12] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Proc. of EUROSPEECH*, 2001.

[13] B. Hao and X. Hei. Voice liveness detection for medical devices. In *Design and Implementation of Healthcare Biometric Systems*, pages 109–136. IGI Global, 2019.

[14] C. Kasmi and J. L. Esteves. Iemi threats for information security: Remote command injection on modern smartphones. *IEEE Transactions on Electromagnetic Compatibility*, 57(6):1752–1755, 2015.

[15] K. A. Lee, B. Ma, and H. Li. Speaker verification makes its debut in smartphone. *IEEE signal processing society speech and language technical committee newsletter*, 2013.

[16] 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 the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 81–90. ACM, 2018.

[17] S. Mochizuki, S. Shiota, and H. Kiya. Voice liveness detection using phoneme-based pop-noise detector for speaker verifcation. In *Odyssey 2018 The Speaker and Language Recognition Workshop. ISCA, Les Sables d'Olonne*, 2018.

[18] D. Mukhopadhyay, M. Shirvanian, and N. Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. In *Proc. of Esorics*, pages 599–621. Springer, 2015.

[19] Nuance. Nuance vocal password. http://www.nuance.com/, 2013.

[20] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, et al. The supersid project: Exploiting high-level information for high-accuracy speaker recognition. In *Proc. of ICASSP*, volume 4, pages IV–784. IEEE, 2003.

[21] M. Sahidullah, D. A. L. Thomsen, R. G. Hautamaki, T. Kinnunen, Z.-H. Tan, R. Parts, M. Pitkanen, M. Sahidullah, D. A. L. Thomsen, R. Gonzalez Hautamaki, et al. Robust voice liveness detection and speaker verification using throat microphones. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 26(1):44–56, 2018.

[22] S. Scanzio, S. Cumani, R. Gemello, F. Mana, and P. Laface. Parallel implementation of artificial neural network training for speech recognition. *Pattern Recognition Letters*, 31(11):1302–1309, 2010.

[23] F. Schiel. Automatic phonetic transcription of non-prompted speech. 1999.

[24] J. Shang, S. Chen, and J. Wu. Defending against voice spoofing: A robust software-based liveness detection system. In *Proc. of MASS*, pages 28–36. IEEE, 2018.

[25] J. Shang, S. Chen, and J. Wu. Srvoice: A robust sparse representation-based liveness detection system. In *Proc. of ICPADS*. IEEE, 2018.

[26] M. Shirvanian and N. Saxena. Wiretapping via mimicry: Short voice imitation man-in-the-middle attacks on crypto phones. In *Proc. of CCS*, pages 868–879. ACM, 2014.

[27] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472, 2005.

[28] R. Togneri and D. Pullella. An overview of speaker identification: Accuracy and robustness issues. *IEEE circuits and systems magazine*, 11(2):23–61, 2011.

[29] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields. Cocaine noodles: exploiting the gap between human and machine speech recognition. *WOOT*, 15:10–11, 2015.

[30] J. Villalba and E. Lleida. Detecting replay attacks from far-field recordings on speaker verification systems. In *Proc. of BIOID*, pages 274–285. Springer, 2011.

[31] J. Villalba and E. Lleida. Preventing replay attacks on speaker verification systems. In *Proc. of ICCST*, pages 1–8. IEEE, 2011.

[32] 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-IEEE Conference on Computer Communications*, pages 2062–2070. IEEE, 2019.

[33] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153, 2015.

[34] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. Dolphinattack: Inaudible voice commands. In *Proc. of CCS*, pages 103–117. ACM, 2017.

[35] L. Zhang, S. Tan, and J. Yang. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proc. of CCS*, pages 57–71. ACM, 2017.

[36] L. Zhang, S. Tan, J. Yang, and Y. Chen. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proc. of CCS*, pages 1080–1091. ACM, 2016.

Jiacheng Shang is the Assistant Professor in the Department of Computer Science at Montclair State University. He received his Ph.D. degree in Computer and Information Sciences from Temple University in 2020. His current research focuses on the security and privacy issues in mobile cyber-physical systems.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.