Telesonar: Robocall Alarm System by Detecting Echo Channel and Breath Timing

Zhenyu Yan[†,*], Rui Tan[¶,◇,△], Qun Song[§,◇], Chris Xiaoxuan Lu[‡]
[†]The Chinese University of Hong Kong, Hong Kong SAR, China
[¶]Singtel Cognitive and Artificial Intelligence Lab for Enterprises, Nanyang Technological University, Singapore
[§]Energy Research Institute @ NTU, Interdisciplinary Graduate School, Nanyang Technological University, Singapore
[‡]University of Edinburgh, United Kingdom
zyyan@cuhk.edu.hk, {tanrui, song0167}@ntu.edu.sg, xiaoxuan.lu@ed.ac.uk

ABSTRACT

Massive fraudulent and phishing robocalls present threats to societies. The integration of artificial intelligence technologies, including dialogue and voice generation systems, renders the robocalls more deceptive. Existing countermeasures such as caller ID, call provenance, voiceprint, and fake voice detection have respective limitations and are heavyweight for end users' smartphones. This paper studies detecting the acoustic echo channel on the remote end of a call based on the received voice. The positive detection result evidencing the physical setup of an audio system is indicative of a human caller. However, the acoustic echo cancellation mechanisms of most audio systems and the use of earphone/headset diminish echoes significantly. To address these issues, the proposed *Telesonar* transmits short chirps during the vulnerable time of echo cancellation, detects the tiny echo remnants from the received voice, and passively analyzes the timing of caller's breath sounds to confirm a human caller. Extensive real experiments under a wide range of settings show that Telesonar correctly recognizes human callers with a rate of over 95%, while wrongly recognizing voice robots as human with a rate of 3.8%.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Applied computing** → Telecommunications.

KEYWORDS

mobile systems, robocall detection, internet-of-things systems

ACM Reference Format:
Zhenyu Yan[†,*], Rui Tan[¶,◇,△], Qun Song[§,◇], Chris Xiaoxuan Lu[‡]. 2022. Telesonar: Robocall Alarm System by Detecting Echo Channel and Breath Timing. In *ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3560905.3568500

*Part of this work was completed while Zhenyu Yan was with the Singtel Cognitive and Artificial Intelligence Lab for Enterprises, Nanyang Technological University, Singapore.
◇Also with School of Computer Science and Engineering, Nanyang Technological University, Singapore.
△Corresponding author.

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568500

1 INTRODUCTION

A survey in 2021 [37] shows that one third of all calls in the U.S. were considered nuisances or fraudulent and have caused a total loss of 29.8 billion US$ in one year. 3.4 million cases of these fraudulent calls are robocalls made with automated programs, which are 20% more than the year before [14]. Due to the low cost and low technical barriers, Internet-based *robocalls* have been used to pinpoint gullible victims in the first stage of fraud [13]. If the victim callee responds to the call (e.g., utters or presses some buttons following the robocall's instruction), the robocall program will forward the call to a human fraudster for the next stage of the scam. This low-cost victim filtering function prompts the massive robocalls. Since smartphones have become a primary personal communication method, in this paper, we aim to develop an edge sensing technique that can assist a smartphone user to confirm in real time that the remote end of a received call is not a voice robot. With such, the risk for the user to be defrauded will be reduced.

Various meta information about a phone call session and features extracted from the received voice have been used to block/detect robocalls. They include caller ID (CID), call provenance, voice recognition, and voiceprint. However, each of the existing approaches has certain limitations, which are discussed in the following.

CID: CID displays the incoming phone number. Telecommunication companies provide various services to block robocalls according to their CIDs. However, attackers can leverage various approaches to spoof their CID to be a legitimate one. Against the callees, robocalls can use neighborhood spoofing, e.g., display a CID of the callee's city.

Call provenance: Call provenance has been used to detect robocalls. The Pindrop system [5, 9] determines the traversal of a call through different networks by extracting and analyzing features from the received voice. The features are caused by the applied voice codecs, packet losses, and noise profiles of the traversed networks. A mismatch between the call source locations claimed by the caller and inferred by Pindrop should raise a suspicion of fraud. The provenance determination needs extensive prior knowledge about telephone networks around the world. Although call provenance services [4, 5, 9] have been commercially available, they are heavyweight due to the need of large databases and are used by large organizations only, such as bank call centers. As such, they are ill-suited for resource-constrained smartphones.

Voice content and voiceprint: For the robocalls playing recorded voices, it is possible to establish a voice database of such robocalls through crowdsourcing and use it to identify fraudulent

robocalls [6]. However, crowdsourcing incurs high communication overhead and privacy concerns. Moreover, should the identification be performed locally by the user's smartphone, it may consume excessive energy especially when the crowdsourced database is large. Besides content-based robocall identification, *voiceprint*, which has been widely used in customer authentication, is recently used for fraudulent call detection [4]. For example, calls with different CIDs carry the same voiceprint may imply fraud. Similar to call provenance determination, the voiceprint analysis is mostly for enterprise call centers, rather than smartphones.

The latest artificial intelligence (AI) technologies, including text-speech conversion and dialogue systems, make robocalls more deceptive. Generative adversarial networks (GANs) such as WaveNet [38] and GANSynth [17] can generate speech that mimics any human voice and sounds very natural to human perception. Deep model-powered dialogue systems (e.g., GPT-3 [10]) have developed capabilities to accomplish real-world conversation tasks. For instance, Google Duplex can call restaurants and hair salons to make reservations on behalf of the user [1]. There are reported cases [2, 34] in which the phishing calls have integrated advanced AI so that the callees do not realize that the caller is a robot. AI can render the voice content/voiceprint-based countermeasures less effective for the following reasons. First, as the advanced dialogue system can generate diverse dialogues, the voice content-based countermeasures are challenged. Second, as the GAN-based voice generator can mimic any person, voiceprinting becomes less effective. The misuse of GANs has triggered interests in applying machine learning to recognize fake voices [36]. But the existing solutions are susceptible to the underrepresentation of the training data, as we will show in §2. In summary, the evident uses of existing AI technologies in robocalls pose new and real threats to individuals especially the vulnerable users like elderly and kids.

To advance from the current state of lacking effective technical countermeasures on smartphones against the crafty threat of AI-powered robocalls, this paper aims at developing a technique that assists smartphone users in identifying the nature of the caller. From a key observation that the AI-powered robocallers only process and generate voices using software systems and do not involve a physical audio system of hardware speaker and microphone, we explore clues indicative of the presence of a physical audio system on the remote end of a voice call. In particular, we study the effectiveness of using the *acoustic echoes* as the indicator. Acoustic echo, which propagates from the speaker to the microphone via multiple paths including the phone's solid slate and the reverberation from the environment, is a ubiquitous issue for audio systems. Thus, most audio systems employ acoustic echo cancellation (AEC) mechanisms to improve audio quality, which however present a challenge to our aimed echo channel detection. The AEC renders the approach of passively detecting the echo channel futile. In addition, when the remote end uses wired earphones or a headset, the echo channel is slim or even absent.

In this paper, we design *Telesonar*[1], which uses echo channel detection and breath sound timing analysis in tandem to determine the nature of the remote end (i.e., human or voice robot). From our analysis, AEC is vulnerable in a time duration right after the establishment of a call session. Thus, Telesonar transmits three short acoustic chirps once a call session is established and then detects the echo remnants from the received voice data. We extensively evaluate the chirping parameters, including length, spectrographic shape, and silence gap, that can bypass AEC better. From our measurements, an effective chirping method uses three exponential chirps, each sweeping $[1, 3]$kHz in 0.5 seconds. If no echo channel is detected, Telesonar passively detects breath sounds in the received audio. If no breath sound is detected within a time window or the breath timing distribution is abnormal, Telesonar yields a negative result indicating a robocaller.

We evaluate Telesonar using both simulations driven by a large telecommunication dataset and experiments with real phone calls. Our evaluation covers a wide range of factors: (1) size of rooms creating reverberation from $2\,\mathrm{m}^2$ to $100\,\mathrm{m}^2$, (2) mouth-microphone distance from 2 cm to 2 m, (3) covering sounds (music, ringtone, pink noise) to reduce the prominence of chirping, (4) remote end's speaker volume (from minimum to maximum), (5) phone type/model (landline phone and four smartphones from high end to low end with different operating systems), (6) phone use mode (handset, speakerphone, wireless and wired earphone/headset), (7) geographic distance (domestic and cross-continental calls), (8) various call applications with different AEC algorithms (GSM, VoLTE, and VoIPs such as Google Hangouts, Skype, Whatsapp, Facebook Messenger), (9) different real-world robocalls, (10) human subjects' acceptance of being probed. The evaluation shows the effectiveness of Telesonar under diverse settings and also user acceptance.

The contributions of this paper are summarized as follows:

- We show the limitations of supervised learning approaches in detecting fake voices under practical settings. The results suggest that deeply tailored solutions beyond simply offloading the efforts to the training process will be needed to effectively address the problem.
- We design Telesonar that identifies the nature of the remote end (human or voice robot) by detecting echo channel and analyzing the caller's breath timing in tandem. The echo channel detection uses active sensing by transmitting several short chirps during the vulnerable time of AEC. The breath timing analysis can detect fake voices with artificial and natural breath sounds generated by an advanced approach [33].
- We achieve satisfactory performance under a wide range of settings. When the remote human caller uses handset, speakerphone mode, or true wireless stereo earphones (e.g., AirPods, Galaxy Buds), Telesonar correctly detects human caller with a rate of 95%, while wrongly recognizes voice robot as human with a rate of 3.8%. When the remote human caller uses wired earphones/headset, the two rates are 84% and 5%, respectively.

Paper organization: §2 states problem. §3 presents a measurement study. §4 and §5 present design and evaluation of Telesonar. §6 discusses adaptive attacks on Telesonar. §7 reviews related work. §8 concludes this paper.

2 PROBLEM DESCRIPTION

In this paper, we consider a call session established between a *caller* and a *callee* via a telecommunication link that may go through one or more networks such as public switched telephone network (PSTN) and Internet. The presentation of this paper is from the perspective of the callee. Thus, the callee's audio system is referred

[1]Telesonar uses telephone line and operates like a sonar to understand the target, either in the active or passive mode.

to as *near end*; the caller's audio system is referred to as *far end*. The aim of our work is to explore approaches to sense information indicative of a caller's nature (i.e., human or voice robot) based on the voice data received by the near end. Throughout this paper, we use the following terminology for the sensing results:

Positive is the detection result indicating a human caller;
Negative is the detection result indicating a voice robot;
False positive refers to wrongly detecting a voice robot as a human caller;
False negative refers to wrongly detecting a human caller as a voice robot.

In §2.1 of this section, we investigate the effectiveness of several recent fake voice detection approaches. Their limitations motivate us to study a new cyber-physical approach that is outlined in §2.2.The challenges in implementing the cyber-physical approach are discussed in §2.3.

2.1 Limitations of Fake Voice Detection

As GAN-based voice generation is the state of the art, we investigate the effectiveness of several latest approaches on the ASVspoof competition [7] in detecting spoofed voices. ASVspoof has two sectors: the *logical access (LA)* sector, in which the attacker can access the ASV system directly and add spoofed voices; and the *physical access (PA)* sector, in which the attacker can set up a speaker and replay the voice samples over the air. Most detection approaches of ASVspoof are based on supervised learning. We test an open-sourced detector [3] of the LA sector that ranked 15th among 50 detectors [36]. It employs an ensemble of three ResNets that use different speech representations as the input. We also test two baseline detectors of the PA sector [27, 35] provided by ASVspoof [36], which apply Gaussian mixture models (GMMs) on LFCC and CQCC features of the input speech. The three detectors have been trained by their authors using the ASVspoof database including both genuine and spoofed speeches. The equal error rate (i.e., the common value of false positive rate and false negative rate) of the three detectors on the ASVspoof database can be found in Table 1. Note that the false negative rate is the ratio that the genuine data samples from CallHome dataset are misdetected as spoofed. They achieve equal error rates of less than 10%.

Then, we use the CallHome American English Speech dataset [11] with 60 hours of telephone conversations to test the detectors. We prepare 240 genuine human voice samples, each having the same length as the ASVspoof test samples. As shown in Table 1, the false negative rates of the three detectors on CallHome are very high. Such underperformance is due to the pattern mismatch between the ASVspoof and CallHome databases. The genuine human speeches in ASVspoof are clean, whereas the CallHome voice traces are subject to background noises and telecommunication channel effects. While it is possible to engineer the training dataset to encompass patterns occurring during telecommunication calls, this approach can be exhaustive and tedious. This result is related to domain shift, a fundamental and pervasive challenge faced by supervised learning in real-world applications. Hence, it is desirable to design a robocall detector that is training-free and robust to domain changes, which can work as an alternative/complementary approach to the existing learning-based detectors.

Table 1: Equal error rates and false negative rates of three fake voice detectors on ASVspoof & CallHome datasets.

Spoofed voice detector	Equal error rate (%)		False negative rate (%)
	Validation	Test	CallHome
ResNets [3]	0.00	6.02	77.08
LFCCs + GMM [27]	2.71	13.54	100.00
CQCCs + GMM [35]	0.43	11.04	100.00

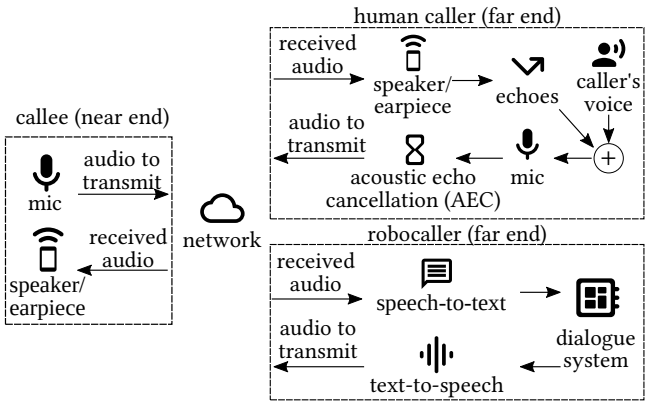


Fig. 1: Two scenarios of a call session. Upper right diagram shows the scenario of a human caller at the far end with an acoustic echo channel. Lower right diagram shows a robocaller without an acoustic echo channel.

2.2 System and Threat Models

The limitations of fake voice detection motivate us to study a cyber-physical approach of detecting the presence of far-end acoustic echo channel instantly on the establishment of the call session. The echo channel is an effective indicator of a human caller. We use Fig. 1 to illustrate the two scenarios with and without the presence of the far-end echo channel.

In the first scenario that captures human callers (upper right part of Fig. 1), the far-end speaker plays out the voice signal received from the near end. The played-out sound, after attenuation, arrives at the far-end microphone via multiple paths, including the direct propagation over the solid structure of the far-end device and the reverberations from the far end's indoor environment, forming acoustic echoes. Thus, the far-end microphone captures the mix of the caller's voice and the acoustic echoes. Then, the far-end AEC processes the signal, aiming at removing all echoes. The result is then transmitted to the near end via the telecommunication link.

In the second scenario (lower right part of Fig. 1) that captures typical AI-powered robocallers, the far end applies a speech-to-text algorithm on the voice signal received from the callee. Then, it uses a dialogue system to generate the response in text. Finally, it applies a text-to-speech algorithm such as WaveNet [38] to generate the voice signal and transmits it to the callee via the communication link. The key difference between the two scenarios is that, the far end in the second scenario does not have an acoustic echo channel.

In this paper, we make the following assumption for the robocaller, i.e., the operator of the robocaller does not set up physical speakers and microphones to create real acoustic echo channels. The reason is that, since a robocaller usually makes massive calls in parallel, setting up a separate physical audio system for each call incurs prohibitive overhead. Targeted fraud and phishing that

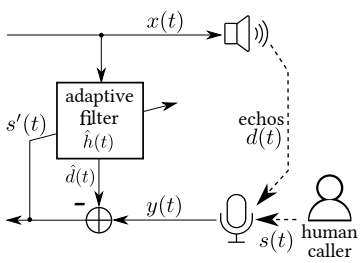


Fig. 2: Principle of AEC.

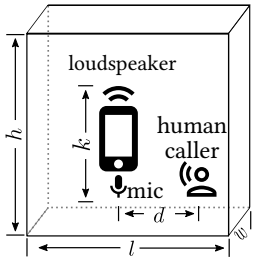
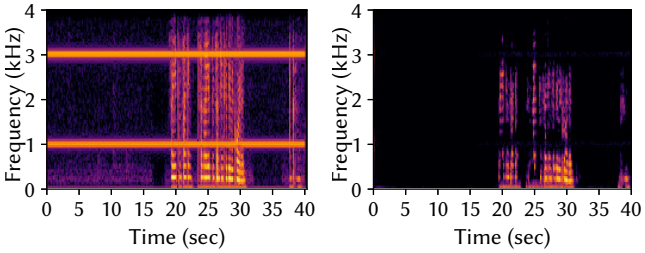


Fig. 3: A simulated case.



(a) Spectrogram of $y(t)$. **(b) Spectrogram of $s'(t)$.**

Fig. 4: Spectrograms of signal captured by microphone $y(t)$ and AEC's output $s'(t)$.

make a single or a limited number of calls at a time can use physical setups, but then the necessity of skillfully employing voice robots diminishes – the adversary should make the calls manually to simplify the procedure and maximize the success chance of targeted fraud. Dealing with fraud calls made by genuine humans is out of the scope of this paper.

We aim to detect the presence of the far-end echo channel based on the voice signals transmitted and received by the callee. The detection is performed at the near end only. The near end device needs to have the capability of inserting a probe signal to the transmitting audio, and listening to the incoming audio. Since we do not need to modify the telecommunication infrastructure, the near end device can be any device that can run our algorithm, such as a smartphone, a landline phone, a VoIP program, or even an automatic answering system (e.g., Google Assistant [39]).

2.3 Objectives and Challenges

This paper attempts to answer the following questions. First, how effective is the *passive sensing* approach of detecting the far-end echo channel, in which the near end does not introduce any probe signal irrelevant to the voice conversation? Second, how effective is the *active sensing* approach of detecting the far-end echo channel, in which the near end introduces some probe signals? On a related question, how to design the probe signal for the effective detection while not downgrading the voice quality much and not annoying the human caller?

In this section, we discuss three main challenges in implementing the far-end echo channel detection.

2.3.1 AEC. Fig. 2 illustrates the principle of AEC at the far end [20]. Denote by t the time and by $x(t)$ the voice signal received from the callee. Now, we analyze the essential acoustic process at the far end, with the distortions introduced by the far-end speaker and microphone hardware ignored. Denote by $h(t)$ the impulse response of the acoustic echo channel; by $d(t)$ the acoustic echo; by $s(t)$ the human caller's voice; by $y(t)$ the acoustic signal received by the far-end microphone. We have $d(t) = h(t) * x(t)$ and $y(t) = s(t) + d(t)$, where the operator $*$ represents convolution. AEC has two inputs: $x(t)$ and $y(t)$. Denote by $\hat{h}(t)$ the estimated impulse response of the echo channel. AEC predicts the echo by $\hat{d}(t) = \hat{h}(t) * x(t)$. Denote by $s'(t)$ the output of AEC, i.e., the estimated caller's voice after echo removal. AEC removes the echo by $s'(t) = y(t) - \hat{d}(t)$. If AEC obtains a perfect estimate of the echo channel (i.e., $\hat{h}(t) = h(t)$), the echo can be completely removed (i.e., $s'(t) = s(t)$).

Most AEC algorithms adopt *adaptive filters* that adjust the parameters of $\hat{h}(t)$ to make the feedback error $s'(t)$ zero. Now, we discuss the following four cases.

Case 1 (callee talk), in which callee keeps talking (i.e., $x(t) \neq 0$) and the caller keeps silent (i.e., $s(t) = 0$): The feedback error signal is $s'(t) = y(t) - \hat{h}(t) * x(t) = \left(h(t) - \hat{h}(t)\right) * x(t)$. Once the adaptive filter converges (i.e., $s'(t) = 0$), we have $\hat{h}(t) = h(t)$ since $x(t) \neq 0$.

Case 2 (caller talk), in which callee keeps silent (i.e., $x(t) = 0$) and the caller keeps talking (i.e., $s(t) \neq 0$): The error signal $s'(t) = y(t) - \hat{h}(t) * x(t) = s(t)$. Thus, there is no way for the adaptive filter to converge.

Case 3 (silence), in which both the callee and the caller keep silent (i.e., $x(t) = s(t) = 0$): The error signal $e(t) = y(t) - \hat{h}(t) * x(t) = 0$. Thus, there is no useful feedback error signal for the adaptive filter to learn $h(t)$.

Case 4 (double-talk), in which both the callee and the caller keep talking (i.e., $x(t) \neq 0$ and $s(t) \neq 0$): The error signal is $s'(t) = y(t) - \hat{h}(t) * x(t) = \left(h(t) - \hat{h}(t)\right) * x(t) + s(t)$. As the error signal contains two exogenous time-varying signals $x(t)$ and $s(t)$, it is difficult for the channel estimation to converge and make $s'(t) = 0$.

The above analysis gives the following key insight: the callee's voice $x(t)$ and the caller's voice $s(t)$ are the *constructive* and *destructive* factors, respectively, to the far end AEC's learning of the far-end echo channel. AEC implementations have integrated various heuristics to seize the Case 1 opportunities. For instance, the AEC can detect Case 1 based on the intensities of $x(t)$ and $y(t)$. When $x(t)$ is intense and $y(t)$ is weak, the AEC learns the initial $\hat{h}(t)$ or updates it to adapt to the changes of the echo channel.

AEC performs well in most cases. Thus, it presents a challenge to detecting the echoes from the output of AEC. To illustrate this, we conduct an experiment to mimic the far end. We set up an audio system, with its speaker playing a dual-tone $x(t)$ and its microphone capturing $y(t)$ that is the mix of a human voice and the echo of $x(t)$. Fig. 4a shows the spectrogram of $y(t)$, in which the human speaks from around the 20th to the 30th second. We can also see the dual-tone echo, with components at 1 kHz and 3 kHz. We feed the $x(t)$ and $y(t)$ to the AEC component of WebRTC, which is a widely adopted real-time communication library. Fig. 4b shows the spectrogram of the AEC output $s'(t)$. During the time period before the human speaks which belongs to Case 1, the $h(t)$ can be estimated well and the echo of $x(t)$ can be effectively removed. When the human speaks, the frequency components of the human speech at around 1 kHz and 3 kHz are suppressed by AEC. This over-suppression is due to that the estimate $\hat{h}(t)$ is not perfect, but it does not impede the human's perception of the voice content. From Fig. 4b, the echo of $x(t)$ has been almost completely removed.

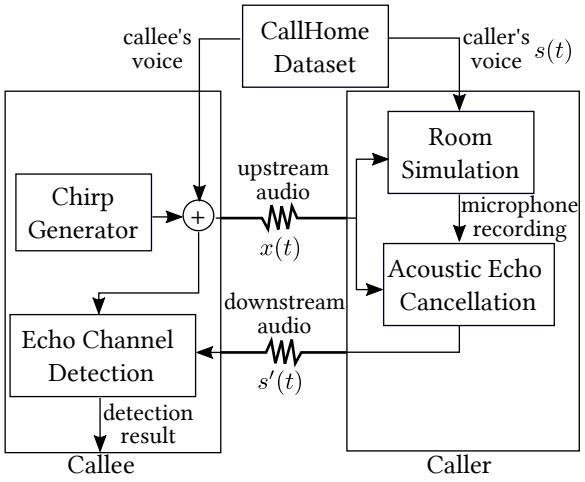
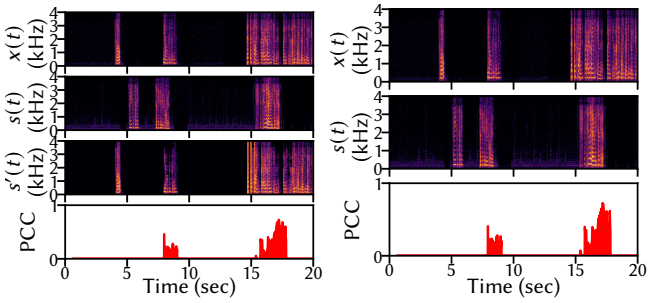


Fig. 5: `CallSim`: a simulator capturing echoing, AEC, and voice data exchanges between caller and callee.



(a) The far end has the echo channel (i.e., a human caller). **(b) The far end has no echo channel (i.e., a robocaller).**

Fig. 6: Spectrograms of voice signals during a call session under passive echo channel detection.

The above analysis and experiment suggest that, to effectively detect the far-end echo channel, Telesonar needs to exploit the vulnerable times of AEC. A short time right after the call session establishment is a promising vulnerable time, because the AEC has not experienced Case 1 to learn a good $\hat{h}(t)$. The changes of $h(t)$ in the midst of a call, e.g., caused by the movements of the far end, can also present vulnerable times. However, the changes of $h(t)$ are opportunistic.

2.3.2 Constraints from telecommunication systems. Limited channel bandwidth for each voice call session is a constraint from the telecommunication systems. The narrowband and wideband voice telecommunications restrict the bandwidth to be 3 kHz and 8 kHz [31], respectively. In the active sensing approach, the probes transmitted by the near end need to fit into the bandwidth. The probes will be audible. Thus, we should design the probes to minimize the negative impact on the hearing comfort of the human caller.

2.3.3 Slim echo channel of wired earphone. The acoustic echo channel can be slim or absent when the far end uses a wired earphone or headset with a separate microphone. Merely relying on echo channel detection may lead to excessive false negatives.

3 MEASUREMENT STUDY

This section presents a measurement study to obtain insights into addressing the challenges discussed in §2.3.

3.1 Measurement Setup and Methodology

Experimenting with real telephony systems faces various barriers (e.g., prohibited access to voice data) and overheads. Thus, we conduct our study and drive the Telesonar design using a simulator that integrates real acoustic signal processing algorithms. The designed Telesonar will be then evaluated in real telephony systems. Our simulator, `CallSim`, integrates four software modules as illustrated in Fig. 5 and can capture the two scenarios depicted in Fig. 1. The four modules are described as follows:

① **Room simulation.** This module simulates an enclosed room environment and computes the acoustic echoes in response to a given acoustic signal. We use the python library *pyroomacoustics* [29] to compute the reverberations. We simulate a $3 \times 3 \times 3\,\mathrm{m}^3$ space with two sources 4 cm apart, as illustrated in Fig. 3. The first source is the callee's voice played by the far end's loudspeaker; the second source is the caller's voice. The output of this module is the sum of the echo computed by *pyroomacoustics* and the caller's voice.

② **AEC.** We use the AEC of WebRTC, an open-source library for constructing real-time multimedia communication applications running on modern browsers and native clients.

③ **Echo channel detection.** This module has two inputs: (1) the callee's voice signal that is transmitted from the callee to the caller; (2) the output of AEC that is transmitted from the caller to the callee. It aims to detect the echo remnant from the output of AEC. The detection is based on the Pearson correlation coefficient (PCC) between the spectrograms of the two input voice signals. The details are presented in §4.2.

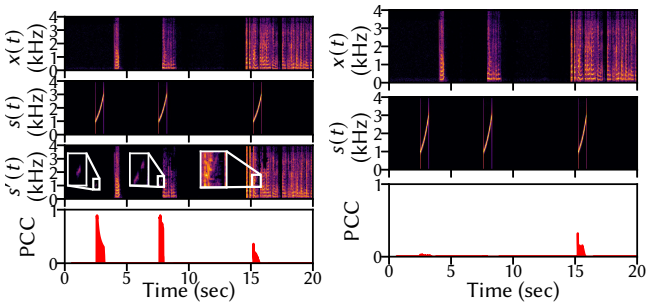
④ **Chirp generator.** This module is used in the active sensing approach to generate probe signals. The sum of callee's voice and the generated probe signal is transmitted to the caller.

We use the CallHome dataset [11] that includes 120 real domestic and cross-continental telephone conversations involving many people. Each conversation is 30 minutes long, consisting of two concurrent narrowband audio streams. One of the streams is the caller's voice; the other is the callee's voice. The voice traces capture realistic factors such as the microphone distortions and telecommunication noises/jitters.

We use a 30-minute conversation from CallHome to drive the measurement study. We compare the PCCs computed by the echo channel detection module under the passive sensing and active sensing schemes. For both sensing schemes, we investigate how the talk cases (caller talk, callee talk, and double-talk) affect the PCC. For active sensing, we investigate the impact of the probe signal configurations (i.e., choice between constant tones and chirp, choice between linear and exponential chirps, and the duration of the probe) on the PCC.

3.2 Measurement Results

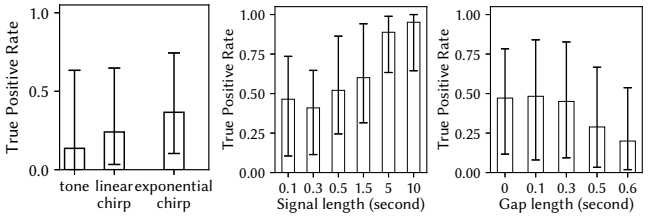
3.2.1 Passive sensing versus active sensing. Fig. 6 shows the results of passive sensing. Fig. 6a shows the spectrograms of the caller's voice $x(t)$, callee's voice $s(t)$, and the output of AEC $s'(t)$, as well as the PCC between $x(t)$ and $s'(t)$, when the far end has an echo channel. The AEC effectively cancels the echoes of callee's voice



(a) The far end has the echo channel (i.e., a human caller).

(b) The far end has no echo channel (i.e., a robocaller).

Fig. 7: Spectrograms of voice signals during a call session under active echo channel detection, in which the callee transmits three exponential chirps.



(a) Shapes of probe signals.

(b) Lengths of exponential chirps.

(c) Gap length between chirps.

Fig. 8: Impact of probe design on echo channel detection. Error bar represents min, mean, max over 120 traces.

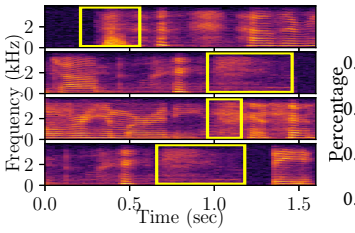


Fig. 9: Spectrograms of four cases of breath sounds (yellow frames) in real traces.

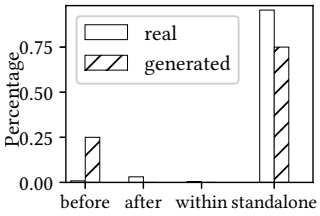


Fig. 10: Breath sound timing distribution in real and synthetic [33] voice traces.

and removes some components of the caller's voice when double-talk occurs. This is consistent with our observation in Fig. 4. Fig. 6b shows the results when the far end has no echo channel. The PCC traces shown in Fig. 6a and Fig. 6b are similar. Specifically, when double-talk occurs, the PCC can be up to 0.75; when there is no double-talk, the PCC is near-zero. The high PCCs during double-talks are primarily due to the similar harmonics of caller's and callee's voices. Since the double-talks result in similarly high PCCs in the absence and presence of the echo channel, passive sensing is not promising.

Fig. 7 shows the results under an active sensing approach, in which the near end uses an exponential chirp that increases from 1 kHz to 3 kHz to detect the far-end echo channel. Fig. 7a shows the results in the presence of far-end echo channel. We zoom in the areas of the spectrogram of $s'(t)$ that are expected to have the echo remnants. We can see that the AEC cannot completely remove the echoes. The corresponding PCCs shown in Fig. 7a at the beginnings of the first and second chirps are up to 0.9. The PCC for the third chirp is low, because although $s'(t)$ contains the echo remnant, it is mainly the caller's voice $s(t)$. Therefore, although AEC may not perform well during double-talks, it is non-trivial to detect the AEC output's inclusion of the echo remnants. From the results in Fig. 7a, the caller's silence period is a good timing for transmitting the probe signal, because we can leverage the convergence process of AEC to create salient echo remnants in the AEC outputs. Fig. 7b shows the results in the absence of far-end echo channel. The PCCs between the exponential chirp and the caller's voice are always low.

3.2.2 Design of probe signals. §3.2.1 shows that active sensing is promising. Now, we investigate the impact of the shape and length of the probe signal, as well as the gap between two consecutive probe signals on the effectiveness of the echo channel detection. We adopt a threshold of 0.1 for the PCC to detect the far-end echo channel. We use the true positive rate to characterize the effectiveness of the probe signal design. The true positive rate is measured as follows. In an experiment where the far end has the echo channel, the callee transmits the probe signal periodically over the entire 30-minute conversation. The ratio of the positive detection decisions to the total number of detection decisions made is the true positive rate. Fig. 8a shows the true positive rates when a constant tone, a linear chirp, and an exponential chirp are adopted. The linear chirp and the exponential chirp sweep the frequency range of $[1\,\text{kHz}, 3\,\text{kHz}]$. The exponential chirp gives the best detection performance.

Fig. 8b shows the impact of exponential chirp length. In general, the detection performance increases with the chirp length. However, long chirps (e.g., 2 seconds, 5 seconds or 10 seconds) are disturbing. Telesonar adopts 0.5 seconds, because longer chirps do not bring much improvement, i.e., proportional improvements to that from 0.3 to 0.5.

Fig. 8c shows the impact of the gap between two chirps. When the gap is larger than 0.3 seconds, the performance becomes worse. This is because with larger gaps, the chirps are more sporadic and will be removed by AEC due to its built-in heuristics. Telesonar adopts 0.1 seconds by default.

3.2.3 Breath sound timing. Breath sounds are pervasive in telecommunication audios and may indicate human callers. Thus, when the echo channel is slim or absent, we aim to exploit breath sounds to passively determine the nature of the far end. A recent research [33] proposes a state-of-the-art deep learning-based technique that synthesizes voices with spontaneous breath sounds, which are perceived as natural ones by human. This technique may be used by the robocaller to improve the deceptiveness of the robot voice. Thus, instead of only examining the existence of breath sounds in the far-end voice, we also examine the timing of breath sounds. Specifically, we consider four cases of breath timing: a) at the beginning of an utterance; b) at the end of an utterance; c) within an utterance; d) standalone (i.e., no concurrence with an utterance). Fig. 9 shows the spectrograms of the four cases of breath sounds detected by an existing algorithm [16]. Fig. 10 shows the distributions of the four cases, for the real CallHome traces and the synthetic traces made publicly available by the work [33]. The results show that real breath sounds appear mostly standalone or at the end of utterances

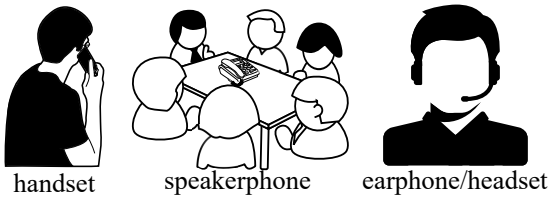


Fig. 11: Three use modes of human caller's far-end device.

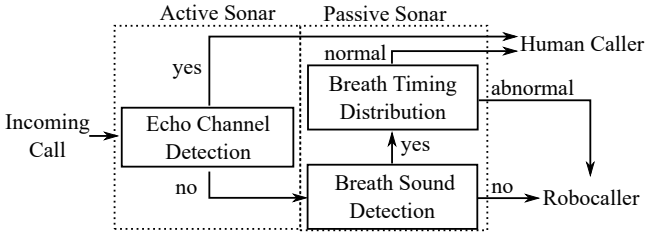


Fig. 12: Workflow of Telesonar.

sometimes. But the breath sounds from synthetic voice traces are more likely to appear at the beginning of utterances. The results show that it is possible to detect the robocaller according to breath sound's timing even if breath sound is synthesized and added to the generated voices.

4 DESIGN OF TELESONAR

4.1 Overview of Design

We consider three use modes of the far-end device as illustrated in Fig. 11: (1) *handset* mode, in which the phone plays the received voice using its top speaker; (2) *speakerphone* mode, in which the far-end device (e.g., a smartphone, a tablet, and a laptop) is held in hand or placed on a table and plays the received audio using its loudspeaker; (3) *earphone/headset* mode, in which the human caller uses an earphone or headset connected to the far-end device to make the call.

In the handset mode, although the volume of the top speaker is low, the solid slate of the phone forms an effective acoustic echo channel. In the speakerphone mode, as the speaker's volume is high, the direct propagation of the played-out acoustic signal to the microphone and the reverberation from the surrounding objects/walls form an effective echo channel. Wireless earphones such as Apple AirPods, Bose QuietComfort 35, and Samsung Galaxy Buds, have direct contact with the head skull, and thus can receive the chirps as well through bone conduction. From the results in §3, the active sensing approach is promising to detect the presence of a far-end echo channel in the handset and speakerphone modes. For rare cases where the far end uses wired earphones/headsets, breath timing is a promising feature to confirm that the far end is a human caller because the microphone is generally placed close to the mouth.

From the above discussion, the echo channel detection and the breath sound analysis are complementary for covering the three use modes. Telesonar integrates these two detection methods by the workflow shown in Fig. 12. Telesonar runs on the user's near-end device. Upon the incoming call session is established, Telesonar transmits a chirp signal to the far end and detects the echo channel. Note that there is normally a silent period at the beginning of a call session due to the human's delay in responding to the ringing tone termination indicating call establishment. Telesonar exploit this silent period as a vulnerable time of AEC. If the far-end echo channel is detected, Telesonar prompts a message to the user confirming that the far end is a human caller. Ideally, this message is prompted once the user touches a button on the smartphone to accept the incoming call, because the active sensing-based echo channel detection takes about one second only. We will present the details of the echo channel detection in §4.2.

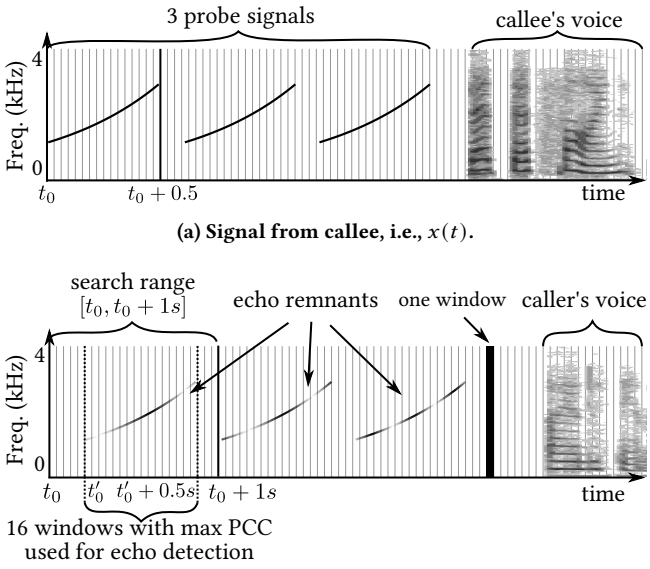
If no echo channel is detected, Telesonar detects and analyzes breath sounds. Telesonar uses a signal processing algorithm proposed in [16] to separate human voices and breath sounds based on the acoustic features related to the vocal tract system. If a breath sound is detected, Telesonar proceeds to accumulate more breath sounds to derive their timing distribution. If the distance between the derived distribution and a reference distribution is smaller than a threshold, Telesonar decides that the far end is a human caller; otherwise, Telesonar alerts that the far end is likely to be a robot. If Telesonar cannot detect a breath sound within 15 seconds, Telesonar also asserts robot. The parameter settings presented above balance the overall trade-off between true and false positive rates from extensive empirical trials.

The echo channel detection is compute-lightweight (0.1 seconds compute time on modern smartphones) and instant (1.7 seconds detection delay), whereas the breath sound detection/analysis needs more computation and time (12 seconds detection delay on average as shown in §5). The workflow in Fig. 12 avoids breath sound detection/analysis when the far-end human caller is in the handset or speakerphone mode. In these cases, Telesonar gives the decision instantly.

4.2 Echo Channel Detection

This section presents the design of the echo channel detection. Each *detection session* begins with a synchronization process, in which the callee sends an exponential chirp to the caller. Then, the callee transmits a sequence of n intermittent exponential chirps. Each chirp lasts for 0.5 seconds and sweeps the frequency band of $[1\,\text{kHz}, 3\,\text{kHz}]$. This frequency range is similar to that of phone's keypad tones from $0.697\,\text{kHz}$ to $1.633\,\text{kHz}$. The gap length between two exponential chirps is 0.1 seconds. The *synchronization* of the two ends and the *detection* of a single chirp's echo from the voice signal received from the far end are discussed below.

Synchronization. Both the transmission of voice signal over the telecommunication link and the AEC computation at the far end introduce delays. As advised by International Telecommunication Union (ITU), the AEC computation can take up to 20 ms. Thus, we need to synchronize the data traces of the transmitted chirp and the chirp's echo (if any) in the signal received from the far end. Fig. 13 illustrates the synchronization process. The near end splits the 0.5-second chirp signal transmitted during $[t_0, t_0 + 500\,\text{ms}]$ into 16 neighboring windows, each lasting for $31.25\,\text{ms}$, and applies the short-time fast Fourier transform to every window to generate the spectrogram consisting of 16 frequency spectra over time. The near end applies the same approach to generate the spectrogram for the signal received from the far end during $[t_0, t_0 + 1000\,\text{ms}]$, since the total delay including the round-trip communication delay



(a) Signal from callee, i.e., $x(t)$.

(b) Signal from caller, i.e., $s'(t)$.

Fig. 13: Illustration of signals from callee/caller and the synchronization process prior to echo detection.

(shall less than 800 ms as per [21]) and local processing delay (normally below 40 ms [32]). This second spectrogram consists of 32 frequency spectra over time. Then, the first spectrogram is slid over the second spectrogram with a step size of one frequency spectrum. During the sliding, a PCC between the first spectrogram and the covered part of the second spectrogram is computed for each step. If the maximum PCC is obtained at the ith step, the first window of the transmitted chirp signal corresponds to the ith window of the signal received from the far end, completing the synchronization. We use t_0' to represent the start time of the ith window of the signal received from the far end.

Detection. After the synchronization, the near end computes the PCC between the frequency spectra of every two corresponding windows respectively from the transmitted chirp signal and the signal received from the far end. If the average PCC over the 16 windows is larger than a predefined threshold, the detector produces a positive detection result for the chirp; otherwise, the detector produces a negative result. The evaluation performed in this work will extensively evaluate the impact of the threshold on the detection performance.

The near end applies the above two steps for each transmitted chirp in a detection session. After that, the near end applies an *OR decision fusion* to aggregate the detection decisions in the detection session. Specifically, if any one or more detection decisions are positive, the final detection decision is positive; otherwise, the final detection decision is negative. This decision fusion effectively deals with the cases in which the echo is partially or completely removed by the far end's AEC. Evaluation in §5.2 will recommend a good setting for the number of fused decisions (i.e., n).

4.3 Breath Sound Verification

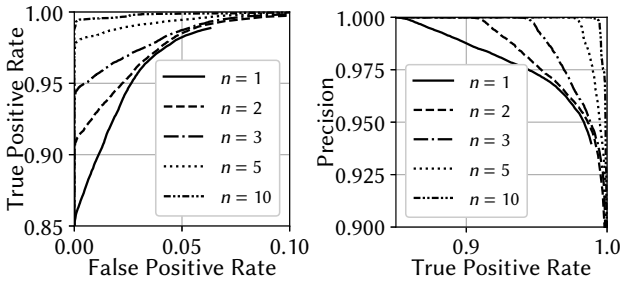
Echo channel detection can verify the human caller within a short time in most of cases, i.e., handset mode, speakerphone mode, and wireless earphones. However, in infrequent cases where the far-end caller uses wired earphones, the echo channel is slim. Based on our finding of the varied distributions of breath sound timing of human's and synthetic speeches in §3.2.3, Telesonar detects caller's breath sounds and compares the distribution of breath sounds against the template distribution to verify the human caller. First, we generate a template distribution of breath timing using the CallHome dataset, containing 60 hours of telephone conversations from 240 people. During the phone call, when the callee's device cannot detect the existence of the caller's echo channel and breath sounds are caught in the audio, the breath timing is recorded and classified based on its location to an utterance, i.e., at the beginning, in the middle, at the end, or standalone. Then, we compute the distance between the breath timing from the received audio and the template distribution. Should the distance be smaller than a threshold, a positive result is generated.

5 PERFORMANCE EVALUATION

5.1 Evaluation Methodology

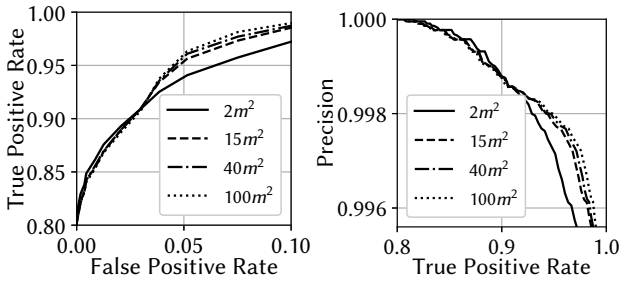
We evaluate Telesonar under two settings, i.e., in `CallSim` and via domestic/cross-continental voice calls using real smartphones. We use `CallSim` to evaluate the impact of various configurable and situational parameters on the detection performance. The parameters include decision fusion setting n, far-end room size, the distance between the human speaker and the far-end microphone. We also evaluate how the chirping affects audio quality. With `CallSim`, we can easily run many experiments under a wide range of settings to generate insightful results. Moreover, we deploy Telesonar to a real smartphone as the near end and use numerous phones (including smartphone and landline PSTN phone) running various voice applications as the far end. Real voice applications usually contain audio processing algorithms and heuristics that `CallSim` does not include. For instance, the solid slate of smartphones may present an echo channel different from the one simulated in `CallSim`. The voice applications generally use the device-specific AECs that are optimized by the device manufacturers and provided by the customized operating systems. Our experiments involve different smartphone models, different use modes (handset, speakerphone, earphone), and different voice call applications.

We use receiver operating characteristic (ROC) and precision-TPR curves as performance metrics. ROC shows true positive rate (TPR) versus false positive rate (FPR). The "positive" here means a detection decision of human caller. The two rates are defined as follows. Let N_x denote the number of instances for a detection result type x, where x can be *true positive* (TP), *false negative* (FN), and *false positive* (FP). The two rates are $\text{TPR} = N_{TP}/(N_{TP} + N_{FN})$ and $\text{FPR} = N_{FP}/(N_{FP} + N_{TN})$. A ROC curve is generated by varying the detection threshold, which depicts the sensitivities of Telesonar to both ground-truth cases. We also use precision-TPR curves to characterizes how much we can trust Telesonar's positive detection results. The precision is computed by $N_{TP}/(N_{TP} + N_{FP})$, and the recall rate is the same as TPR. In each experiment, Telesonar continuously emits 0.5s-chirps separated by 0.1s gaps throughout the call session and performs echo channel detection for every group of three chirps. This evaluation methodology is different from our proposed design of only emitting three chirps at the beginning of a call (cf. §4), but it allows us to have enough detection results to calculate TPR and FPR. Since AEC performs worst at the beginning of the call, the performance of Telesonar that only



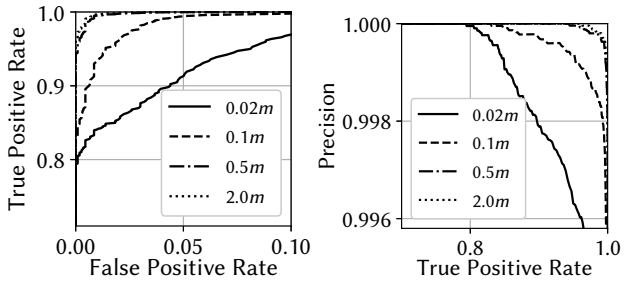
(a) ROC. (b) Precision-TPR.

Fig. 14: Impact of decision fusion.



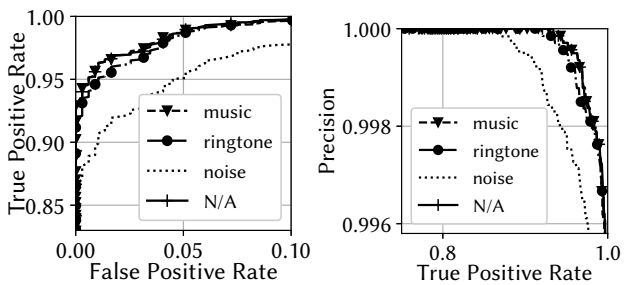
(a) ROC. (b) Precision-TPR.

Fig. 15: Impact of room size.



(a) ROC. (b) Precision-TPR.

Fig. 16: Impact of speaker-phone distance.



(a) ROC. (b) Precision-TPR.

Fig. 17: Impact of covering sounds.

performs detection at the call start will be better than that measured in our evaluation.

5.2 Evaluation with `CallSim`

We use all CallHome data traces to drive the simulations. In the simulations, we disable the breath timing analysis in the workflow shown in Fig. 12, i.e., Telesonar detects robocalls based on absences of echo channel and breath sounds. The breath timing analysis is enabled in the experiments with real phones presented in §5.3.

Impact of decision fusion. We use decision fusion discussed in §4.2 to improve Telesonar's sensitivity. With a larger n setting, Telesonar transmits more chirps and requires a longer time to generate a fused decision, which negatively affects detection timeliness. For each CallHome trace, we vary n from 1 to 10. No decision fusion is applied when $n = 1$. Fig. 14 shows the ROC and precision-TPR curves of different n. Fig. 14a shows that, for the same FPR, TPR increases with n. Fig. 14b shows that precision increases with n for the same TPR, When n increases from 1 to 3, there are substantial ROC/precision improvements. With $n = 3$, the total chirping time is 1.7 seconds. Higher n settings, though leading to better detection performance, cause more disturbances. Thus, we adopt $n = 3$.

Impact of room size. The far end's room enclosure may affect the echo channel. We vary the room size from $2\,\mathrm{m}^2$ (like a washroom) to $100\,\mathrm{m}^2$ (like a hall). The room height is 2.6 m. For each room size, we run 20 experiments. In each experiment, the locations of the human speaker and the microphone at the far end are randomly generated. The ROC and precision-TPR curves in Fig. 15 show that the room size has little impact on the echo channel detection performance. This is because the echo propagating via the direct path from the far-end speaker to the microphone is the main echo component. It implies that the echo channel detection performance highly depends on the effectiveness of the direct echo path (e.g., the solid slate of a smartphone), whereas the reverberation from the ambient plays a less significant role. When the far end is in a smaller room, the detection performance is slightly worse. This is because, in a small room, the caller's voice can be better captured by the microphone, drowning out the echoes.

Impact of distance between human speaker and far-end microphone. In different calling scenarios at the far end, the considered distance varies. The voice volume captured by the far-end microphone, which depends on the distance, may affect the far-end AEC and the echo channel detection at the near end. We vary this distance, which is denoted by d in Fig. 3, from 2 cm to 2 m. The ROC and precision-TPR curves in Fig. 16 show that Telesonar performs better when the d is larger. This is because when the human speaker is closer to the microphone, the captured voice volume is higher, drowning out the probe.

Impact of covering sounds. Telesonar can make the probe signal less noticeable by mixing it with other sounds that the caller may be familiar and comfortable with. We evaluate the impacts of three covering sounds on the echo channel detection performance, i.e., a music clip, a ringtone, and pink noise. Fig. 17 show the resulting ROC and precision-TPR curves. The curve labeled "N/A" is the result using the bare probe signal. We can see that music and ringtone introduce little impact on the echo channel detection performance. However, pink noise degrades the performance. This is because pink noise has a wide frequency range that fully covers the chirp's frequencies, causing AEC to suppress the chirp. The above results suggest that we can embed the probe signal into music or ringtones for a less weird probing process. Note that §5.4 will

report the results of a user study regarding the impact of the probe signal on the human caller's comfort.

Breath sound detection. We evaluate the breath sound detector presented in §4.1 using 120 CallHome genuine human voices and 120 synthetic audio samples from the Fake-or-Real dataset [25]. It achieves 79.17% TPR and 10.83% FPR. It takes 12 seconds on average and up to 20 seconds to yield the first detection result.

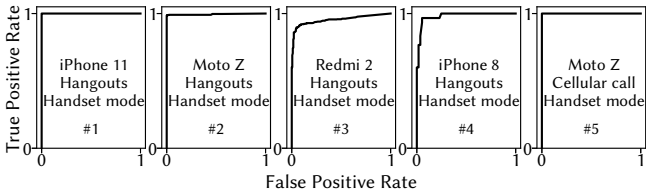
Overall detection performance. On the 120 CallHome traces and 120 synthetic samples, the concatenated echo-breath detection achieves 98% TPR when FPR is 10%. Note that `CallSim` simulates the loudspeaker mode.

5.3 Experiments with Real Phones

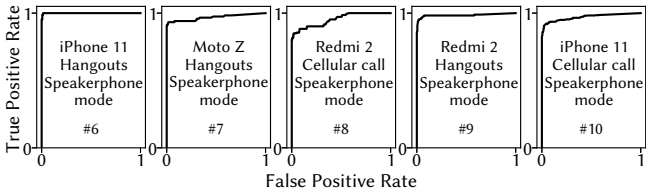
We conduct a set of experiments using real smartphones as near and far ends. Upon the call establishment, the near end transmits three chirps. To evaluate Telesonar's effectiveness on various caller devices, we use four smartphones (iPhone 11 Pro Max, iPhone 8, Moto Z, and Redmi 2) and a landline PSTN phone as the far end. The first three smartphones are high-end models that may have optimized audio systems. We make the calls through both VoIP applications (Google Hangouts, Skype, Whatsapp, and Facebook Messenger), cellular networks, and PSTN. For smartphones, we experiment in all the handset, speakerphone, and earphone modes. Several experiments are conducted with the caller and callee located in two cities more than 10,000km apart.

To improve consistency and comparability of experiments, we use a voice played by the loudspeaker of a smartphone to mimic the human speaker's voice. Thus, this smartphone is referred to as the *pseudo human speaker*. The distance between the tested phone and the pseudo human speaker is 10 cm. Note that we adopt a pseudo human speaker in most tests to make the speech content same for the human caller and non-human caller cases, which can evaluate Telesonar's performance under the exact same speech. In addition, the pseudo human speaker can ensure fair comparisons with various impacting factors. We also evaluate Telesonar with a real human caller at the end of this subsection. At the near end, to superimpose chirps and analyze the received data, we need to access the voice transmitting and receiving processes at the same time. To achieve this, we connect the callee's smartphone to a desktop computer via Bluetooth. The computer runs Ubuntu 18.04.3 with BlueZ 5.48 and PulseAudio 11.1 as the Bluetooth and audio drivers. We configure the smartphone to use Bluetooth HeadSet Audio Gateway (HSP/HFP) profile in PulseAudio's volume control panel so that the smartphone acts as a relay for the computer. The computer runs the echo channel detector. This setup allows us to experiment with the phone's built-in voice call function and off-the-shelf VoIP applications that are installed on both the near-end and far-end smartphones.

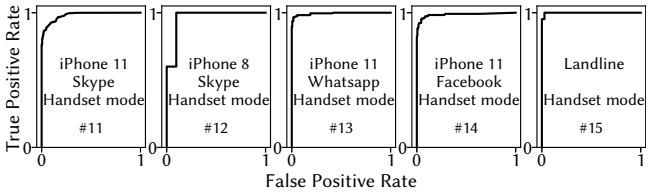
The ground truth in the above setup is human caller. To have the ground truth of robot, we make phone calls to several local service lines with automatic answering, which include a telephone company's customer service, a postal company's enquiries line, and a bank's hotline. The calls are made with the following two near-end devices: an Apple iPhone 11 Pro Max using telephone company A's GSM network, and a Google Pixel 4 using telephone company B's VoLTE network. We connect a laptop to the phones as a Bluetooth headset for transmitting the chirps and recording the incoming voice. Our setup acts as the near end running Telesonar. A positive detection result of echo channel is a false positive.



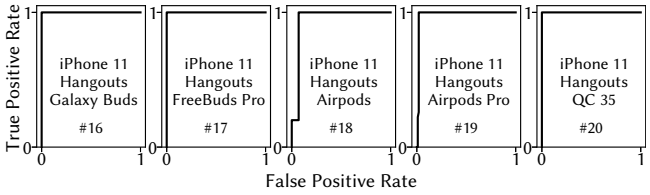
(a) The far-end phone in handset mode.



(b) The far-end phone in speakerphone mode.



(c) Calls over various VoIP apps and a landline phone.



(d) The far-end phone uses Bluetooth earphones.

Fig. 18: Echo channel detection performance with real phones over telecommunication systems.

Fig. 18 shows ROCs from 15 experiments under various settings. The results are explained as follows.

Handset mode. The TPR is obtained from the case where the far-end phone is used in the handset mode; the FPR is obtained from the calls to the local service lines. Experiments #1 to #4 in Fig. 18a are conducted using Google Hangouts. Consistent with our results in Fig. 7b, Telesonar can detect the echo remnants in the voice received by the callee. Among the four experiments, when the requested FPR is below 5%, the TPR is above 95%. Experiment #5 in Fig. 18a makes a phone call through the cellular network where the far-end phone is used in the handset mode. Because the far-end AEC is yet converged when the call is just established, the beginning part of the callee's received audio has shown clear chirp echoes. Hence, Telesonar achieves 100% TPR.

Speakerphone mode. The TPR is obtained from the case where the far-end phone is used as a speakerphone; the FPR is obtained from the calls to the local service lines. Experiments #6 to #10 in Fig. 18b show that Telesonar works well on both Google Hangouts and cellular calls. TPR is above 93% when FPR is below 5%. Some parts of the far-end audio are erased. Thus, Telesonar has slightly lower TPR in the speakerphone mode, compared with the handset mode. This is because that the loudspeaker is much louder than

the top speaker and the far-end device uses heuristics to deal with double-talk for better audio quality.

Performance of Telesonar on other VoIP applications. We evaluate the cases where the far-end phone runs Skype, Whatsapp, and Facebook Messenger. We conduct four experiments, i.e., #11 to #14. The results are shown in Fig. 18c. The ROCs show that Telesonar performs satisfactorily on all these VoIP applications. When FPR is below 5%, the average TPR is 86%. Telesonar has slight performance degradations on these VoIP applications compared with Hangouts. This may be caused by their more effective proprietary AEC, compared with that of the open-sourced WebRTC used by Hangouts.

Wireless earphones/headsets. We evaluate Telesonar when the far end is connected to popular wireless earphones or headsets. In particular, we test four earphones (i.e., Samsung Galaxy Buds Pro, Huawei Freebuds Pro, Apple AirPods 2nd generation, Apple AirPods Pro) and one headset (i.e., Bose QuietComfort 35). The ROC curves in #16 to #20 show that Telesonar has perfect performance when an iPhone 11 is connected with five popular Bluetooth earphones/headsets. We note that Telesonar achieves slightly lower accuracy in #18 on Apple Airpods (2nd generation) because the longer distance between the microphone and the speaker/head skull results in reduced signal strengths of the received echoes. Nevertheless, the results confirm that wireless earphones/headsets can receive the probe signal through bone conduction because of its direct contact with the head skull.

Landline PSTN. We evaluate Telesonar when the far end is a landline PSTN telephone. Experiment #15 in Fig. 18c shows that when FPR is low than 5%, TPR is at least 100%.

Geographic distance. The experiments #4 and #12 are conducted with the callee and caller located in two cities more than 10,000km apart, while other experiments are conducted with the callee and caller located in the same city. When FPR is below 5%, The TPR involving the cross-continental calls is 8% lower than the average of the domestic calls. The results show that the long-distance calls result in slightly lower ROCs. This is because of the increased noise and distortion levels of the voices due to the cross-continental telecommunications.

In summary, from the 15 experiments presented in Fig. 18, the echo detection of Telesonar achieves at least 86%, 93%, 96% TPR when FPR is 1%, 5%, 10%, respectively. The full workflow in Fig. 12 achieves 90%, 95%, 96% TPR when FPR is 0.5%, 3.8%, 6.9%, respectively. In all experiments, the precision is above 99.77%. Due to space constraints, the details regarding precision are omitted.

Wired earphones/headsets. We make calls over cellular networks to a far-end device of Apple iPhone 11 Pro Max with wired earbuds and headsets. The breath detector yields TPR of 84% and FAR of 5%.

Loudness of the probe chirp. We transmit the probe signal over cellular networks and measure its loudness on the far end by following ITU-R BS.1770-4 standard [24]. The chirp loudness at the far end is −23.7 dB LUFS, which is closed to recorded phone conversations from CallHome (−22.7 dB LUFS on average). This is because most telephone systems have adaptive volume control.

Far end speaker volume setting. We set the speaker volume of the iPhone 11 Pro Max as the far end to the maximum and minimum (but not muted) under both the handset and speakerphone modes. The results in Fig. 19 show that the far-end speaker volume has little impact on Telesonar's performance. In particular, when FPR is 5%, the minimum volume setting causes 4.55% and 0% TPR drop in

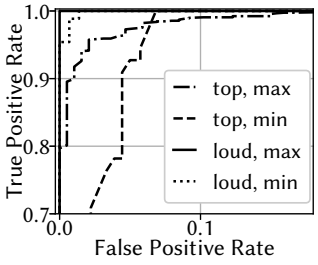


Fig. 19: Impact of far-end speaker volume.

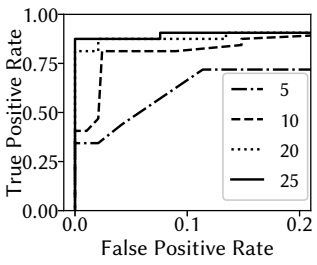
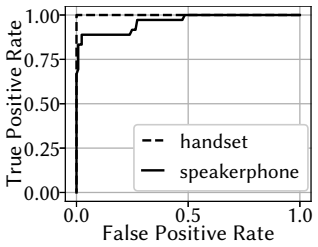
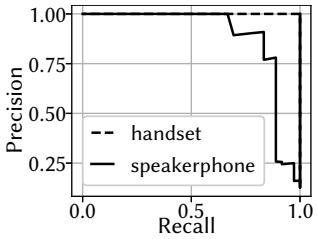


Fig. 20: Impact of the number of breath sounds.



(a) ROC.



(b) Precision-TPR.

Fig. 21: An end-to-end experiment with a human caller.

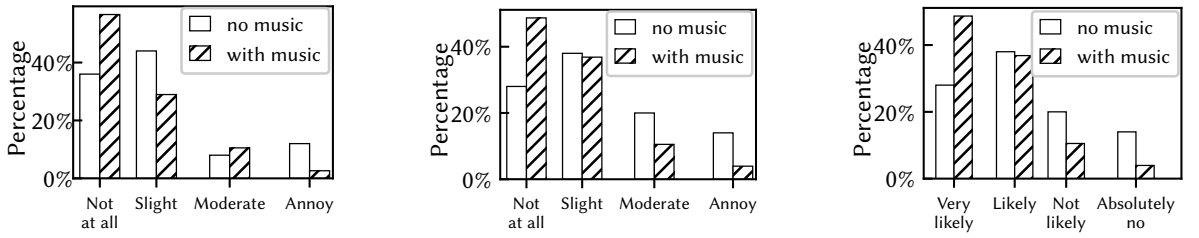
the handset and speakerphone modes, respectively. This is because, even with the lowest volume setting, the direct propagation path over the phone's solid structure is still effective.

Counteracting attack with generated breath sounds. As the source code of the voice synthesis approach with spontaneous breath sounds [33] is not publicly available, we are unable to build an advanced voice robot with spontaneous breaths. Thus, we separately evaluate FPR in `CallSim` driven by the synthetic traces made publicly available by [33] and TPR with real phones. Fig. 20 shows the ROCs when Telesonar collects different numbers of breath sounds for breath timing analysis. When FPR is below 10%, the TPRs are 44%, 81%, 88%, and 91% when the breath sound count is 5, 10, 20, and 25, respectively. As the median of an adult's breath interval is 3.75 seconds [18], Telesonar can achieve acceptable accuracy (10% FRP, 81% TPR) via 30 seconds passive sensing against such a crafted robocaller.

Real human caller. Previous experiments use loudspeakers as pseudo human speakers for good reproducibility and fair comparison regarding various affecting factors. We conduct an end-to-end experiment with a human caller using an iPhone 11 via Google Hangouts in either handset or speakerphone mode. The callee plays the chirps at the beginning of the call. Figs. 21a and 21b show the ROC and precision-TPR curves of the experiments, in which Telesonar performs better when the callers are in the handset mode. This aligns with the observations in the tests with non-human callers. Both experiments show that Telesonar achieves satisfactory results with human callers and real phones under practical settings.

5.4 Caller Acceptance Study

As the human caller is subject to Telesonar's probing, we conducted an acceptance study. (IRB approval has been obtained. Details are omitted for anonymity.) We recruited 50 and 76 adult volunteers in two rounds. The study consists of the following steps. (1) We



(a) Speaker mode. **(b) Earphone mode.** **(c) Acceptance of Telesonar.**

Fig. 22: User study of Telesonar.

introduce the background of Telesonar. Specifically, we play two recordings of AI agents making phone reservations with a hair salon and a restaurant [8] and state that Telesonar is designed to detect robocalls that may apply similar AI agents for massive frauds. (2) We play a one-minute real phone call conversation recording without probe signals using a loudspeaker. After that, we play the same conversation recording with the probe signals added at the beginning. In addition, we ask each volunteer to listen to the two recordings using earphones. (3) We ask the volunteers to fill a questionnaire regarding the annoyance level of the probe signal, assuming that they are the callers who hear the probe signal. We play the probe signal without and with background music in the two rounds. Fig. 22(a) and (b) show the distributions of the volunteers' responses. In the first round without background music, most volunteers find the probe slightly annoying; in the second round with background music, most volunteers find it not annoying at all. The volunteers feel that the probe from the earphone is more annoying than that from the loudspeaker. Some volunteers comment that the probe signal at the beginning of a call sounds a bit weird, but do not feel uncomfortable. Some volunteers mention that the probe signal sounds less annoying after being listened to multiple times. These results are not out of our expectations, because the probe signal's frequencies are similar to the phone keypad tones. (4) Lastly, we ask the volunteers to give an overall rating on whether they accept Telesonar's probing. Fig. 22(c) shows the distribution of the ratings. Without background music, a majority of volunteers rate "likely accept"; with background music, a majority of volunteers rate "very likely accept." These results give useful information for understanding the human callers' acceptance. We believe that Telesonar can easily get acceptance if it is only used to protect vulnerable users such as the elderly and kids.

6 ADAPTIVE ATTACKS

The results in §5 show that Telesonar is effective against the *present* robocalls that play pre-recorded voices or employ the existing voice robots. Such present robocalls have caused widespread threats to the population, especially the vulnerable people. In this section, we discuss several potential approaches that an adaptive adversary can take to further upgrade the robocall system based on the details of Telesonar to bypass detection. We would like to highlight that these attack upgrades are speculative and inexistent at present. Although they may not be fully addressed by Telesonar and its enhancements discussed below, the high costs of implementing the upgraded attacks as analyzed below provide important understanding that they are unlikely launched at large scales.

Replay attack. An adaptive attacker can record the probe signal, process it to generate a signal that mimics the echo. Then, the attacker plays this fake echo in the subsequent calls. As Telesonar only uses the echoes that match in timing and spectrographic shape to confirm echo channel, Telesonar can be enhanced to counteract the replay attack by introducing randomness to the probe signal's timing and spectrographic shape. This enhancement will force the attacker to perform real-time detection of the chirp with an unknown spectrographic shape, which presents technical barriers. The real-time generation of the fake echo also demands significant computing power, similar to the simulating echo channel attack that will be analyzed shortly. Although the robocall operator may use a powerful server for the real-time processing, the total delay is likely to exceed Telesonar's detection window. Alternatively, the attacker may stream a scaled-down version of any received audio back to the callee, which avoids the need of performing real-time chirp detection. To counteract this, Telesonar can perform the probing and detection at random times during the entire call. This will enforce the attacker to perform the straightforward streaming-back all the time, rendering the attack easily noticeable by the callee.

Simulating echo channel. The attacker may simulate the echo channel using a setup similar to `CallSim` to fool Telesonar. However, the simulation incurs compute overhead, e.g., a delay of 40.6 ms on a premium i9-7900X CPU running at its turbo frequency of 4.3 GHz. This delay violates ITU's requirement of 20 ms [32]. The i9-7900X is one of the commodity CPUs with the highest turbo frequencies. The persistent attacker needs to accelerate the execution by parallel computing on multiple CPU cores or more GPU cores due to lower frequencies, which however present technical challenges. In addition, such cores are exclusively used to handle a single robocall session. Non-trivial monetary hardware investment and operating expenses are needed to make massive robocalls in parallel. From this sense, Telesonar, as a low-/zero-cost solution, increases the costs of successful attacks.

Breath timing simulation The adaptive attacker may extract the distribution of breath sound timing from CallHome dataset as Telesonar does. However, the barrier of generating human-like breath sounds with a given distribution is non-trivial. The reasons are two-fold. First, evaluations in §5.3 test Telesonar on the speech generated by the state-of-the-art DNN-based audio synthesis method with natural breath sounds [33]. The model's design in [33] has already considered the timing of breath sound and can be successfully detected by Telesonar. Second, another adaptive attack is to mix recorded breath sounds into the speech audio based on the template distribution of breath timing. However, the unnatural breath sounds can be easily noticed by the callee. The attacker may attempt to re-engineer the approach described in [33] to achieve a target breath sound timing distribution. Since the training data and source codes of [33] are not available (even after request), we were

unable to attempt this. Note that re-engineering the training of the two DNNs in [33] to be bound to a target breath sound timing is a one-time effort and can possibly bypass Telesonar's detection. However the re-engineering it is a non-trivial task. In addition, it is highly likely that the loss functions for training the two DNNs need to be re-engineered.

7 RELATED WORK

This section reviews the existing studies on acoustic voice liveness and genuity detection.

Voice liveness detection. The prevalence of voice controllable systems (VCSs), e.g., Apple Siri, triggers research's attention on their security. The *hidden voice command* attack [12] can generate human-indecipherable sounds that will be interpreted by speech recognition systems as meaningful voice commands. The study [26] leverages the non-linearity of microphone to create high-frequency sounds that are beyond human's hearing range but are recorded as audible sounds by the microphone. As such, attackers may use ultrasonic speakers to send inaudible commands to a victim VCS [42]. Given the above attacks, countermeasures determining the *liveness* of any received voice are important. The liveness here means that the voice is generated by a human in real time. Voiceprint recognition that determines the genuity of voices cannot be employed for liveness detection because it is vulnerable to the replays of genuine voices. Zhang et al. [44] proposed a liveness detection approach that uses a smartphone's two-channel stereo microphones to capture the time-difference-of-arrival (TDoA) of phonemes produced at different physical positions of the user's vocal tract system. However, the approach requires the user to hold the smartphone at a specific position. In [43], Zhang et al. proposed another liveness detection approach that uses the smartphone's loudspeaker to emit an inaudible acoustic tone at 20 kHz and the microphones to capture the reflections from the user's moving articulators when speaking a passphrase. The Doppler frequency shifts due to the articulators' movements suggest liveness. This approach requires that the smartphone is held either to the user's ear or in front of the mouth.

Telesonar can be classified as a liveness detection approach, in that it detects the presence of *live* speaker and microphone at the far end. Telesonar is different from the audio liveness detection approaches in [43, 44], in that it detects the liveness on the far end, whereas the approaches [43, 44] perform local detections only. Thus, the approaches [43, 44] cannot be used to detect robocalls, because we cannot require the attackers' devices to run their detection algorithms.

Voice genuity detection. Since the emergence of voice conversion [30] and speech synthesis technologies [40, 41], speech misuse has been concerned. Identifying synthetic audio has drawn increasing research interests. Various approaches are proposed leveraging extracted features, such as the average inter-frame difference of log-likelihood in [28] and relative phase shift in [15]. However, these approaches tackle known attack techniques only. Recently, GAN-based voice generation like [38] can generate speech which sounds like a real person. DNN-based approaches [22, 27, 35, 36] have been proposed to verify the voice genuity. However, as shown in §2, three recent pre-trained detectors perform very poorly when being tested with a dataset of genuine human voices recorded during phone call sessions. Our results point to the adaptation issue of learning-based approaches. In addition, there are extensive works [19, 23] on the generation of adversarial examples serving as inputs to mislead the classification results of a victim model. Adversarial examples can be generated with either full or no knowledge about the victim model, which can be used to bypass such AI-based robocaller detectors. However, Telesonar exploits the existence of the physical echo channel, which is hard to manipulate for large-scale robocallers. Nevertheless, along with the advance of voice genuity detection, Telesonar provides complementary information regarding the physical setup of the far end, in the pursuit of robocall detection.

8 CONCLUSION

This paper investigated the possibility of detecting the acoustic echo channel at the far end of a voice call. Major challenges are from the AEC mechanism of most audio systems and the use of earphone/headset. As such, we integrate 1) an active detector of transmitting short chirps from the near end and then detecting the echo remnants and 2) passive breath sound detection and timing analysis. We conducted extensive experiments with a simulator and real phones under a wide range of real-world settings. The results show that Telesonar achieves satisfactory detection performance. While Telesonar well suits smartphones, under a broader scope, the information regarding the presence of an acoustic echo channel and also human breath timing at the far end can be used with other forensic metrics such as caller ID, call provenance, and voiceprint to counteract the increasingly misused robocalls for fraudulent and phishing purposes.

ACKNOWLEDGMENTS

Part of this study is supported under the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU). This project is also supported by Direct Grant (4055167) from Faculty of Engineering, The Chinese University of Hong Kong and Kan Tong Po International Fellowship (KTP\R1\221016) from the Royal Society.

REFERENCES

[1] 2018. The Google Assistant can help you get things done over the phone - YouTube. https://www.youtube.com/watch?v=-qCanuYrR0g.
[2] 2018. Voice Phishing Scams Are Getting More Clever. https://krebsonsecurity.com/2018/10/voice-phishing-scams-are-getting-more-clever/comment-page-4/.
[3] 2019. Deep Residual Neural Networks for Audio Spoofing Detection. https://github.com/nesl/asvspoof2019.
[4] 2020. InsightLake. http://www.insightlake.com/call-fraud.html.
[5] 2020. Pindrop. https://www.pindrop.com/.
[6] 2020. Voice Biometrics Group. https://www.voicebiogroup.com/starting/detect-fraud-in-your-contact-center.html.
[7] 2021. Automatic Speaker Verification - Spoofing and Countermeasures Challenge (ASVspoof). https://www.asvspoof.org/.
[8] 2021. Google AI Blog: Google Duplex. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html.
[9] Vijay A. Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael T. Hunter, and Patrick Traynor. 2010. PinDr0p: Using Single-Ended Audio Features To Determine Call Provenance. In *CCS*. ACM, 109.
[10] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
[11] Alexandra Canavan, David Graff, and George Zipperlen. 1997. CALLHOME American English Speech LDC97S42.
[12] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. 2016. Hidden Voice Commands. In *Security*. Usenix.

[13] Federal Trade Commission. 2020. Robocalls. https://www.consumer.ftc.gov/features/feature-0025-robocalls.

[14] Federal Trade Commission. 2021. National Do Not Call Registry Data Book for Fiscal Year 2021. https://www.ftc.gov/reports/national-do-not-call-registry-data-book-fiscal-year-2021.

[15] Phillip L De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga. 2012. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Trans. on Audio, Speech, and Language Processing* 20, 8 (2012), 2280–2290.

[16] Sri Harsha Dumpala and KNRK Raju Alluri. 2017. An algorithm for detection of breath sounds in spontaneous speech with application to speaker recognition. In *International conference on speech and computer*. Springer.

[17] J. Engel, K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts. 2019. GANSynth: Adversarial Neural Audio Synthesis. In *ICLR*.

[18] S. Fleming, M. Thompson, R. Stevens, C. Heneghan, A. Plüddemann, et al. 2011. Normal ranges of heart rate and respiratory rate in children from birth to 18 years of age: a systematic review of observational studies. *The Lancet* 377, 9770 (2011), 1011–1018.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[20] Eberhard Hänsler and Gerhard Schmidt. 2005. *Acoustic echo and noise control: a practical approach*. Vol. 40. John Wiley & Sons.

[21] ITUT ITU-T. 2003. Recommendation G. 114. *One-Way transmission time* (2003).

[22] Alessandro Lieto, Daniele Moro, Francesco Devoti, Claudia Parera, Vincenzo Lipari, Paolo Bestagini, and Stefano Tubaro. 2019. " Hello? Who Am I Talking to?" A Shallow CNN Approach for Human vs. Bot Speech Classification. In *ICASSP*. IEEE, 2577–2581.

[23] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Asia CCS*. ACM, 506–519.

[24] ITU Recommendation. 2015. ITU-R BS. 1770-4. *Algorithms to measure audio programme loudness and true-peak audio level* (2015).

[25] Ricardo Reimao and Vassilios Tzerpos. 2019. FoR: A Dataset for Synthetic Speech Detection. In *IEEE SpeD*. 1–10.

[26] N. Roy, H. Hassanieh, and R. Roy Choudhury. 2017. BackDoor: Making Microphones Hear Inaudible Sounds. In *MobiSys*. ACM.

[27] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. 2015. A Comparison of Features for Synthetic Speech Detection. In *Interspeech*.

[28] Takayuki Satoh, Takashi Masuko, Takao Kobayashi, and Keiichi Tokuda. 2001. A robust speaker verification system against imposture using an HMM-based speech synthesis system. In *Seventh European Conference on Speech Communication and Technology*.

[29] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. 2018. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *ICASSP*. IEEE.

[30] Yannis Stylianou, Olivier Cappé, and Eric Moulines. 1998. Continuous probabilistic transform for voice conversion. *IEEE Transactions on speech and audio processing* 6, 2 (1998), 131–142.

[31] ITUT Switzerland. 1988. G. 711: Pulse Code Modulation (PCM) of voice frequencies. *ITU-T Recommendation G* 711 (1988).

[32] ITUT Switzerland. 1994. G. 174: Transmission performance objectives for terrestrial digital wireless systems using portable terminals to access the PSTN. *ITU-T Recommendation G* (1994).

[33] É. Székely, G.E. Henter, J. Beskow, and J. Gustafson. 2020. Breathing and Speech Planning in Spontaneous Speech Synthesis. In *ICASSP*. IEEE.

[34] The Wall Street Journal. 2019. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case. https://on.wsj.com/38Oj4JA.

[35] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. 2017. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language* 45 (2017), 516–535.

[36] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K.A. Lee. 2019. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *Interspeech*.

[37] Truecaller. 2021. 2021 U.S. Spam & Scam Report. https://truecaller.blog/2021/06/28/us-spam-scam-report-21/.

[38] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In *Arxiv*. https://arxiv.org/abs/1609.03499

[39] The Verge. 2020. How to get the most out of Google Pixel's call screening feature. https://www.theverge.com/2020/2/6/21122390/google-assistant-screen-call-robocalls-spam-phone-pixel. (Accessed on 01/09/2021).

[40] Heiga ZEN. 2007. The HMM-based speech synthesis system version 2.0. *Proc. of ISCA SSW6, Bonn, Germany, Aug. 2007* (2007).

[41] H. Zen, K. Tokuda, and A.W. Black. 2009. Statistical parametric speech synthesis. *speech communication* 51, 11 (2009), 1039–1064.

[42] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *CCS*. ACM.

[43] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *CCS*. ACM.

[44] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. VoiceLive. In *CCS*. ACM.