VoiceListener: A Training-free and Universal Eavesdropping Attack on Built-in Speakers of Mobile Devices

LEI WANG, Zhejiang University, China and ZJU-Hangzhou Global Scientific and Technological Innovation Center, China
MENG CHEN, Zhejiang University, China
LI LU*, Zhejiang University, China
ZHONGJIE BA, Zhejiang University, China
FENG LIN, Zhejiang University, China
KUI REN, Zhejiang University, China

Recently, voice leakage gradually raises more significant concerns of users, due to its underlying sensitive and private information when providing intelligent services. Existing studies demonstrate the feasibility of applying learning-based solutions on built-in sensor measurements to recover voices. However, due to the privacy concerns, large-scale voices-sensor measurements samples for model training are not publicly available, leading to significant efforts in data collection for such an attack. In this paper, we propose a training-free and universal eavesdropping attack on built-in speakers, *VoiceListener*, which releases the data collection efforts and is able to adapt to various voices, platforms, and domains. In particular, *VoiceListener* develops an aliasing-corrected super resolution mechanism, including an aliasing-based pitch estimation and an aliasing-corrected voice recovering, to convert the undersampled narrow-band sensor measurements to wide-band voices. Extensive experiments demonstrate that our proposed *VoiceListener* could accurately recover the voices from undersampled sensor measurements and is robust to different voices, platforms and domains, realizing the universal eavesdropping attack.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; • **Security and privacy** → **Embedded systems security**.

Additional Key Words and Phrases: Eavesdropping, speakers, aliasing correction, super resolution.

ACM Reference Format:
Lei Wang, Meng Chen, Li Lu, Zhongjie Ba, Feng Lin, and Kui Ren. 2023. *VoiceListener*: A Training-free and Universal Eavesdropping Attack on Built-in Speakers of Mobile Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 32 (March 2023), 22 pages. https://doi.org/10.1145/3580789

*Li Lu is the corresponding author. Email: li.lu@zju.edu.cn

Authors' addresses: Lei Wang, Zhejiang University, School of Cyber Science and Technology, Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, China and ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou, China, kleiwang@zju.edu.cn; Meng Chen, Zhejiang University, School of Cyber Science and Technology, Hangzhou, China, meng.chen@zju.edu.cn; Li Lu, Zhejiang University, School of Cyber Science and Technology, Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Hangzhou, China, li.lu@zju.edu.cn; Zhongjie Ba, Zhejiang University, School of Cyber Science and Technology, Hangzhou, China, zhongjieba@zju.edu.cn; Feng Lin, Zhejiang University, School of Cyber Science and Technology, Hangzhou, China, flin@zju.edu.cn; Kui Ren, Zhejiang University, School of Cyber Science and Technology, Hangzhou, China, kuiren@zju.edu.cn.

2474-9567/2023/3-ART32 $15.00
https://doi.org/10.1145/3580789

1 INTRODUCTION

As the development of voice services, an increasing number of industry IoT and mobile devices have deployed voice assistants, including mobile operating system's built-in assistant (e.g., Apple Siri for iOS[4], Google Assistant[17] and Samsung Bixby[41] for Android), smart speakers (e.g., Amazon Alexa for Echo[1], Google Assistant for Google Home[17]). Even the traditional operating systems deploy relative solutions, such as Apple Siri for macOS[4] and Microsoft Cortana for Windows[35]. A report[9] predicts that the global voice assistant market is anticipated to the valuation of around $7.8 billion by 2023, at 39.3% Compound Annual Growth Rate (CAGR) during 2017∼2023. However, due to its personalized service nature, the voice interactions usually embed sensitive and private information. Representative examples include a personal chat or an oral password for authentication that could be embedded in voices during the interactions. Combined with the broadcasting playing of speakers, many users have privacy concerns on the voices gradually. Even the popularity of sensor-enriched mobile devices intensifies such a voice leakage threat, which also arouses the research interests.

Except for the most straightforward voice eavesdropping by microphones, recent studies turn to employ built-in sensors as side channels for imperceptible eavesdropping. Early work Gyrophone[34] exploits the built-in gyroscopes to capture voices emitted by speakers, which are further used for personal information inference, including gender identification, speaker verification and speech recognition. Following works[2, 49] turn to the more accurate accelerometer for hot word detection, significantly outperforming the gyroscope-based solution. A more recent study AccelEve[5] extends this attack to general voice eavesdropping using deep learning-based approaches. All of these works demonstrate the feasibility of using built-in sensors to eavesdrop voices broadcasted from speakers, and most of them are based on prevalent machine learning methods, i.e., the data-driven approaches. Although the adversaries may benefit from the low-effort attack design by the data-driven approaches, the lack of large-scale voices-sensor measurements samples probably restrict the propagation of such an attack. Even worse, due to the significant privacy concern such datasets are also not expected to be available publicly in the future.

Different from aforementioned researches, our work aims to propose a training-free and universal eavesdropping attack for built-in speakers, which releases the requirement of data collection and could cross different voices, platforms and domains. The basic idea is to exploit model-driven methods for uncovering voice leakage side channels by built-in sensors and recovering the voices from undersampled sensor measurements. To realize a training-free and universal eavesdropping attack across different voices, platforms and domains, we face several key challenges. *Signal distortion*: the limited sampling rate of built-in sensors results in narrow-band sensor measurements and severe aliasing interference, making voice recovering challenging. *Voice diversity*: the targeted voices played by speakers are generated by different subjects (e.g., different persons or an AI-synthesized assistant), also with rich texts, whose diversity should be handled by the attack. *Platform difference*: Due to the variation in design and encapsulation, there exist significant differences on different device models, such as relative positions between sensor and speaker, sensor model, etc., which brings difficulty in designing a universal solution. *Domain variation*: a speaker broadcasting scenario could occur in any environment and under any user interaction, such as placing on an office table or holding in hand while walking on the street, which should be fit without the prior knowledge for the attack.

In this paper, we first introduce the threat model of eavesdropping on the built-in speakers of mobile devices, and investigate the potential side channels for voice leakage. We then define the design goals of the universal eavesdropping attack. Based on the goals, we propose *VoiceListener*, a model-driven method to realize the universal eavesdropping on the voices broadcasted from built-in speakers. In *VoiceListener*, an adversary pre-implants a spy APP in the user's mobile device to invoke the built-in sensors to collect corresponding sensor measurements when the built-in speaker broadcasts the voices. Leveraging voice leakage side channels, i.e., the vibrations and magnetic field variations, we select the accelerometer, gyroscope and magnetometer as the eavesdropper. To collect the valid sensor measurements, *VoiceListener* selects the most sensitive axis from the 3-D sensor measurements

and denoises the raw signals for voice recovering. Due to the limited sampling rate of built-in sensors, the sensor measurements are severely distorted with narrow-band information and significant aliasing compared with raw voices. To handle this, we design an aliasing-corrected super resolution, including an aliasing-based pitch estimation and an aliasing-corrected voice recovering, to convert the undersampled sensor measurements to understandable voices. The proposed algorithm is a model-driven approach, without labor-intensive data collection and labeling, as well as the time-consuming model training. Experimental results demonstrate that *VoiceListener* could achieve acceptable performance in voice recovering, outperforming state-of-the-art solutions.

Our contributions are highlighted as follows.

- We demonstrate a training-free and universal eavesdropping attack, *VoiceListener*, which leverages various built-in sensors to capture the voices leaked from speaker playing across different voices, platforms, and domains.
- We design an aliasing-based subharmonic summation method to fully exploit undersampled signals for accurate pitch estimation in sensor measurements.
- We propose an aliasing-corrected audio super resolution algorithm to recover an understandable voice from the undersampled and distorted sensor measurements, which releases the training efforts and handles the interference of undersampled aliasings.
- We conduct extensive experiments under real device models to evaluate the performance of *VoiceListener*, and the results show *VoiceListener* could outperform typical super resolution more than 60% in voice recovering.

The rest of this paper is organized as follows. We first review related works in Section 2. Section 3 presents the threat model and design goals of the universal eavesdropping attack. The design details of proposed *VoiceListener* is shown in Section 4. Section 5 presents evaluation results for *VoiceListener*. Finally, we discuss several potential countermeasures and make a conclusion in Section 6 and 7, respectively.

2 RELATED WORK

In this section, we review existing researches about voice eavesdropping attacks on mobile speakers and audio super resolution algorithms.

Speaker eavesdropping via built-in sensors. To avoid user awareness, current researches employ nonacoustic sensors as a pseudo-microphone to eavesdrop broadcasting voices. Early work Gyrophone[34] employs built-in gyroscopes of smartphones to capture voices emitted from loudspeakers for gender identification, speaker verification and even speech recognition. Limited by the low sampling rate, Gyrophone can only achieve a low accuracy. Following work Accelword[49] turns to employ an accelerometer to investigate the feasibility of hot words detection. Inspired by this observation, AccelEve[5] further develops a neural network-based solution to recover whole voices and recognize the speeches. In order to take full advantage of multiple sensors, recent works[14, 20] fuse the measurements from geophone, gyroscope and accelerometer to reconstruct intelligible speech signals. However, this work is constrained under the strong assumption of tight time synchronization of multiple distributed sensors, which is hard to realize compared with single-sensor solutions. V-Speech[32] captures noise-robust speech by the extremely high sampling rate vibration sensor in smart glasses. Furthermore, two recent works improve the learning-based eavesdropping model by well-designed features. Considering the frequency aliasing effect, the Neural Frequency Unfolding Model[46] introduces a simple alias unfolding layer in the network. Another latest work Vibphone[42] feeds extra critical features to learning-based model, and tries to generalize it to newly issued devices. All of these works demonstrate the feasibility of using built-in sensors to eavesdrop voices broadcasted from speakers, and most of them are based on prevalent machine learning methods. Though machine learning methods benefit adversaries with the low-effort end-to-end attack design, the lack of large-scale voices-sensor measurements samples probably restrict the impact of such an attack. Even worse,

due to the privacy concern, such datasets are not expected to be release in the future, which is significantly different from other typical machine learning tasks. Instead, we turn to another direction to exploit a model-driven approach for the eavesdropping, which releases the requirement of data collection and could introduce more severe threats on smart devices.

Audio super resolution. Typically, the audio super resolution indicates the bandwidth extension approach for improving the quality of telecommunications. The basic idea is to reconstruct the wide-band spectral components at the receiver side based on the available narrow-band voices from the transmitter side, which could be categorized into excitation signal extension and wide-band spectral envelope estimation. Makhoul et al.[30] first introduce spectral duplication for regeneration of high-frequency excitation signals. However, this method requires an anti-aliasing filter on the transmitter side, indicating it cannot adapt to every telecommunication scenario. Toward this end, other works turn to estimate the wide-band spectral envelope for super resolution, including Codebook Mapping[13, 38], and statistical models (e.g., GMM-based method[36, 37], HMM-based method[23, 48]). However, the previous solution induces significant computational complexity in the vector quantization, and the latter one requires amounts of data to achieve reliable estimations. As the development of deep learning, recent studies[25, 27, 29, 31, 39, 45] apply this technique for the data-driven audio super-resolution, which outperforms standard bandwidth extension approaches. However, the dependence on training with a specific dataset limits the robustness and universe of these approaches.

Different from the aforementioned learning-based eavesdropping and hardware-needed super resolution, our work aims to design a model-driven aliasing-corrected super resolution algorithm, and validates its effectiveness in various perspectives under larger datasets. Table 1 compares our work with two most related works, i.e., Gyrophone[34] and AccelEve[5], in terms of solution type, task type, and dataset for evaluations. Our work not only releases the training efforts for adversaries, but also adapts to various impact factors, including voices, platforms and domains.

3 ATTACK STATEMENT

In this section, we introduce the threat model of speaker eavesdropping attack, and investigate the voice leakage based on the mechanical structure of speakers. Furthermore, we present the design goals of the training-free and universal eavesdropping attack.

3.1 Threat Model

The threat model is shown in Fig. 1. In this attack, we assume that a legitimate user plays voice on a mobile device, representative examples include hands-free calls, video chats, voice messages, etc. During the voice playing via speakers, the user holds the device with an arbitrary position and rotation including placing it on an arbitrary surface, and holding it towards the ear.

For curious or malicious purposes (e.g., retrieving financial, business and personal information), an adversary intends to eavesdrop the voice produced by speakers of the user's device. To avoid raising the user's awareness,

Table 1. Comparison of *VoiceListener* and existing models.

Work	Solution Type	Task Type	Dataset	Device
Gyrophone[34]	Data-driven	Speech recognition	TIDIGITS(include speech of digits)	3
AccelEve[5]	Data-driven	Speech recognition Speech reconstruction	AudioMNIST(include speech of digits) Self-collected(include digits and letters)	3
VoiceListener	Model-driven	Speech recognition Speech reconstruction	AudioMNIST(include speech of digits) TIMIT(include various phonemes)	10

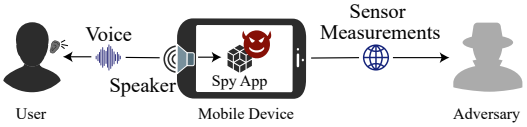


Fig. 1. Threat model.

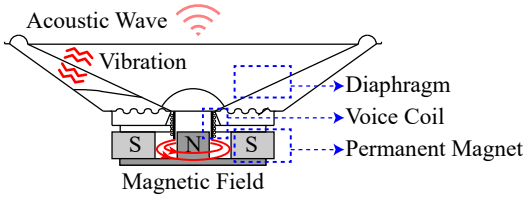


Fig. 2. Mechanical structure of an electric speaker.

we assume the adversary could not have access to the user's device physically, thus unable to eavesdrop or record with an ambient microphone. In this case, the adversary could be the Operating System (OS) providers, SDK or APP developers, who is able to pre-install a specialized APP in the user's device. Representative examples include an online shopping APP or social media APP aims to retrieve user interest from voices for precise recommendation. Considering the dense privacy in voices, it is unexpected for the user to witness their leakage to these commercial service providers. Due to the background design of mobile OSes, such a pre-installed APP could always run in the background and invoke built-in sensors to collect corresponding measurements when speakers are playing voice. Such a situation is not rare in practice, e.g., Apple being accused of listening in on the conversations between users and Siri[7], Amazon being revealed that its voice assistant Alexa is at times recording private conversations[44], and Baidu being sued for monitoring the user's phone calls[12], etc. The collected sensor measurements are further transmitted to the adversary (e.g., a cloud server) through the Internet. After receiving sensor measurements, the adversary performs a set of techniques to reconstruct the original voice. In this attack, we assume the adversary has no prior knowledge of text contents of the voices played by speakers.

3.2 Voice Leakage of Speakers

To eavesdrop the voice leaked from speakers, we first investigate the mechanical structure of speakers, and explore their potential leakage channels for eavesdropping. A speaker is an electroacoustic transducer that converts electronic signals into corresponding sounds. Due to the affordable price and satisfactory performance, electric speakers have taken the leading market share for the deployment on mobile devices currently, which are thus our main targets in this paper.

We first revisit the working principle of an electric speaker. Fig. 2 shows the mechanical structure of an electric speaker. When a user enables the speaker for voice playing, the digital signal is first converted into the current via Digital-Analog-Converter (DAC), which is further used to activate the speaker. Then, the current flows through the voice coil, which turns the coil into a temporary Alternating Current (AC) electromagnet. By interacting with the static magnetic field induced by built-in permanent magnets, different magnetic forces are generated on the metal coil. As the coil is connected with the diaphragm, various traction is introduced on the diaphragm. With this process, the magnetic energy is transformed into the mechanical energy, i.e., inducing the vibration of diaphragm. Such a vibration finally produces the acoustic waves, i.e., the voice, being human-perceived.

According to the working principle of speakers, the acoustic wave results from the diaphragm's vibration and the change of the coil's magnetic field. In other words, the voice played by speakers is highly related to the vibration and changing magnetic field, being the main channels of voice leakage. Since the speaker and other built-in sensors are on the same device, the vibration and magnetic field can be captured easily for eavesdropping.

3.3 Design Goals

To realize a training-free and universal eavesdropping attack for speakers, we define several key goals for the attack design, i.e., voice-, platform- and domain-independence.

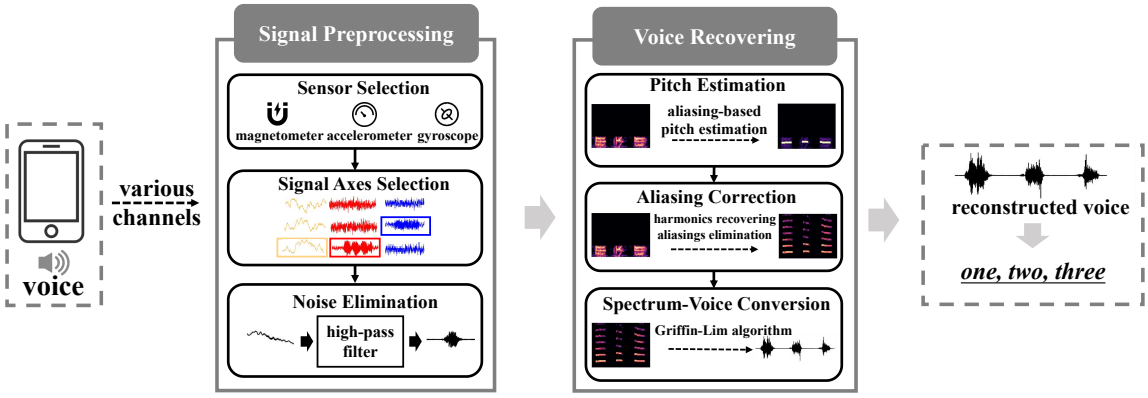


Fig. 3. Architecture of *VoiceListener*.

3.3.1 Voice-Independence. Voice-independence aims to achieve the universal eavesdropping across different texts and subjects. On the one hand, the voices played by speakers contain rich text contents, and the adversary should be capable of retrieving information from the corresponding sensor measurements under the various texts. Thus, text-independence is necessary for a universal eavesdropping attack. On the other hand, the voices are generally from different subjects, whose individual-related features are also involved in the collected sensor measurements. Therefore, a subject-independent voice recovering method is required for the attack.

3.3.2 Platform-Independence. A platform-independent attack should be performed across various devices on different sensors. A universal eavesdropping attack's targets include various victims, whose devices belong to different brands, models, and even configurations. To launch a platform-independent attack, the voice recovering should be robust to device differences. In addition, different devices are equipped with different kinds of sensors (e.g., accelerometer, gyroscope, magnetometer) with various abilities (e.g., sampling rate, sensitivity, accuracy). Thus, the cross-sensor capability is also required for the universal attack.

3.3.3 Domain-Independence. Domain in this attack refers to the environment settings and user interactions. As mentioned in Section 3.1, the user stays in the private space such as personal office, home, hotel, etc. Hence, the robustness to different environments is necessary for the attack. Furthermore, the user staying in the private space could induce unpredictable interactions with the devices in different states (e.g., sitting while holding the device, walking while placing the device on a table). Therefore, the domain-independence demands the cross-interaction capability.

4 DESIGN OF *VOICELISTENER*

In this section, we present design details of the eavesdropping attack meeting the three goals aforementioned.

4.1 Attack Overview

In order to capture the voice leakage from speakers without significant data collection efforts, we propose a model-driven method *VoiceListener*, which recovers understandable voices from undersampled sensor measurements in mobile devices for the universal eavesdropping attack. The basic idea is to explore a model-driven method, which reveals the theoretical relationship between undersampled sensor measurements and understandable voices without training a model with a large amount of collected data, so as to realize a universal eavesdropping attack.

Fig. 3 shows the architecture of *VoiceListener*, including *Signal Preprocessing* and *Voice Recovering*, which jointly realize its universality.

In the *Signal Preprocessing*, Considering the non-permission invoking and tight correlation with voice leakage, *VoiceListener* first employs three built-in sensors, i.e., accelerometer, gyroscope, and magnetometer, for eavesdropping to realize the sensor-independent eavesdropping. With these sensors, the eavesdropping attack could prevent from the interference induced by external sound sources. To accurately recover the voices, *VoiceListener* further selects the most sensitive axis from the 3-D sensor measurements based on the signal-noise-ratio. Considering noises induced by user interactions during the eavesdropping, we further introduce a noise elimination scheme based on the filter techniques for accurate voice recovering. By the sensitive axes selection and noise elimination, *VoiceListener* can mitigate the interference of environmental noises and user interactions, to realize the domain-independent eavesdropping.

Based on the raw signal collected from *Signal Preprocessing*, *Voice Recovering* applies a training-free super resolution approach to convert the undersampled sensor measurements into understandable voices. The speech contents are embedded in the pitch and corresponding harmonics in voices theoretically. Based on the theory, we design an aliasing-based subharmonic summation approach, which considers both the normal harmonics and undersampled aliasings to accurately derive the pitch. Based on the estimated pitch, *VoiceListener* reconstructs the high-frequency harmonics, and corrects the low-frequency aliasing to recover the amplitude spectrum of the voices. After that, the Griffin-Lim algorithm is further employed to derive the voices from the amplitude spectrum by iteratively estimating the phase spectrum. Since the recovering relies on the principles of voice spectrum only, the voices recovered by *VoiceListener* are not interfered by the variation of speaker tones, speech texts and device types.

4.2 Signal Preprocessing

To capture the voice leakage from speakers, we first select appropriate sensors based on the attack design goals, and then derive the most sensitive axis for each sensor as the raw signals. To further mitigate the impact of user movements, we denoise the raw signals for the following voice recovering.

4.2.1 Sensors Selection. Following the assumption of the threat model, radio-based[47] and lidar-based[40] solutions fail due to their line-of-sight requirements. Hence, we employ the built-in sensors in mobile devices for eavesdropping.

Along this direction, we revisit the latter two channels of voice leakage, i.e., vibrations and magnetic fields. Among various sensors, motion sensors (i.e., accelerometer and gyroscope) and magnetometer are able to capture the two signals respectively. Motion sensors are originally designed to sense the device's linear and angular accelerations for daily purposes (e.g., step counting), and magnetometer supports the direction identification for localization. As the intelligence demand and hardware development, these sensors are gradually capable to capture more fine-grained vibrations and magnetic field variations, leading to the feasibility of capturing the side channel of voices. Recent studies[5, 34] also demonstrate the feasibility. Toward this end, we select magnetometer, accelerometer, and gyroscope as the sensors for eavesdropping.

4.2.2 Signal Axes Selection. Typically, there are three measurements corresponding to the three axes for each selected sensor. However, the speaker eavesdropping is not necessarily required to use all of them. As speakers and built-in sensors are integrated into the same mobile device, the relative position between speakers and built-in sensors is fixed after leaving factory, and their coordinate system is also fixed for mobile operating systems (e.g., iOS and Android)[3, 16]. Since the fundamental vibration mode of the diaphragm in a speaker is the piston mode[26], the movement of diaphragm could be approximately considered as a one-dimension movement. Considering the unchanged relative position, the fixed coordinate system and the one-dimension movement

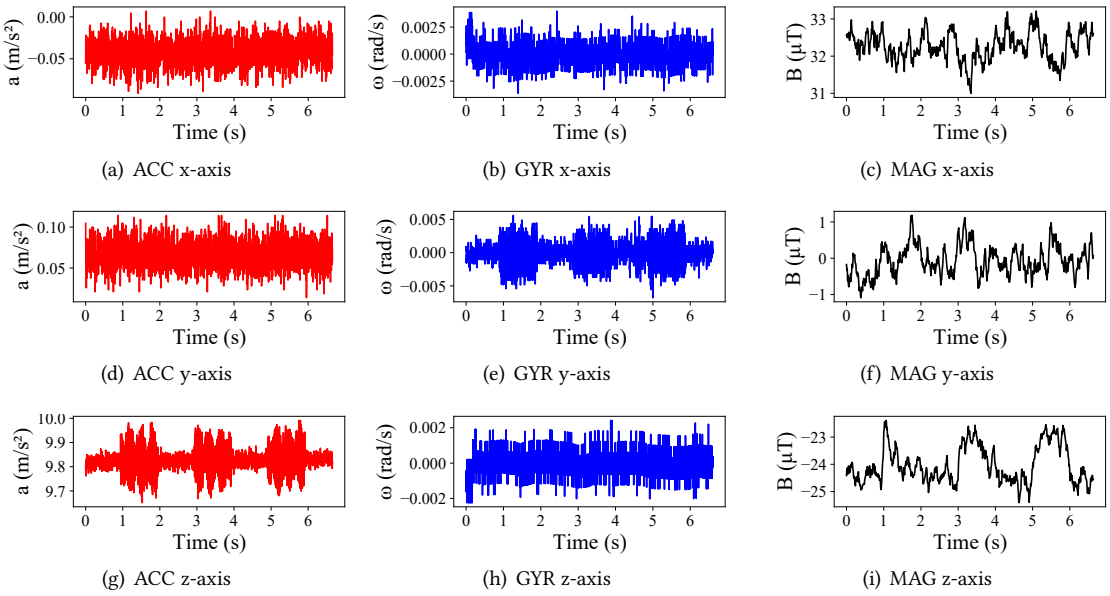


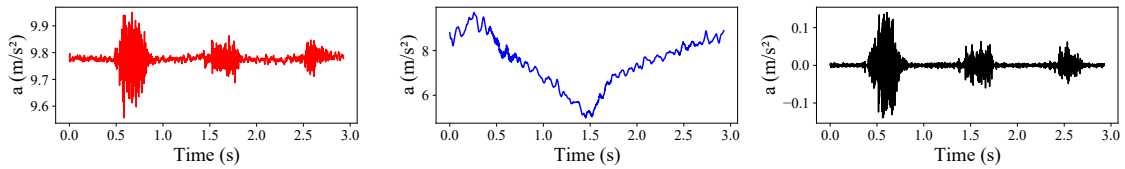
Fig. 4. Responses of accelerometer (ACC), gyroscope (GYR) and magnetometer (MAG) to 100Hz single tone signal.

of speakers, there exists a most sensitive axis of the built-in sensors for eavesdropping the speaker's voices. To validate this, we conduct an experiment where the sensor measurements are collected when the speaker plays a single tone. Fig. 4 shows sensor measurements of the accelerometer, gyroscope and magnetometer. We can see that for each sensor, there exists an axis exhibiting significant patterns for the single tone, i.e., z-axis, y-axis and z-axis for accelerometer, gyroscope and magnetometer, respectively. Even for the same device, the significant fluctuation patterns for each sensor are different. Specifically, compared Fig. 4(g) with Fig. 4(i), the pattern of accelerometer is more distinct than that of magnetometer, indicating more voice information caught by accelerometer. This is because the sampling rate of accelerometer is much higher than that of magnetometer. Another example can be shown by comparing Fig. 4(g) with Fig. 4(e). Although the accelerometer and gyroscope have same sampling rate, the pattern of accelerometer is more distinct than that of gyroscope, because gyroscope measuring angular velocity is not sensitive enough to the piston mode of speaker vibration. These results indicate a different performance impact caused by sensor heterogeneity, thus inducing the necessity of axes selection.

Since the manufacturers do not provide the relative position between the speaker and three sensors usually, we should further find the most sensitive axis for accurate eavesdropping. We utilize Signal-Noise-Ratio (SNR) to measure the sensitivity of each axis for built-in sensors. Before calculate SNR, signals should input to the high pass filter. Based on SNR, the most sensitive axis SA can be selected as

$$SA = \underset{i \in \{x,y,z\}}{\arg\max} \lg \frac{PS_i}{PN_i}, \tag{1}$$

where PS_i is the power of signal in i axis, PN_i is the power of noise in i axis. Both powers could be measured offline before the attack is issued. After that, *VoiceListener* could select the most sensitive axis for each sensor, and collect the raw data for further eavesdropping. Specifically, when a new device is used for eavesdropping, *VoiceListener* gets the signal by measuring the sensor outputs when the device is playing a single tone, and



(a) Raw signal in table setting (b) Raw signal in handhold setting (c) Filtered signal in handhold setting

Fig. 5. Accelerometer's responses to the given audio under different settings.

obtains the noise when the device keeps silent. Then, based on (1), we could derive the SNR for each axis of sensors. Taking the accelerometer in Fig. 4 as an example, we can derive the SNRs of x-axis, y-axis and z-axis are 0.84dB, 0.37dB and 8.57dB. Hence, we select z-axis with the largest SNR value as the most sensitive axis for eavesdropping.

4.2.3 Noise Elimination. As mentioned in Section 3.1, the user's behavior is not constrained in our threat model. Thus, when *VoiceListener* eavesdrops on the speaker with built-in sensors, there are probably significant interferences on the sensor measurements induced by the users' behaviors, such as walking, shaking, etc. To visualize such an interference, we conduct an experiment to measure the sensor response under different settings. For simplicity, we only take accelerometer' measurements as an example. In the experiment, we collect the accelerometer measurements of a Huawei P40 while the speaker playing the voices (i.e., speaking one, two, three), under the table setting (i.e., placing the device on the table statically) and handhold setting (i.e., holding the device and walking). Fig. 5(a) and Fig. 5(b) show the response of accelerometer in the table setting and handhold setting respectively. We can find that compared with the table setting, the accelerometer response under the handhold setting exhibits significant variations for the movements, shadowing the signal induced by the speaker broadcasting. Hence, we should remove the interference before the voice recovering from the sensor measurements.

Fortunately, compared with the voice leakage in high frequency above 85Hz, human movements are in low-frequency band below 80Hz[5]. Thus, we utilize an 8^{th}-order digital high-pass filter to eliminate the interference of user's movements on the captured signals. Considering motion sensors are more sensitive to human movements than the magnetometer, we set the filter's cut-off frequency to 80Hz and 20Hz for motion sensors and the magnetometer respectively. Fig. 5(c) shows the filtered signal in the handhold setting, we find the voice leakage from the speaker's vibrations has been separated.

4.3 Voice Recovering

According to the Nyquist theorem, the frequency band of these sensors only can reach 250Hz. However, the audio can be understood by humans only if its frequency band approaches 1,000Hz. This is because the identification of vowels is necessary for understanding a voice, which requires most F1 formants to remain in the voice. And almost all the F1 formants of vowels are below 1,000Hz[8]. Hence, our goal is to reconstruct the 1000Hz voices from the 250Hz undersampled sensor measurements. Intrinsically, this problem is similar to the audio super resolution, which aims to extend the bandwidth of transmitted voices from a band-limited audio.

However, traditional audio super resolution algorithms could not work well on the narrow-band sensor measurements. The core rationale lies on the signal aliasing elimination on the narrow-band sensor measurements. Traditional audio super resolution methods are designed to recover 8,000Hz voices from 4,000Hz telecommunication signals. Considering the rich audio information embedded underlying the large bandwidth (i.e., 4,000Hz), these methods usually assume there is no signal aliasing of the telecommunication signals. Under this assumption,

existing methods design many mechanisms, including Codebook Mapping[13, 38], and statistical models (e.g., GMM-based method[36, 37], HMM-based method[23, 48]), to directly recover the voices. However, due to the low sampling rate of embedded sensors, the collected sensor measurements are only within 250Hz, which are mixed with many downsampled voice harmonics, thus causing significant aliasing. Hence, traditional audio super resolution methods cannot be directly applied to recover voices from sensor measurements. To handle this problem, we propose an aliasing-corrected super resolution method for recovering the narrow-band sensor measurements to understandable voices.

4.3.1 Pitch Estimation. Typically, an understandable voice consists of a pitch and several harmonics whose frequencies are integral multiples of the pitch's frequency. Thus, the energy sum of the pitch and its corresponding harmonics could be significantly larger than that of other signals and their corresponding ones with integral multiple frequencies. Based on the principle, we employ the subharmonic summation[21], which accumulates the energy of each signal and its corresponding harmonics, and derives the signal with the largest energy summation value as the pitch. Due to the undersampled aliasing in our collected sensor measurements, the typical subharmonic summation cannot be directly applied to estimate the pitch in the signals. To solve this problem, we propose an aliasing-based subharmonic summation, which takes both the linear harmonics and non-linear aliasing into consideration for accurate pitch estimation.

Different from wide-band voices (i.e., $0 \sim 4000Hz$), the narrow-band sensors measurements are only in the range of $[0, 250]Hz$, involving significant aliasings whose frequency is non-linear to that of pitch. Aliasing is intrinsically a voice harmonic in the high-frequency band, which is undersampled to the low-frequency band, interfering with the pitch estimation. Fortunately, such an undersampling embeds the relationship between original high-frequency harmonic f and undersampled low-frequency aliasing $A(f)$, i.e.,

$$A(f) = |f - \lfloor \frac{f}{SR} + 0.5 \rfloor \times SR|, \tag{2}$$

where SR is the sampling rate of built-in sensors. Based on (2), we elaborate on the aliasings as additional harmonics for pitch estimation. The core of our proposed aliasing-based subharmonic summation is to derive the possibility $H(t, f)$ that the pitch's frequency is f, i.e.,

$$H(t, f) = \sum_{k=1}^{n} M(t, kf) + \sum_{k=n+1}^{m} M(t, A(kf)), \tag{3}$$

where $M(t, f)$ is the Short-Time Fourier Transform (STFT) spectrum of the signal at time t and frequency f, $k \in N^+$ is the times variable, n and m are the indexes for cut-off frequencies of the undersampled signal and understandable voice respectively, derived as $n = \lfloor \frac{SR}{2f} \rfloor$ and $m = \lfloor \frac{F_c}{f} \rfloor$, in which F_c is the cut-off frequency of understandable voices. Considering frequencies above $1250Hz$ are not necessary for the pitch estimation[21], F_c could be set as $1250Hz$. In (3), the first term accumulates the low-frequency harmonics, and the second term's summation includes the undersampled high-frequency aliasings. Combining both terms, the pitch could be estimated more accurately. Based on Eq. (3), the pitch could be derived as

$$f_p = \underset{85Hz<f<255Hz}{\arg\max} \; H(t, f), \tag{4}$$

where the frequency range $[85, 255]Hz$ is the typical pitch frequency for adults.

4.3.2 Aliasing Correction. Based on the estimated pitch, we further reconstruct the voice spectrum from sensor measurements. Typical super solution approaches leverage the frequency relationship between pitch and harmonics to recover narrow-band signals to wide-band ones. However, due to the aliasing induced by the undersampled process, these kinds of solutions cannot be directly applied to reconstruct the voice spectrum from sensor measurements. Hence, we propose an aliasing-corrected super resolution algorithm, as shown in 1.**??**

Algorithm 1 Aliasing Correction Algorithm

Input: M_{old}: spectrum matrix of sensor measurement, F_{begin}: lowest frequency of reconstructed spectrum, F_{end}: the highest frequency of reconstructed spectrum.
Output: M_{new}: spectrum matrix of reconstructed voice.

```
1:  F_begin ← 0Hz; F_end ← 1000Hz;
2:  for each frame t of spectrum do
3:      f_p ← arg max_{85Hz<f_c<255Hz} H(t, f_c);
4:      for f ∈ [F_begin, F_end] do
5:          if f ∈ [SR/2, F_end] then
6:              if f mod f_p = 0 then
7:                  Calculate A(f) based on Eq. (2);
8:                  M_new(t, f) ← M_old(t, A(f));
9:          else
10:             if f mod f_p = 0 then
11:                 M_new(t, f) ← M_old(t, f);
12:             else
13:                 M_new(t, f) ← 0;
14: return M_new
```

The proposed algorithm mainly completes two tasks, i.e., recovering the high-frequency harmonics and eliminating the low-frequency aliasings. *VoiceListener* first estimates the pitch as mentioned in Section 4.3.1. Based on the integral multiple relationship between the frequencies of pitch and harmonics, *VoiceListener* regenerates harmonics in the high frequency band (i.e., $[\frac{SR}{2}, F_{end}]$), as shown in Line 5~8. For the amplitude of the harmonic with frequency f, we employ its corresponding aliasing $A(f)$'s amplitude $M_{old}(t, A(f))$ as the harmonic's amplitude $M_{new}(t, f)$. After that, *VoiceListener* searches the undersampled aliasing in the low frequency band (i.e., $[0, \frac{SR}{2}]Hz$) for voice correction as shown in Line 9~13. Specifically, we remain the amplitude $M_{old}(t, f)$ if f is a valid harmonic's frequency, and elaborate it on the reconstructed spectrum $M_{new}(t, f)$. Otherwise, the corresponding amplitude could be set as $M_{new}(t, f) = 0$ for spectrum reconstruction. With the aliasing correction, we reconstruct the wide-band voice spectrum from the narrow-band sensor measurements.

4.3.3 Spectrum-Voice Conversion. To recover an understandable voice, *VoiceListener* further converts the amplitude spectrum to voices. Revisiting the Fourier theorem, a time-domain voice could be converted to an amplitude spectrum and a phase spectrum by the Fourier transformation. To recover the voice nondestructively, *VoiceListener* is required to estimate the phase spectrum based on the amplitude spectrum. We employ the Griffin-Lim algorithm[19] to recover the voices merely from the amplitude spectrum.

Given the amplitude spectrum M, Griffin-Lim algorithm aims to estimate a signal whose amplitude spectrum is more similar to the given M. The algorithm first randomly generates a phase spectrum P_0. Combined the given M and P_0, we perform inverse STFT (iSTFT) to recover a voice x_0. After that, the algorithm performs STFT on the recovered x_0 to obtain a new amplitude spectrum $\hat{M}$ and phase spectrum P_1. Due to the difference between the ground truth of phase spectrum P and the estimated P_0, there exists a significant variance between M and $\hat{M}$. Hence, we only remain the phase spectrum P_1 while replacing the randomly generated P_0 in the next iteration of the algorithm. By analogy, we iteratively calibrate the phase spectrum P_i by the iSTFT and STFT operations until the correspondingly generated amplitude spectrum $\hat{M}$ is similar to the given M. With the iterations, *VoiceListener* could generate the time-domain voices from the recovered amplitude spectrum.

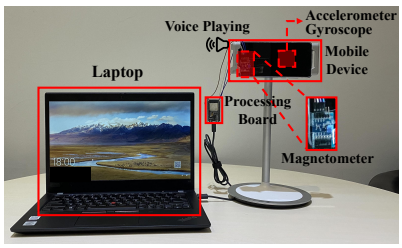


Fig. 6. Experimental setup.

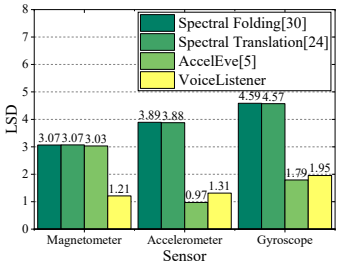


Fig. 7. LSD of *VoiceListener* and three baselines[5, 24, 30] from different sensors.

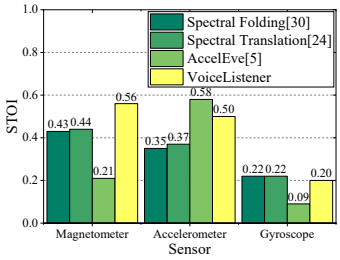


Fig. 8. STOI of *VoiceListener* and three baselines[5, 24, 30] from different sensors.

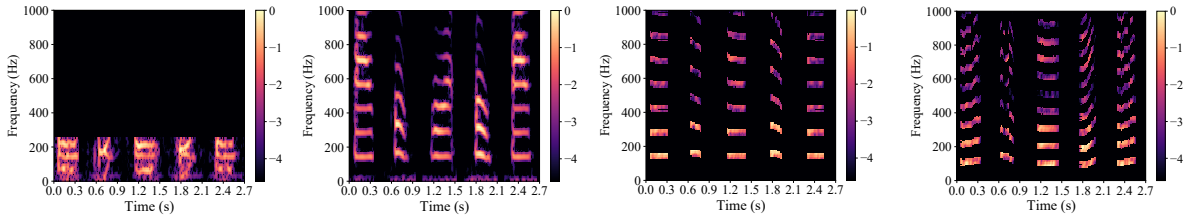
5 EVALUATION

In this section, we evaluate the performance of *VoiceListener* on the quality of recovered voice.

5.1 Experimental Setup & Methodology

We implement *VoiceListener* on a platform as shown in Fig. 6. In the platform, we employ 10 smartphone models, i.e., Huawei P40, Huawei P20 Pro, Honor V30, OPPO Find X2, Mi 10, Motorola Edge S, Redmi k30 Pro, Vivo IQOO 3, Samsung S20 Ultra and Samsung S20, as the victims' mobile devices for voice broadcasting by speakers. On the other hand, for the sensing front-end, we directly collect the sensor measurements of both accelerometer and gyroscope from the three devices' motion sensors. As for the magnetometer's measurements, due to the low sampling rate (i.e., around 100Hz), we turn to use an external magnetometer MMC3416xPJ[33] attached on the device to simulate built-in magnetometers for data collection. Note that many built-in magnetometers (e.g., MMC5603NJ) are actually capable with more than 500Hz sampling rates, and have been integrated in various smartphone models (e.g., Vivo S1 Pro and Huawei STK-AL00). But due to limited computational capability of other components in the device (e.g., CPU), the sampling rate of built-in magnetometers is constrained. Following the evolution route of accelerometer and gyroscope (i.e., 200Hz in 2014[34] while 500Hz in 2020[5]), it is highly expected that the sampling rate of magnetometer would also be improved to its designated one. The maximum sampling rates of the 10 smartphones' built-in sensors are in the range of $[392, 500]Hz$. The collected data is then transmitted to the processing back-end, i.e., a ThinkPad X13 laptop, on which the sensor measurements are used for voice recovering by *VoiceListener*.

Since digits are the basic components of various private and sensitive information (e.g., PIN, ID number, CVS code of credit cards), we further employ AudioMNIST[6] as the voice sources played by speakers, among which 6000 voices from 20 speakers with different genders and ages among 23~41 are selected. Moreover, considering AudioMNIST[6] containing digits only in the voice samples, we further supplement a more complicated dataset, i.e., TIMIT[15], to enrich the diversity of voice sources, for evaluation of *VoiceListener*. The experiments are repeated in three different environments, i.e., a lab, a dorm, and a dining hall, of different sizes and furniture layouts. And the measured background noise levels of the three environments are 45.3dBSPL (dB of Sound Pressure Level), 48.9dBSPL and 74.6dBSPL, respectively. We repeat the experiments under three different user activities when playing voices with speakers, i.e., table setting (placing the device on a table), handhold setting (holding the device and walking), tap setting (tapping on the device screen continuously). In each experiment, we play concatenated voices including 40~50 single voice samples with the speaker, and collect corresponding sensor measurements from built-in sensors via Android APP *phyphox* and the external magnetometers respectively. In total, we collect 31,800 measurements as samples for the evaluation of *VoiceListener*.



(a) Magnetometer measurements (b) Ground-truth voices (c) Recovered voices by *VoiceListener* with aliasing-based pitch estimation (d) Recovered voices by *VoiceListener* with subharmonic summation[21]

Fig. 9. Spectrums of magnetometer measurements, ground-truth voices and recovered voices.

To evaluate the performance of voice recovering, we employ a widely-used metric for voice similarity comparison, i.e., Log-Spectral Distance (LSD)[18] that measures the magnitude spectrum similarity between recovered voices and ground truth, as defined by $LSD(\hat{x}, x) = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{X}(t,k) - X(t,k))^2}$, where $\hat{X}(t,k)$ and $X(t,k)$ are the log-spectral power of reconstructed signal $\hat{x}$ and ground-truth voice signal x at the t^{th} time frame and k^{th} frequency band, respectively. In addition, we employ Short-Time Objective Intelligibility (STOI)[43] to evaluate the intelligibility of recovered voices, which ranges from 0 to 1 with higher STOI indicating better intelligibility. Also, another metric Word Accuracy evaluates the performance of speech recognition, which is defined as $WA = \frac{C}{N}$, where C is the number of correct words predicted by the model, and N is the total number of words.

5.2 Overall Performance

We first evaluate the performance of *VoiceListener* on voice recovering. To validate the effectiveness of *VoiceListener*, we implement three State-Of-The-Art (SOTA) methods as baselines, i.e., two basic audio super resolution algorithms *Spectral Folding*[30] and *Spectral Translation*[24], and a machine learning-based method *AccelEve*[5] in the evaluation. For a fair comparison, we collect the sensor measurements under the default setting, i.e., with voices from the same 10 persons under the same device (Huawei P40), environment (lab), and user interaction (table setting). Different from *VoiceListener* and two audio super resolution algorithms, the SOTA *AccelEve* requires additional data for training. Hence, we pre-train *AccelEve* model with 4,000 accelerometer measurements, and evaluate all the four methods with other 1,000 measurements from each sensor respectively.

Fig. 7 shows the LSD of *VoiceListener* and the three SOTA methods[5, 24, 30] from different sensors. For magnetometer, we can see that LSD of *VoiceListener* is far less than other three methods around 60.0%, indicating a satisfactory performance of *VoiceListener* in voice recovering. On the other hand, for accelerometer and gyroscope, it also can be observed that *VoiceListener* could achieve LSDs less than the two audio super resolution

Table 2. Statistical test results between *VoiceListener* and baselines.

Compare with VoiceListener	Magnetometer		Accelerometer		Gyroscope	
	K-W test	Levene test	K-W test	Levene test	K-W test	Levene test
Acceleve	2.40×10^{-34}	4.00×10^{-2}	3.66×10^{-40}	2.90×10^{-3}	1.54×10^{-19}	1.50×10^{-1}
Spectral Folding	4.77×10^{-55}	5.60×10^{-1}	4.58×10^{-66}	5.45×10^{-34}	4.79×10^{-67}	8.96×10^{-9}
Spectral Translation	7.83×10^{-66}	3.20×10^{-3}	1.06×10^{-66}	4.18×10^{-25}	4.79×10^{-67}	1.20×10^{-13}

algorithms above 65%. Though *VoiceListener*'s LSD is higher compared with *AccelEve*, their difference is only 26.2% and 9.1% respectively, indicating a competitive performance of *VoiceListener* with the machine learning-based solution. Moreover, we find *VoiceListener* could achieve similar LSDs under different sensors, validating its sensor independence in our design goals, which significantly outperforms machine learning-based *AccelEve*. Also, it can be observed that the LSDs under gyroscope are significantly larger than that under two other sensors. This is because the vibration of diaphragm could be approximately regarded as a one-dimension movement, while the gyroscope is sensitive to the angular acceleration, leading to less information in the sensor measurements. Fig. 8 shows the STOI of *VoiceListener* and the three baselines[5, 24, 30] from different sensors. For magnetometer, the STOI of *VoiceListener* is about 166% higher than that of *AccelEve*, and about 27% higher than the two audio super resolution methods. For accelerometer, the STOI of *VoiceListener* is slightly lower than that of *AccelEve*, but much higher than two audio super resolution methods. For gyroscope, all the voice reconstruction methods get low STOIs, which is consistent with previous results. We further validate the statistical difference on the results by Kruskal-Wallis H-test and Levene test, in which we first sample results from the LSDs of *VoiceListener* and three baselines, and then conduct the tests on them, respectively. Table 2 shows the p-value in each statistical test. If p-value is lower than $\alpha = 0.05$, there exist significant differences on population median and variance between *VoiceListener* and baselines. From the table, we can see that under the three sensors, *VoiceListener* could achieve the significant statistical difference with all the three baselines. These results indicate that *VoiceListener* could achieve satisfactory performance under different sensors. To visualize the recovering performance of *VoiceListener* more straightforwardly, we further present an example of magnetometer measurements, the corresponding ground-truth voices and the recovered voices, as shown in Fig. 9. Compared Fig. 9(a) with 9(b), we can find that the bandwidth of sensor measurement is only 1/4 of the raw voices, embedding significant aliasings. But after the voice recovering as shown in Fig. 9(c), most high-frequency components are recovered and low-frequency aliasings are corrected, thus leading to a low LSD of 1.275 between the recovered and ground-truth voices.

Considering that pitch estimation is the critical fundamental in voice recovering, we further conduct two experiments to validate its effectiveness. First, we compare the performance of aliasing-based pitch estimation with the subharmonic summation[21]. Table 3 shows the LSDs of *VoiceListener* with different pitch estimation methods. We can observe that the performance of our aliasing-based pitch estimation is better than subharmonic summation about 15.8% on magnetometer, indicating the improvements of our proposed pitch estimation method. To further illustrate the difference, we present the spectrums of two recovered voices under our pitch estimation and subharmonic summation methods, as shown in Fig. 9(c) and 9(d), respectively. It can be seen that around 1.9s, the recovered pitch and harmonics by subharmonic summation significantly differs from ground truth voices, indicating an incorrect estimation result. On the other hand, considering the pitch frequency for male and female lying on different bands (85-180Hz and 165-255Hz respectively), this experiment validates the sensitivity of the candidate frequency ranges on our attacks' universality. From Table 4, we observe that the LSDs of *VoiceListener* are a little less than that under a narrower candidate frequency range. This is because a smaller size of the frequency range can narrow down the number of candidates, leading a higher accuracy of pitch estimation. This result indicates that if the adversary could introduce more prior knowledge of voice source's gender, to improve the performance of voice recovering.

Table 3. LSDs of *VoiceListener* with different pitch estimation methods.

Pitch estimation methods	**Magnetometer**	**Accelerometer**	**Gyroscope**
aliasing-based pitch estimation	1.06	1.20	1.76
subharmonic summation[21]	1.26	1.27	1.89

5.3 Impact of Device Models

With the development of mobile devices, more manufacturers participate in the market, leading to various brands and series of devices. To dominate different market levels and target customers, the mobile devices with different brands and series integrate different hardware, including CPU, sensors, etc., whose technical indicators are varied. For example, the sampling rates of the 10 devices in our experiments are in the range of $[392, 500]Hz$. These differences introduce diverse impacts on collected sensor measurements, which is the rationale of learning-based solutions' performance degradation. Hence, we evaluate the performances of *VoiceListener* and *AccelEve* on different device models. To ensure fairness, *VoiceListener* and *AccelEve* in the evaluation only know the sampling rate of sensors, without any other parameters of sensors (e.g., the frequency response) as the prior knowledge.

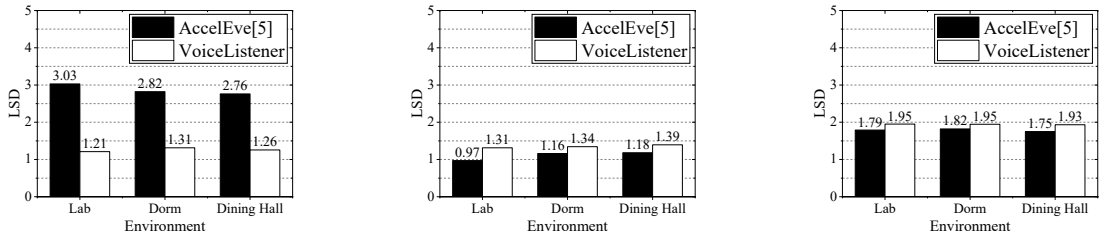
Table 5 shows the LSD of *VoiceListener* and *AccelEve* on different device models. For magnetometer, we can see that LSDs of *VoiceListener* are all less than those of *AccelEve* above 50% on each device. This is because the magnetometer measurements exhibit difference with the accelerometer measurements (i.e., the training data for *AccelEve*), inducing the performance degradation on voice recovering. But for accelerometer, not all LSDs of *VoiceListener* are higher than that of *AccelEve*. Except for Huawei P40 and Huawei Honor V30, all the LSDs of *VoiceListener* are actually lower than those of *AccelEve* on other 8 devices. Moreover, the CVs for magnetometer and accelerometer of *VoiceListener* are 0.05 and 0.06, which are less than 0.06 and 0.23 of *AccelEve* respectively. All these results indicate that *VoiceListener* outperforms *AccelEve* for better robustness against device variation. This result demonstrates the device independence of *VoiceListener*, outperforming machine learning-based *AccelEve*. For gyroscope, due to significant noises introduced by the measurements, both *VoiceListener* and *AccelEve* exhibit poor performance.

Table 4. LSDs of *VoiceListener* under different candidate frequency ranges of pitch.

Voice type	Candidate frequency range of pitch	Magnetometer	Accelerometer	Gyroscope
Male voice	85Hz to 255Hz	1.22	1.26	2.23
	85Hz to 180Hz	1.10	1.19	2.14
Female voice	85Hz to 255Hz	1.13	1.28	1.97
	165Hz to 255Hz	1.12	1.24	1.83

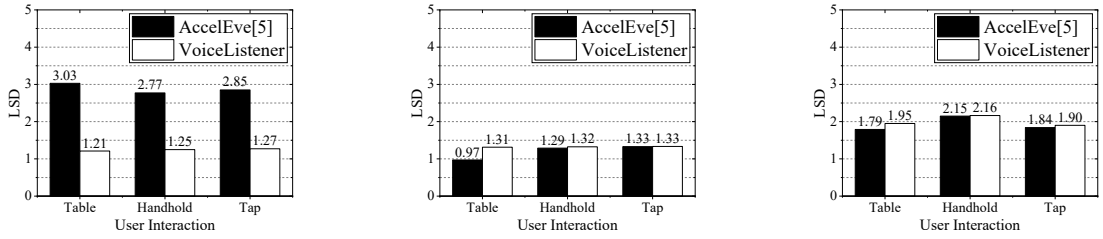
Table 5. LSD of VoiceListener and AccelEve (i.e., Our and SOTA[5]) on different device models.

Model	Magnetometer		Accelerometer		Gyroscope	
	Our	SOTA[5]	Our	SOTA[5]	Our	SOTA[5]
Huawei P40	1.21	3.03	1.31	0.97	1.95	1.79
Huawei P20 pro	1.21	2.69	1.16	1.41	1.70	1.95
Huawei Honor V30	1.31	2.75	1.34	1.28	1.76	2.18
OPPO Find X2	1.17	2.79	1.27	1.86	2.47	1.81
Mi 10	1.39	2.52	1.34	2.15	2.54	1.92
Motorala edge S	1.32	3.15	1.44	2.39	2.49	2.35
Redmi k30 pro	1.21	2.84	1.47	2.01	2.37	2.16
vivo IQOO3	1.27	2.68	1.43	1.84	2.45	1.83
Samsung S20 Ultra	1.35	2.79	1.30	1.67	2.33	2.51
Samsung S21	1.25	2.64	1.43	2.08	2.09	2.04



(a) Magnetometer (b) Accelerometer (c) Gyroscope

Fig. 10. LSD of *VoiceListener* and *AccelEve* in different environments.



(a) Magnetometer (b) Accelerometer (c) Gyroscope

Fig. 11. LSD of *VoiceListener* and *AccelEve* under different user interactions.

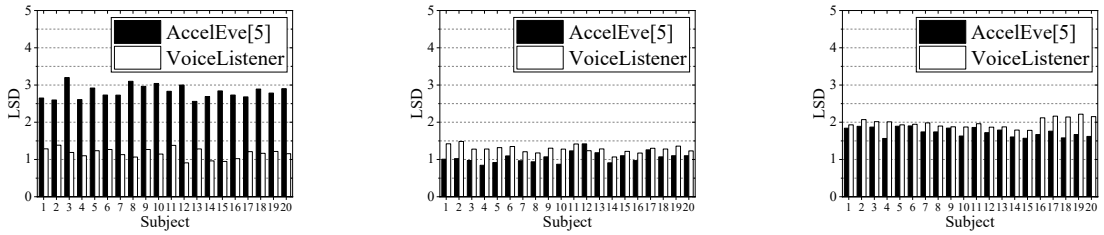
5.4 Impact of Environments and User Interactions

Since *VoiceListener* relies on the built-in sensors for raw data collection, the domain of the victim users' devices, i.e., the ambient environments and user interactions, would affect the eavesdropping. On the one hand, due to voices broadcasting from speakers lying on the human-audible range, the ambient noises in the environments may interfere the eavesdropping (e.g., high-level noises could degrade the microphone-based eavesdropping). On the other hand, the interactions between victim users and their mobile devices may also degrade the eavesdropping performance, because the built-in sensors used in *VoiceListener* (i.e., magnetometer, accelerometer, gyroscope) are originally designed to capture such behaviors. Hence, we evaluate the performance of *VoiceListener* in different environments (i.e., a lab with 45.3*dBSPL* noise, a dorm with 48.9*dBSPL* noise, and a dining hall with 74.6*dBSPL* noise), and under different user interactions (i.e., table setting without interactions, handhold setting with minute interactions, and tapping setting with significant interactions).

Fig. 10 shows LSDs of *VoiceListener* and *AccelEve* in different environments. We can see that *VoiceListener* shows less variation than *AccelEve* in the three different environments. Specifically, the CVs of LSD for *VoiceListener* under the three environments are 17.9%, 71.6% and 74.8% lower than those of *AccelEve* for the three sensors, respectively, validating better robustness of *VoiceListener* in various environments. Also, as shown in Fig. 11, *VoiceListener* still exhibits higher robustness than *AccelEve* under different user interactions, with 47.0%, 95.5% and 31.4% lower CVs of LSD for the three sensors, respectively.

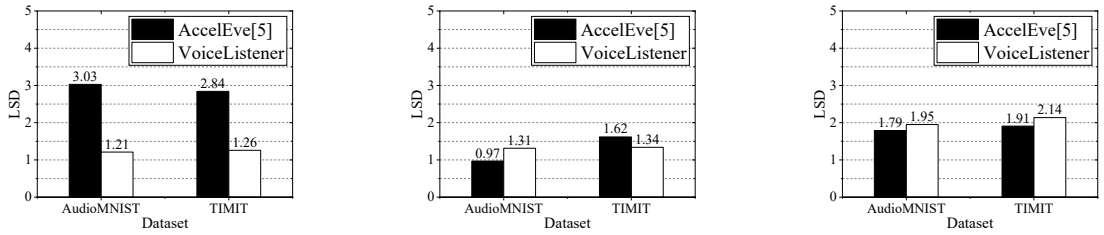
5.5 Impact of Subjects

During the voice interactions, the subject of broadcasted voices could be different persons or even an AI-synthesized speaker. And the texts of voice interactions are varied and unpredictable. Hence, voice-independence,



(a) Magnetometer (b) Accelerometer (c) Gyroscope

Fig. 12. LSD of *VoiceListener* and *AccelEve* under different subjects.



(a) Magnetometer (b) Accelerometer (c) Gyroscope

Fig. 13. LSD of *VoiceListener* and *AccelEve* under different datasets.

i.e., subject-independence and text-independence, is necessary for a universal eavesdropping attack. In this section, we evaluate the performance of *VoiceListener* under different subjects. Fig. 12 shows the performances of *VoiceListener* and *AccelEve* under different subjects. It can be observed that *VoiceListener* still exhibits less variations than *AccelEve* under different subjects, but the difference is rather small. Specifically, the CVs of LSD of both *VoiceListener* and *AccelEve* are lower than 0.13, indicating that both methods realize good subject-independence. This is because the evaluation dataset AudioMNIST only contains single digits, whose harmonic structures are much simpler than those of complex speech words and sentences with richer phonemes.

5.6 Impact of Datasets

The aforementioned experiments are all under the AudioMNIST[6] dataset, which only involves voice samples of single digits. Considering more complex speech texts in voice interactions, we further evaluate the performance of *VoiceListener* on TIMIT[15] dataset, which contains complex words and sentences with all the phonemes used in normal interactions. In this experiment, we select 37 persons from TIMIT[15], each of which employs 10 sentences of voice samples for the evaluation. Fig. 13 shows LSDs of *VoiceListener* and *AccelEve* under different datasets. For magnetometer, the LSD of *VoiceListener* is less than *AccelEve* about 55%, remaining the leading performance of *VoiceListener*. On the other hand, for accelerometer, when transferring to TIMIT, *AccelEve* suffers from more than 65% performance degradation, which is far worse than *VoiceListener* (only 2.0%). This is because *VoiceListener* is a training-free solution, i.e., requires no prior knowledge of speech texts, thus more robust under different datasets. For gyroscope, due to the significant noises embedded in the sensor measurements, both *VoiceListener* and *AccelEve* suffers from performance degradation, with similar standard deviation under different datasets.

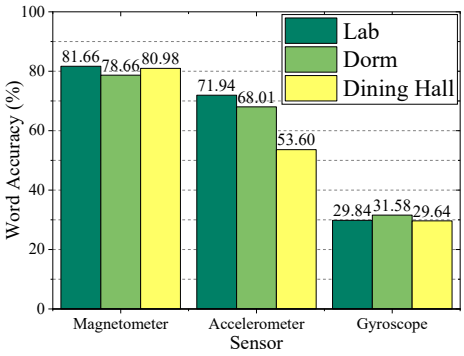


Fig. 14. Word accuracy of VoiceListener in different environments.

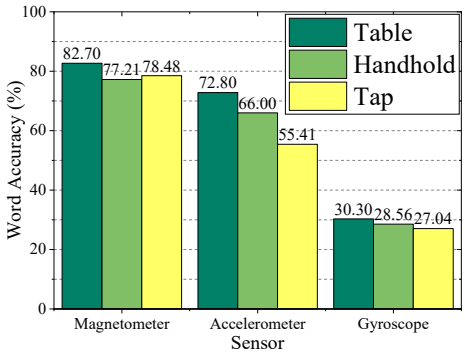


Fig. 15. Word accuracy of VoiceListener under different user interactions.

5.7 Performance on Speech Recognition

We further evaluation the performance of *VoiceListener* on speech recognition. In this experiment, we implement a spatial-temporal deep neural network to realize the speech recognition for performance evaluation.

5.7.1 Implementation. Specifically, the input for the neural network is Mel-Frequency Cepstral Coefficient (MFCC) of recovered voice, which are derived by a 20ms Hanning window with the step of 10ms, and combined with their first and second derivatives as a 3-channel tensor. After that, we feed the extracted features to spatial-temporal model, i.e., Convolutional Long short-term memory fully connected Deep Neural Network (CLDNN) and Connectionist Temporal Classification (CTC), for speech recognition. In particular, the model consists of three residual blocks as the spatial layer for frequency convolution, each of which 3×3 filters and ReLU activation are applied, followed by a 2×2 max pooling with a stride of 2. Then, bidirectional RNN with LSTM units is selected as the temporal layer, whose output values are concatenated and fed to a fully connected layer for classification. Finally, softmax is used to obtain the posterior probability Y. The loss function is defined as:

$$h(X) = \arg\max_{(X,Z)\in\mathcal{D}} p(Z|X). \tag{5}$$

where $p(Z|X) = \sum_{\pi\in\Phi^{-1}(Z)} \prod_{t=1}^{T} y_{(t,\pi_t)}$, Φ is a mapping from predicted label sequence π to ground truth label sequence Z, and $y_{(t,\pi_t)}$ represents the posterior probability corresponding to the π_t^{th} label at time t. We further adopt SGD with momentum as the optimizer to speed up convergence, and a dropout strategy is applied to alleviate the issue of overfitting.

5.7.2 Performance. To demonstrate the sensor-independence of VoiceListener, we involve two State-Of-The-Art (SOTA) learning-based solutions, i.e., Gyrophone[34] and AccelEve[5], which are designed for gyroscope and

Table 6. Word accuracies of *VoiceListener* and SOTA models.

Sensor	**Gyrophone**[34]	**AccelEve**[5]	**VoiceListener**
Magnetometer	N/A	N/A	82.7%
Accelerometer	N/A	78.0%	72.8%
Gyroscope	26.0%	N/A	30.3%

accelerometer respectively, for evaluation. Table 6 shows the word accuracy of *VoiceListener*, Gyrophone and AccelEve, respectively. It can be seen that the proposed *VoiceListener* could achieve the word accuracy of 82.7%, 72.8%, 30.3% on magnetometer, accelerometer and gyroscope respectively. Though the word accuracy under gyroscope is lower than that under the other two sensors due to its sensitivity, *VoiceListener* still outperforms Gyprophone with an improvement of 4.3%. Moreover, compared with more recent AccelEve, our proposed solution can achieve competitive performance with only 5.2% degradation. Considering the sensor-dependence of *VoiceListener*, our solution can be deployed on a magnetometer and outperforms the accelerometer-based solution with an improvement approaching 5%. These results demonstrate the feasibility and effectiveness of *VoiceListener* on sensor-dependent voice eavesdropping.

We further evaluate the performance of *VoiceListener* on speech recognition under different domains. Fig. 14 shows the word accuracy of *VoiceListener* in different environments. Similar small variations of word accuracy can be also observed. Specifically, the CVs of word accuracy are only 1.6% and 2.8% for the magnetometer and gyroscope respectively, which that for the accelerometer is 12.2%, which is greater than others. This is because the dining hall environments introduce more inevitable vibrations (such as someone walking around), which is easily captured by the accelerometer, thus leading to a larger CV. If only considering the lab and dorm environments, the CV is 2.8%, closing to that under another two sensors. Moreover, Fig. 15 shows the word accuracy of *VoiceListener* under different user interactions. It can be observed similar small variation of word accuracy with CVs of only 2.9% and 4.6% for magnetometer and gyroscope respectively. For accelerometer, despite the tap interaction that introduces significant interference, its CV of word accuracy is 4.9%, closing to that for another two sensors. All of these results further validate satisfactory robustness in different environments and user interactions. We also evaluate the performance of *VoiceListener* with speech recognition on TIMIT. The result shows that the phoneme accuracies on magnetometer, accelerometer and gyroscope are all lower than 10%, which is not good as that under AudioMNIST. The reasons are two-fold. First, different from simple ten-class classification in AudioMNIST, the speech recognition is an open-set task to predict diverse phoneme sequences and word combinations. On the other hand, to realize a strong speech recognition model, a large number of high-quality speech datasets are essential. For example, the dataset LibriSpeech includes 360 hours of clean English speech for model training. However, our collected dataset only contains 2 hours undersampled sensors measurements, which hardly supports us to train a robust speech recognition model.

6 DISCUSSION

6.1 Prospective Applications

Except for launching eavesdropping attacks, the core part of *VoiceListener*, i.e., the aliasing-corrected super resolution method, may also be employed for positive applications.

Lossy Voice Compression. In this work, *VoiceListener* realizes a universal audio super resolution from undersampled voice signals, which could serve as a decompression tool for audio data transmission applications. Specifically, the transmitter can compress voices through downsampling operations so as to transmit them at a larger data rate with the same bandwidth. Then the receiver can apply *VoiceListener* to decompress the voice to recover the lost high-frequency components. Although such a compression process is lossy, it is expected to facilitate some scenarios pursuing real-time response rather than high fidelity, e.g., live voice call.

Adversarial Example Defense. Adversarial example attacks [10, 11, 28] are a serious threat to machine learning-based audio systems, which inject imperceptible perturbations with subtle energy on benign input to spoof neural networks. Since *VoiceListener* aims to reconstruct intelligible voices from undersampled signals, this will restore valuable high-energy pitch and harmonics while discarding the low-energy components in voices. Such a downsample-upsample transformation is also a promising defense tool as other transformation-based

methods [22] (e.g., Re-quantization, frequency filtering, Mel transformation), for purifying malicious adversarial perturbations in voice.

6.2 Countermeasures

In this subsection, we discuss several potential countermeasures of *VoiceListener*.

6.2.1 Permission Constraint. In modern mobile operating systems (e.g., Android and iOS), built-in sensors can be directly accessed by third-party APPs without any permission from users. As demonstrated by our work, all these seemingly harmless sensors could be exploited for speaker eavesdropping without any training effort, further validating the vulnerability of these sensors combined with existing literatures[5, 20, 34, 49]. Though recent operating system updates have elaborated on the sensor usage notification for user awareness, users could not directly control their usages. Toward this end, the most straightforward method to defend against such attacks is to constrain the permission of sensor usage. The operating system providers could provide more strict invocation policies for built-in sensors. A simple solution is to adopt the same invocation policy with microphone, i.e., third-party APPs must request users for approval before invoking built-in sensors, and notify the specific usage for users as the contracts.

6.2.2 Sampling Rate Restriction. Built-in sensors in mobile devices are usually invoked for environmental sensing (e.g., gravity, temperature, humidity) and user-device interaction (e.g., motion tracking). To meet different requirements, the sampling rate of built-in sensors including accelerometer, gyroscope, magnetometer could be set in the range of $[0, 500]Hz$. Considering the battery life, the developers usually employ the sampling rate between 0Hz to 100Hz for normal uses (e.g., SENSOR_DELAY_NOMAL=200ms, SENSOR_DELAY_UI=60ms, SENSOR_DELAY_GAME=20ms[16]), which is much lower than their maximum sampling rate (e.g., 500Hz). Since *VoiceListener* relies heavily on sensor measurements with sufficient information, restricting the sampling rates of built-in sensors could help to defend the eavesdropping, while not affecting the daily uses. Such an approach could worsen the aliasing problem, hindering accurate voice recovery. Therefore, from the perspective of APPs, the sampling rate of built-in sensors could be restricted as a countermeasure.

6.2.3 Signal Jamming. In *VoiceListener*, pitch estimation is a prerequisite for harmonic reconstruction and aliasing correction, without which the amplitude spectrum could not be accurately recovered. To this end, signal jamming on the pitch of original voices played by the speaker could be employed to protect voices from leakage. Specifically, we adaptively generate a digital jamming signal as a narrow-band signal (e.g., 60Hz around the voices' pitch), which could mask the necessary pitch captured by the built-in sensors. On the other hand, the user's perception would not be interfered with significantly, because most harmonics still remain in the voices. This solution could be separately implemented as a plug-in for users to defend against such an attack.

7 CONCLUSION

In this paper, we demonstrate a universal eavesdropping attack, *VoiceListener*, which could recover the voices played by built-in speakers from undersampled sensor measurements by a model-driven algorithm, releasing the data collection efforts and adapting to various voices, platforms and domains. In particular, the most sensitive axis from 3-D built-in sensors is selected based on SNR as the raw signals for voice recovering. After that, we design an aliasing-corrected super resolution algorithm, including an aliasing-based pitch estimation and an aliasing-corrected voice recovering, to convert the narrow-band raw signals to wide-band voices. Experimental results show that the proposed *VoiceListener* could achieve acceptable performance in voice recovering using a magnetometer, accelerometer and gyroscope. And it could adapt to various voices, platforms and domains. Moving forward, considering the varying internal structures and sensing capability of different built-in sensors, we are interested in exploring the leaked voice information from distinct dimensions and granularities by fusing these

different sensor measurements, to further improve the recovered voice quality for supporting the prospective applications.

ACKNOWLEDGMENTS

This research is sponsored by National Key R&D Program of China (2020AAA0107700), National Natural Science Foundation of China (62102354, 62032021, 62172359, 61972348), Fundamental Research Funds for the Central Universities (2021FZZX001-27).

REFERENCES

[1] Amazon. 2021. Amazon Alexa - Learn what Alexa can do | Amazon.com. https://www.amazon.com/b?node=21576558011. (2021).
[2] S Abhishek Anand and Nitesh Saxena. 2018. Speechless: Analyzing the threat to speech privacy from smartphone motion sensors. In *Proceedings of IEEE S&P*. 1000–1017.
[3] Apple. 2021. Getting Raw Gyroscope Events. https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events. (2021).
[4] Apple. 2021. Siri - Apple. https://www.apple.com/siri/. (2021).
[5] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based practical smartphone eavesdropping with built-in accelerometer. In *Proceedings of NDSS*. 23–26.
[6] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2018. Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. *arXiv preprint arXiv:1807.03418* (2018). arXiv:1807.03418
[7] Fox Bussiness. 2019. Apple's Siri is eavesdropping on your conversations, putting users at risk: Report. https://www.foxbusiness.com/technology/apples-siri-is-eavesdropping-on-your-conversations-putting-users-at-risk. (2019).
[8] John Cunnison Catford. 1988. *A practical introduction to phonetics*. Clarendon Press Oxford.
[9] CowBoy Channel. 2021. Voice Assistant Industry Size, Market Share :2021 Market Research with Growth, Manufacturers, Segments and 2023 Forecasts Research. https://www.thecowboychannel.com/story/43600953/voice-assistant-industry-size-market-share-2021-market-research-with-growth-manufacturers-segments-and-2023-forecasts-research. (2021).
[10] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. In *Proceedings of IEEE S&P*. San Francisco, CA, USA, 694–711.
[11] Meng Chen, Li Lu, Zhongjie Ba, and Kui Ren. 2022. PhoneyTalker: An Out-of-the-Box Toolkit for Adversarial Example Attack on Speaker Recognition. In *Proceedings of IEEE INFOCOM*. London, United Kingdom, 1419–1428.
[12] ChinaDialy. 2018. Suit claims Baidu apps illegally tap data. http://www.chinadaily.com.cn/a/201801/06/WS5a5016cfa31008cf16da568a.html. (2018).
[13] Julien Epps and W Harvey Holmes. 1999. A new technique for wideband enhancement of coded narrowband speech. In *Proceedings of IEEE Workshop on Speech Coding Proceedings. Model, Coders, and Error Criteria*. 174–176.
[14] Ming Gao, Yajie Liu, Yike Chen, Yimin Li, Zhongjie Ba, Xian Xu, and Jinsong Han. 2022. InertiEAR: Automatic and Device-independent IMU-based Eavesdropping on Smartphones. In *Proceedings of IEEE INFOCOM*. 1129–1138.
[15] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n* 93 (1993), 27403.
[16] Google. 2021. Android Developer. https://developer.android.com/guide/topics/sensors/sensors_overview. (2021).
[17] Google. 2021. Google Assistant, your own personal Google. https://assistant.google.com/. (2021).
[18] Augustine Gray and John Markel. 1976. Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24, 5 (1976), 380–391.
[19] Daniel Griffin and Jae Lim. 1984. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32, 2 (1984), 236–243.
[20] Jun Han, Albert Jin Chung, and Patrick Tague. 2017. Pitchln: eavesdropping via intelligible speech reconstruction using non-acoustic sensor fusion. In *Proceedings of ACM/IEEE IPSN*. 181–192.
[21] Dik J Hermes. 1988. Measurement of pitch by subharmonic summation. *The Journal of the Acoustical Society of America* 83, 1 (1988), 257–264.
[22] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian J. McAuley, and Farinaz Koushanfar. 2021. WaveGuard: Understanding and Mitigating Audio Adversarial Examples. In *Proceedings of USENIX Security*. 2273–2290.
[23] Peter Jax and Peter Vary. 2003. Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model. In *Proceedings of IEEE ICASSP*, Vol. 1. I–I.
[24] Peter Jax and Peter Vary. 2003. On artificial bandwidth extension of telephone speech. *Signal Processing* 83, 8 (2003), 1707–1719.
[25] Volodymyr Kuleshov, S Zayd Enam, and Stefano Ermon. 2017. Audio super-resolution using neural nets. In *Proceedings of ICLR*.

[26] Guy Lemarquand, Romain Ravaud, Iman Shahosseini, Valérie Lemarquand, Jean Moulin, and Elie Lefeuvre. 2012. MEMS electrodynamic loudspeakers for mobile phones. *Applied Acoustics* 73, 4 (2012), 379–385.
[27] Xinyu Li, Venkata Chebiyyam, Katrin Kirchhoff, and AI Amazon. 2019. Speech Audio Super-Resolution for Speech Recognition.. In *Proceedings of ISCA INTERSPEECH*. 3416–3420.
[28] Zhuohang Li, Yi Wu, Jian Liu, Yingying Chen, and Bo Yuan. 2020. AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations. In *Proceedings of ACM CCS*. Virtual Event, USA, 1121–1134.
[29] Teck Yian Lim, Raymond A Yeh, Yijia Xu, Minh N Do, and Mark Hasegawa-Johnson. 2018. Time-frequency networks for audio super-resolution. In *Proceedings of IEEE ICASSP*. 646–650.
[30] John Makhoul and Michael Berouti. 1979. High-frequency regeneration in speech coding systems. In *Proceedings of IEEE ICASSP*, Vol. 4. 428–431.
[31] Michael I Mandel and Young Suk Cho. 2015. Audio super-resolution using concatenative resynthesis. In *Proceedings of IEEE WASPAA*. 1–5.
[32] Héctor A. Cordourier Maruri, Paulo Lopez-Meyer, Jonathan Huang, Willem Marco Beltman, Lama Nachman, and Hong Lu. 2018. V-Speech: Noise-Robust Speech Capturing Glasses Using Vibration Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4 (2018).
[33] MEMSIC. 2021. MMC3416xPJ. http://www.memsic.com/uploadfiles/2021/02/20210210110317113.pdf. (2021).
[34] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. In *Proceedings of USENIX Security*. 1053–1067.
[35] Microsoft. 2021. Cortana - Your personal productivity assistant. https://www.microsoft.com/en-us/cortana. (2021).
[36] D Murali Mohan, Dileep B Karpur, Manoj Narayan, and J Kishore. 2011. Artificial bandwidth extension of narrowband speech using Gaussian mixture model. In *Proceedings of IEEE International Conference on Communications and Signal Processing*. 410–412.
[37] Kun-Youl Park and Hyung Soon Kim. 2000. Narrowband to wideband conversion of speech using GMM based transformation. In *Proceedings of IEEE ICASSP*, Vol. 3. 1843–1846.
[38] Yasheng Qian and Peter Kabal. 2002. Wideband speech recovery from narrowband speech using classified codebook mapping. In *Proceedings of Australian International Conference on Speech Science, Technology*. 106–111.
[39] Nathanaël Carraz Rakotonirina. 2021. Self-Attention for Audio Super-Resolution. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*. 1–6.
[40] Sriram Sami, Yimin Dai, Sean Rui Xiang Tan, Nirupam Roy, and Jun Han. 2020. Spying with your robot vacuum cleaner: eavesdropping via lidar sensors. In *Proceedings ACM SenSys*. 354–367.
[41] Samsung. 2021. Samsung Bixby: Your Personal Voice Assistant | Samsung US. https://www.samsung.com/us/explore/bixby/. (2021).
[42] Weigao Su, Daibo Liu, Taiyuan Zhang, and Hongbo Jiang. 2022. Towards Device Independent Eavesdropping on Telephone Conversations with Built-in Accelerometer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4 (2022).
[43] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*. IEEE, 4214–4217.
[44] The New York Times. 2019. Amazon's Alexa Never Stops Listening to You. Should You Worry? https://www.nytimes.com/wirecutter/blog/amazons-alexa-never-stops-listening-to-you/. (2019).
[45] Heming Wang and Deliang Wang. 2020. Time-frequency loss for CNN based speech super-resolution. In *Proceedings of IEEE ICASSP*. 861–865.
[46] Tianshi Wang, Shuochao Yao, Shengzhong Liu, Jinyang Li, Dongxin Liu, Huajie Shao, Ruijie Wang, and Tarek Abdelzaher. 2021. Audio Keyword Reconstruction from On-Device Motion Sensor Signals via Neural Frequency Unfolding. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3 (2021).
[47] Teng Wei, Shu Wang, Anfu Zhou, and Xinyu Zhang. 2015. Acoustic eavesdropping through wireless vibrometry. In *Proceedings of ACM MobiCom*. 130–141.
[48] Sheng Yao and Cheung-Fat Chan. 2005. Block-based bandwidth extension of narrowband speech signal by using CDHMM. In *Proceedings of IEEE ICASSP*, Vol. 1. I–793.
[49] Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. 2015. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of ACM MobiSys*. 301–315.