Void: Voice liveness detection through a spectrogram analysis of voice commands

Anonymous Author(s)

ABSTRACT

Popular mobile devices are now being equipped with voice assistants such as Siri and Google Now to provide new ways to interact with devices using voice. However, due to the open nature of voice channels, adversaries could easily record people's use of voice commands, and replay them to spoof voice assistants. To defend against such spoofing attacks, we present a lightweight and efficient voice liveness detection system called Void (**Vo**ice l**i**veness **d**etection): it exploits different characteristics between human voices and voices replayed through speakers with respect to spectral power patterns analyzed over an audible frequency range to effectively detect voice spoofing attacks. To evaluate the performance of Void, we performed experiments on the two datasets: (1) 229,991 voice samples collected from 120 participants with 15 speakers and (2) 18,016 voice samples in the "ASVspoof 2017" dataset with 42 participants and 26 speakers. For both datasets, Void is capable of achieving accuracy of over 99% and 98% in detecting voice replay attacks with less than 1% and 5.1% equal error rate (EER), respectively. Moreover, we demonstrate that Void is resilient against various forms of adversarial attacks with hidden voice commands and inaudible voice commands – Void achieves 96% and 93% accuracy in detecting even hidden voice command and inaudible voice command attacks, respectively.

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**; **Usability in security and privacy**;

KEYWORDS

Voice replay attack; Voice liveness detection; Spectrogram analysis

1 INTRODUCTION

Voice assistants are becoming ever more powerful and capable of processing advanced commands. Popular services like Siri (Apple), Alexa (Amazon), Now (Google), Cortana (Microsoft), and Bixby (Samsung) allow people to shop online, place phone calls, send instant messages, schedule appointments, check emails, create to-do lists, control smart home appliances, access banking services, and so on. Such security-critical commands need to be protected with authentication schemes. Google's "Trusted Voice" [1], Lenovo's voice-based unlock feature [2], and Tencent's "Voiceprint" feature available in WeChat [3] use voice biometrics to identify and authenticate users. With recent advances in multi-factor authentication, voice biometrics are also used together with secret "wake up" words (voice passwords) to strengthen authentication security.

However, recent studies [8, 9, 23] demonstrated that voice-as-biometric authentication systems are insecure and prone to various forms of voice presentation attacks including replay attack, voice synthesize attack, and voice morphing attack. Among those attacks, the most serious security threat is to use pre-recorded voice samples (intentionally) collected from genuine users to deceive a voice assistant into processing voice commands that appear to be coming from genuine users. This attack is often referred to as a "replay attack," where an adversary tries to spoof speech recognition systems by playing pre-recorded voice samples. Recently, there was an accident in which a TV news report in San Diego about the child that accidentally ordered a dollhouse via Amazon's Alexa inadvertently set off some viewers' Echo devices, which in turn tried to order dollhouses using Alexa [10]. This happening shows a potential risk of replay attacks. Therefore, many tech giants such as Google and Apple have already taken those vulnerabilities seriously, and warned their users about its risks in some cases[1].

To distinguish between legitimate voice samples from a genuine user and the replayed ones, several voice liveness detection techniques (e.g., using an additional accessory device [8]) have been suggested. Although their approach achieved around 97% accuracy, such techniques rely on users carrying a specific hardware device. To improve the detection accuracy without any extra hardware support, deep learning-based approaches [7, 28] were also introduced. The best known solution seems highly effective in terms of detection accuracy (with 6.73% equal error rates (EER)) but is computationally too expensive to deploy; four machine learning models including three deep learning models (two CNNs and one RNN) and one SVM model were required.

To reduce such computational burden and achieve high accuracy in detecting voice replay attacks, we present Void (**Vo**ice l**i**veness **d**etection), an efficient voice liveness detection system based on the "cumulative power of each frequency" in spectrograms.

Void was designed based on two essential characteristics that differentiate human voices from machine-generated sounds including replayed human voices: (1) Most loudspeakers aim to play sound over given reproducible frequency ranges accurately, and *inherently* adds additive noises at various frequencies to achieve that. Consequently, the overall power over the audible frequency range is scattered with some uniformity. (2) With human voice, power in lower frequencies is relatively higher than that in higher frequencies [11, 27]. Due to those two characteristics, we found that there are significant differences between human voices and replayed voices in cumulative power distribution over signal frequencies – our approach exploits such differences to detect replayed voices. Our experiment results, conducted on voice samples collected from 120 participants under various conditions (e.g., with/without background noise) and a large-scale dataset of voice samples (the "ASVspoof 2017" dataset [27]), demonstrate the feasibility of Void with various loudspeakers and environmental conditions. Our key contributions are summarized as follows:

[1] While enabling the Trusted Voice feature on Nexus devices, Google explicitly warns that it is insecure than password and can be exploited by the attacker with a very similar voice.

- We propose a spectral power-based voice presentation attack detection system that does not require any additional hardware, and solely uses frequency and cumulative power distributions to generate key features that differentiate human and machine – resulting in a solution that does not require heavy and complex computations, and performs decisions in real-time with minimal delays (see Section 5).
- We evaluate Void on two different datasets – with varying demographics, distances, speakers, and background noises – to show that it can overall achieve over 99% of accuracy for varying distances and speaker genders. Void is particularly capable of achieving accuracy of over 99% (with less than 1% EER) to classify 229,991 voice samples with 120 participants, 14 different built-in speakers and 4 different high-quality standalone speakers, and over 98% (with 5.1% EER) to classify 18,016 voice samples (in the "ASVspoof 2017" dataset [27]) with 42 participants, 8 different built-in speakers and 18 different high-quality standalone speakers, respectively (see Section 7.2 and 7.3).
- We show that Void is also resilient to hidden and inaudible voice command attacks [16–18, 24, 25] that exploit sound samples that are difficult to understand (by human listeners) but are still recognized as valid voice commands. We show that Void achieves an accuracy of 96% in detecting hidden voice commands, and an accuracy of 93.3% in detecting inaudible voice commands (see Section 7.4).
- We show that Void can be implemented efficiently. Compared with the existing machine learning based approaches (e.g., CQCC in 2017 ASVspoof challenge [7]), Void is significantly efficient and lightweight. In our prototype implementation, the number of features used in Void is 167 while the number of features used in CQCC is 25,560. Moreover, Void can detect replay attacks with only 3.5 seconds for training time and 0.18 seconds for testing time (on average) while CQCC takes 929.97 seconds for training time and 0.63 seconds for testing time (see Section 7.2).

2 THREAT MODEL

Voice authentication is the process of verifying a user's identity by extracting acoustic features that are related to the user's behavioral and physiological characteristics.

There exist various voice impersonation attack methods to generate voice samples that resemble a victim's voice. However, we specifically target two types of machine-based voice impersonation attacks: (1) voice replay attacks [5, 8, 9] and (2) hidden [24, 25] and inaudiable voice attacks [16–18].

2.1 Voice replay attacks

An attacker uses a recording device in a close proximity to a victim, and records the victim's utterances (spoken words) used to interact with voice assistants [5, 8, 9]. To replay the recorded voice commands, the attacker can use either built-in, low-quality speakers on her phone or external, high-quality speakers (e.g., Bose speaker).

Voice replay attack is the most accessible (easiest) attack to perform, but it is also the most difficult attack to detect due to the similarities between a victim's real voice and recorded (and replayed) voice.

2.2 Hidden and inaudible voice command attacks

We also consider more sophisticated attacks (called *hidden* and *inaudible* voice command attacks) [16–18, 24, 25] that were recently developed to secretly deliver voice commands to a victim's voice assistant.

An adversary equipped with a sophisticated playback device can generate sound samples that are unintelligible or inaudible to human listeners but can be interpreted as valid commands by the target voice assistant. Consequently, those hidden and inaudible commands are executed by the voice assistant without the device owner's awareness. Those hidden and inaudible voice commands could be created using sophisticated machine learning techniques through either black-box model or white box model [25]. Inaudible voice commands can be particularly generated by exploiting hardware non-linearity with a ultrasonic speaker and could affect the performance of Void. Recent studies [16–18] have successfully demonstrated the feasibility of such attacks.

Therefore, we evaluate the performance of Void against those two types of voice command attacks, hidden [24, 25] and inaudiable voice command attacks [16–18], respectively.

3 REQUIREMENTS

3.1 Latency and resource usage requirements

Existing commercialized voice assistants (e.g., Siri and Google Now) utilize server-based speech and voice recognition technologies, and make intensive use of computational resources including CPU, GPU, memory, and data storage. Our conversations with several engineers at a large IT company (that runs their own voice assistant service) revealed that there are latency and computational power usage requirements that must be considered upon deploying any kind of machine learning-based services. This is because additional use of computational power and memory through continuous invocation of machine learning algorithms may incur unacceptable costs for businesses, and unwanted latency for processing voice commands. A single GPU may be expected to concurrently process 100 or more voice sessions (streaming commands), indicating that machine learning algorithms must be *lightweight*, *simple*, and *fast*.

Hence, businesses are very strict about adding and continuously running new data-driven algorithms on their already exploding GPUs and CPUs. If deployment of new data-driven analytic services are necessary, businesses expect algorithms to be optimized through the use of computationally efficient algorithms, minimal number of features, and minimal number of machine learning algorithms.

3.2 Detection accuracy requirements

Our main objective is to achieve higher accuracy while keeping the latency and resource usage requirements at modest level. Therefore, our goal is to opt for a computationally efficient machine learning solution which could be practically deployed and achieves higher accuracy. The primary security goal of Void is to achieve an EER

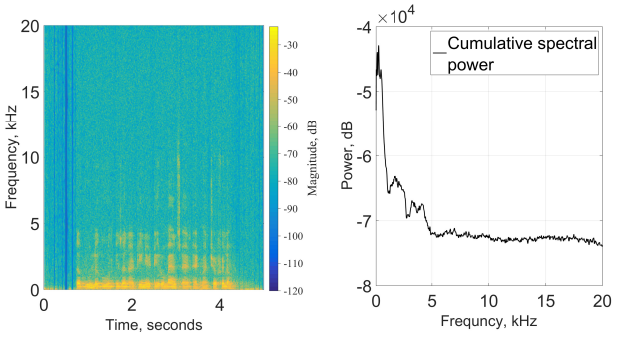

Figure 1: Spectrogram of an example phrase *"The Blue Lagoon is a 1980 romance and adventure film."* uttered by a live user (left) and cumulative power spectral decay of the corresponding command (right).

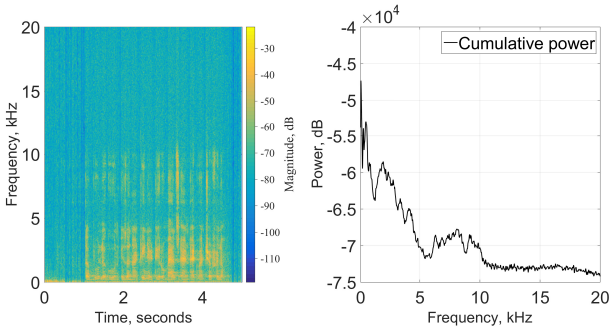

Figure 2: Spectrogram of the same example phrase (as in Figure 1) replayed using iPhone 6S plus (left) and cumulative power spectral decay of the corresponding command (right).

that is similar to or lower than the EER (6.73%) of the best performing algorithm [28] from the 2017 ASVspoof challenge. Considering that their deep learning algorithm used a combination of multiple (heavy) classification techniques[2], achieving 10% (or lower) EER with the use of more computationally lightweight and fast algorithms would be a practical and scalable solution for businesses to consider.

4 DIFFERENCES IN VOICE CHARACTERISTICS

In this section, we discuss frequency-specific spectral power characteristics between human voices and voices replayed through loudspeakers.

4.1 Characteristics of human voice

Human voice frequency is a part of human sound production process, in which vocal cords are the primary sound source. The mechanism for generating human voice consists of three main parts: lungs, vocal folds within larynx (also called voice box), and articulators.

Human male voice covers a frequency range of 80Hz to 8kHz, and female voice covers between 350Hz to 17kHz [13]. According to the Nyquist-Shannon sampling theorem, the sampling frequency f_s must be at least twice the highest component of voice frequency for effective reconstruction of voice signals. Thus, f_s needs to be at least 34kHz to cover both adult male and female voices. Also, the fundamental frequency of a typical adult male's voice ranges from 85 to 180Hz, and that of a typical adult female's voice ranges from 165 to 255Hz [20, 21]. Thus, the fundamental frequency of most speeches fall below the bottom of the "voice frequency" range.

The voice signals received by a device also contains certain additive noises and microphones' non-linear behaviors[3], therefore, we expect that spectrograms of the received audio signals will show extra frequency components. As a result, the accumulated power over certain frequencies would increase due to additive noise signals. Figure 1 (left) shows the spectrogram of a voice command *"The Blue Lagoon is a 1980 romance and adventure film"* uttered live by human, and processed by an audio chipset in a laptop. The sampling frequency f_s for the phrase utterance was set to 44.1kHz, and the utterance duration was 5 seconds. It is clear that most of the spectral power lies in the frequency range between 20Hz and 1kHz. The cumulative spectral power contributed by each frequency is also shown in Figure 1 (right). There is an exponential power decay of human voice at frequency around 1kHz. Void utilizes such characteristics to classify human voices.

4.2 Characteristics of loudspeakers

To perform a voice replay attack, an adversary is required to convert digital voice signals into audible sounds using a conventional loudspeaker. Loudspeakers' manufacturers often use the term *frequency response* to describe the frequency range that a loudspeaker can reproduce. The frequency response of a loudspeaker depicts how strong a loudspeaker can reproduce sound across audible frequency range[4]. The frequency response curve is usually displayed higher for frequencies played with high volume, and displayed lower for frequencies played with lower volume [19]. Typically, the frequency response for a loudspeaker varies from 3 to 30dB, dropping off drastically at very low bass and very high frequencies. Generally, a flat frequency response indicates that a loudspeaker is able to reproduce sounds accurately[5]. However, it is not practically possible due to imperfections in speaker manufacturing processes and non-linear characteristics in electronic components such as microphones, amplifiers, and loudspeakers [12, 19].

Figure 2 shows spectrogram and the corresponding cumulative spectral power of a voice phrase played back through an iPhone 6S Plus. Note that the spectrogram shows some uniformity in the

[2]Top performing approaches in the 2017 ASVspoof challenge employ multiple subsystems and classifiers with computationally intensive features. These approaches use multiple classifiers with a large number of features extracted from MFCC, IMFCC, CQCC PLPCC, etc. At the decision making step, output scores from all classifiers are fused using the Bosaris toolkit [14].

[3]Due to the inherent non-linear characteristic of microphones, they produce additional signals in lower frequency ranges [15, 16].

[4]The theoretical range of human hearing is generally from 20Hz (low bass tone) to 20kHz (highest treble notes). Thus, a loudspeaker should be sensitive to voices in that frequency range.

[5]For an ideal speaker, the perfect frequency response plot would look like a flat line across the entire audible frequency range.

power distribution between 1 and 5kHz. We observe that the cumulative spectral power (see Figure 2 (right)) does not show an exponential decay; instead, it shows more like a linear decay between 1 and 5kHz.

4.3 Key insights

We observe that human voices and voices replayed through loudspeakers show uniquely different patterns of power decay. Based on our observations, we present the following two key insights:

(1) With human voices, we observe that most of the signal's power is concentrated in lower frequencies, specifically below 1kHz. This observation is aligned with the findings from [11], which states that power in lower frequencies is greater than that of higher frequencies. However, in the replayed version of the same voice phrase using portable devices with built-in speakers, the power is spread out over the range of frequencies between 20Hz and 10kHz (see Figure 2). Portable devices with built-in speakers show almost similar behavior in their spectral power distribution, i.e., power decay rate is not exponential, but gradually decreases over frequencies. The possible explanation for such power spread over the wide range of audible frequencies could be due to low-quality hardware boosting certain frequencies, such as frequency response phenomena of loudspeakers or hardware non-linearity. We found similar power decay patterns in 14 portable devices (10 smartphones and 4 laptops) with built-in speakers. Therefore, one key insight is in detecting replay attacks launched from portable devices is by detecting linear decay in spectral power over the range of frequencies (see Appendix A). The distinguishing characteristics of power decay pattern in replay attacks from built-in speakers and live-human enable us to detect attacks launched using built-in speakers. Because most of the power is concentrated in lower frequencies (exponential power decay) in the case of live-human voices whereas the power decay shows more of a linear trend in the case of replayed attacks from built-in speakers. Thus, by analyzing power patterns over audible frequency range, replay attacks launched using built-in speakers of smartphones, tablets, laptops could be detected with high accuracy.

(2) We also observe that the power decay patterns of high-quality standalone speakers are not similar to those of built-in speakers on portable devices. In practice, voice commands replayed using a Bose speaker [22] are rather similar to those in live-human voices; i.e., most signal power is concentrated at frequencies below 1kHz, thus making it difficult to detect such high quality loudspeakers. However, in the case of live-human voices, the sound signals directly go into the receiver's microphone with less degradation in sound quality. However, in the case of replayed voice samples from high quality standalone speakers, the quality of received voice samples are generally more degraded than live-human case due to additive noise signals and hardware non-linearity. In other words, power patterns in the replay attacks from high quality standalone speakers are quite non-deterministic compared with those in live-human voices.

In order to verify the validity of our key insights, we analyzed the cumulative spectral power behaviours over different recording and playback devices and confirmed that our key insights still remained valid under various experiment conditions. For recording human voices, we used three different laptops with a sound card manufactured from different vendors.

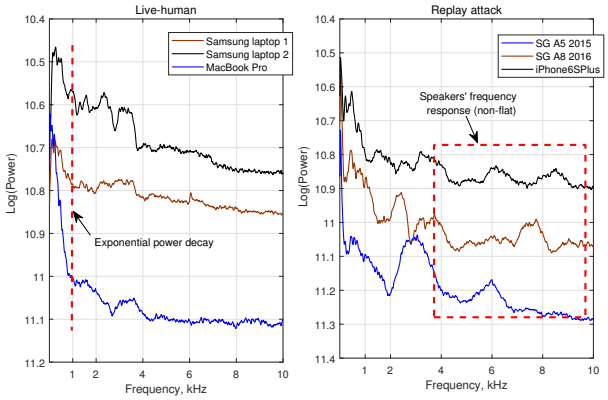

Figure 3: Diversity in high-quality recording and playback devices. Live-human: Two Samsung and one MacBook Pro laptops are used to record live-human voices (left). Replay attack: Three smartphones (Samsung Galaxy A5, Samsung Galaxy S8 and iPhone 6S Plus) are used to record live-human voices and replay the recorded voices (right).

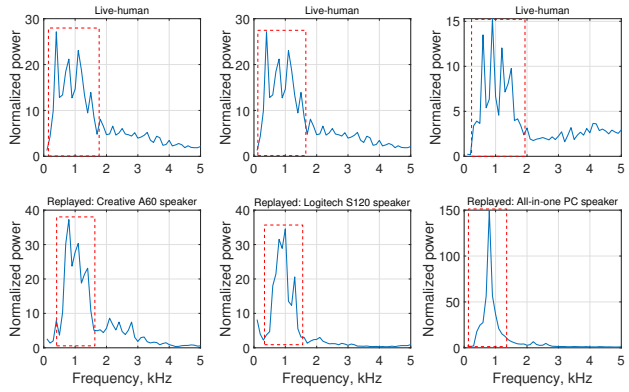

Figure 4: Signal power frequency range between 20Hz and 5kHz. Live-human (top three): fine-grained power fluctuations can be observed over the frequency range from 20Hz to 2kHz. High-quality standalone speakers (bottom three): the power over the same frequency range is more concentrated with less fluctuations in power.

The log-scaled power behaviors over frequencies are presented in Figure 3 (left). There exist locally slight differences between the three laptops, but their overall patterns remains similar, i.e., the exponential decay of power in the lower frequencies could be observed clearly. We can see that the slope of decay rate in spectral

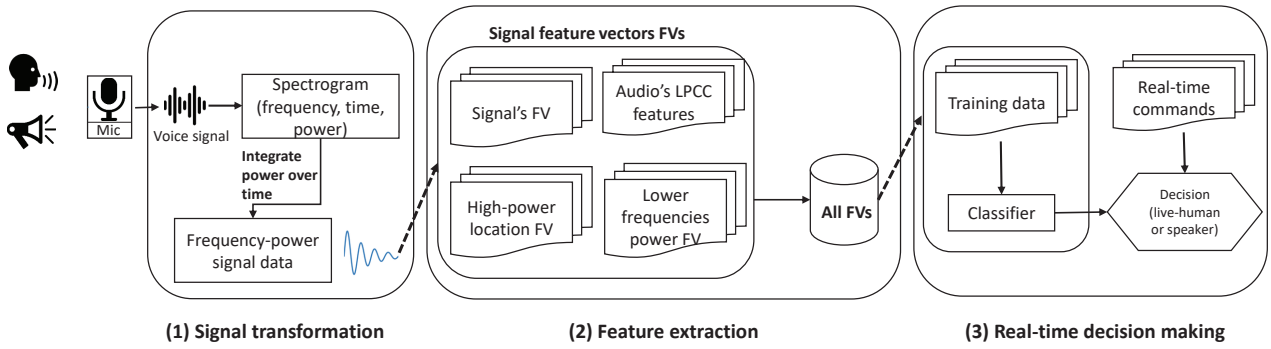

Figure 5: High-level design of Void, consisting of spectral power analysis over audible frequency range.

power is greatly changed at around 1kHz (see the left side of the red-dotted line in Figure 3 (left)). The key point here is to observe that live-human voices recorded at laptops' microphones show an exponential decay in spectral power at around 1kHz. It seems natural because human voice frequencies mostly fall in the lower frequencies [13].

Figure 3 (right) shows the spectral power behaviors of voice samples playbacked from three different smartphones (Samsung Galaxy A5 2015, Samsung Galaxy S8 2016, and iPhone 6S Plus). Unlike live-human voices (Figure 3 (left)), certain frequencies above 4kHz are showing significant variations in power. We can also see that the power decay in lower frequencies does not show a exponential power decay pattern, but is more like linear between 20Hz to 4kHz.

Figure 4 shows normalized signal power for live-human voices and voices replayed through three different high-quality standalone speakers, respectively. We can observe that most of the power is concentrated in the lower frequencies for both live-human voices and voices replayed. However, we can also see that the power patterns of live-human voices are quite different from those of the voices replayed at higher frequencies (see red-dashed rectangles in Figure 4). These differences in power patterns could be exploited to effectively detect replay attacks from high-quality standalone speakers.

5 SYSTEM DESIGN

Void is designed to satisfy the requirements specified in Section 3 based on two key ideas (see the details in Section 4.3): (1) examining signal power distributions over the audible frequency range, and computing the linearity degree of signal power in order to detect voices replayed through low-quality loudspeakers, and (2) analyzing low-power and high-power frequencies to effectively detect voices replayed through high-quality loudspeakers.

Void consists of three stages as illustrated in Figure 5: signal transformation stage (see Section 5.1), feature extraction stage (see Section 5.2), and decision making stage (see Section 5.3). The overall procedure of Void is described in Algorithm 1. The voice command $Voice_{\text{in}}$, window size W and a weighting factor ω are given as the input to Algorithm 1. In this algorithm, we use the following notations. S_{pow} represents the cumulative spectral power density per frequency of $Voice_{\text{in}}$ (see an example of S_{pow} in Figure 3). W represents the size of a single segment of S_{pow} to properly capture the dynamic characteristics of S_{pow} with a small number of segments. A weighting factor ω between 0 and 1 is used to calculate a threshold for feature values at higher frequencies. Those parameter values were determined experimentally with a large number of test samples. Last, $pow(i)$ in Algorithm 1 represents the accumulated power in i_{th} segment of S_{pow}. We consider only voice signals below 15kHz because most of the signal power for voice commands lies with in 15kHz range.

Algorithm 1 Void's overall procedure.

Input: $Voice_{\text{in}}$, W and ω
Output: `live-human` or `replayed`
Stage 1: Signal transformation
1: Compute STFT of for input voice command $Voice_{\text{in}}$
2: Compute S_{pow} from STFT
Stage 2: Feature extraction
3: Divide S_{pow} into k segments where $k = \lfloor \frac{\text{size}(S_{pow})}{W} \rfloor$.
4: **for** i_{th} segment Seg_i from $i = 1$ to k **do**
5: $pow(i)$ = the sum of power in Seg_i.
6: $< pow >$ = Vectorize($pow(1), \cdots, pow(k)$)
7: $FV_{LPF} = < pow >$
8: FV_{LDF} = LinearityDegreeFeatures($< pow >$)
9: FV_{HPF} = HighPowerFrequencyFeatures($< pow >, \omega$)
10: Compute LPC coefficients of $Voice_{\text{in}}$ and store the results as FV_{LPC}
Stage 3: Decision making
11: $FV_{\text{Void}} = \{FV_{LPF}, FV_{LDF}, FV_{HPF}, \text{and } FV_{LPC}\}$
12: Run SVM classifier with FV_{Void} and provide the class label (either `live-human` or `replayed`) as output

5.1 Signal transformation

Our system mainly relies on the spectral power analysis of audio signals. In the signal transformation stage, given an input voice command signal $Voice_{\text{in}}$, the short-time Fourier transform (STFT) is applied to obtain the cumulative spectral power density per frequency of $Voice_{\text{in}}$ over time[6]. The power in each frequency band is summed over time (see the Step 1-2 in Algorithm 1). Thus signals

[6]The terms 'cumulative spectral power' and 'power' are used hereafter interchangeably.

are transformed into two dimensions containing frequency and corresponding power. The obtained signal spectrogram contains frequencies and the corresponding power (dB) over time (see Figure 1 and 2).

5.2 Feature extraction

The vector S_{pow} computed from the first stage are used as the input to the second stage to extract features for classification.

For feature extraction, Void sequentially computes the following four classes of features: lower power frequencies features (FV_{LPF}), signal power linearity degree features (FV_{LDF}), higher power frequencies features (FV_{HPF}), and audio signal's linear prediction cepstrum (LPC) coefficients features (FV_{LPC})[7]. The first three feature classes are computed from S_{pow} while FV_{LPC} is computed directly from the raw voice command signal $Voice_{\text{in}}$.

5.2.1 Lower power frequencies features: In the second stage of Algorithm 1, we first divide the signal S_{pow} into k short segments of equal-length according to the given window size W (see the step 3 in Algorithm 1). If the size of S_{pow} is not divisible by W, we just omit the last segment. Next, we compute the sum of power in each segment Seg_i for $i = 1$ to k (see the step 4 and 5 in Algorithm 1). We then vectorize the first k segments of power density values as $< pow >$ (see the step 4 in Algorithm 1). The vector $< pow >$ is directly used for FV_{LPF} (see the step 5 in Algorithm 1). At this step, we obtained the cumulative spectral power density values for all k segments. Note that power density values for each segment are in order of increasing frequency, starting from the lowest frequency of a voice sample. However, we are only interested to retain power density values only within 5kHz because we experimentally observed that there is a clear difference between live-human voices and replayed voices in the power pattern in lower frequencies below 5kHz.

5.2.2 Signal power linearity degree features: Given the vector $< pow >$ of k segments' power density values, we compute the signal's feature vector (FV_{LDF}) to measure the degree of linearity of spectral power that can be particularly helpful to detect voices replayed from portable devices with built-in speakers showing high linearity of spectral power over the audible frequency range (see Appendix A).

Algorithm 2 LinearityDegreeFeatures

Input: $< pow >$
Output: FV_{LDF}={ρ, q}.
1: Normalize $< pow >$ with $sum(< pow >)$ to obtain $< pow >_{\text{normal}}$
2: Accumulate the values of $< pow >_{\text{normal}}$ to obtain pow_{cdf}
3: Compute the auto-correlation coefficients of pow_{cdf} and store the results as ρ
4: Compute the quadratic coefficients of pow_{cdf} and store the results as q

Algorithm 2 describes the procedure for computing the linearity degree of $< pow >$. Initially, $< pow >$ is normalized by dividing each value in $< pow >$ by the total signal power ($sum(< pow >)$) (see the step 1 in Algorithm 2). The normalized power signal vector $< pow >_{\text{normal}}$ is then used to compute the cumulative distribution of spectral power, denoted by pow_{cdf} (see the step 2 of the Algorithm 2). In this step, $< pow >_{\text{normal}}$ is accumulated in a step-wise fashion.

For the linearity degree of pow_{cdf}, we compute the following two features (see the step 3 and 4 in Algorithm 2): auto-correlation coefficients ρ and quadratic curve fitting coefficients q of pow_{cdf}. Auto-correlation of a cumulative distribution can be used to quantify the linearity of the cumulative distribution. However, we found that ρ is not highly sensitive in identifying the distinguishable exponential growth of power in live-human voices at frequencies between 20Hz and 1kHz (see Appendix A). Therefore, we introduce the quadratic curve fitting coefficients q of signal pow_{cdf} as another metric to quantify the degree of linearity for the cumulative distribution function. Finally, all the computed coefficients {ρ, q} are stored as FV_{LDF}.

5.2.3 Higher power frequencies features: Given the vector $< pow >$ of k segments' power density values and the peak selection threshold ω, we compute the feature vector (FV_{HPF}) to capture the dynamic characteristics of spectral power in higher frequencies.

As discussed in Section 4.3, live-human voices and voices replayed through high-quality standalone speakers show quite similar patterns in the overall distribution of spectral power. Therefore, the only features for measuring the degree of linearity in the cumulative distribution function of power may not be sufficient to distinguish live-human voices from voices replayed through high-quality standalone speakers with high accuracy.

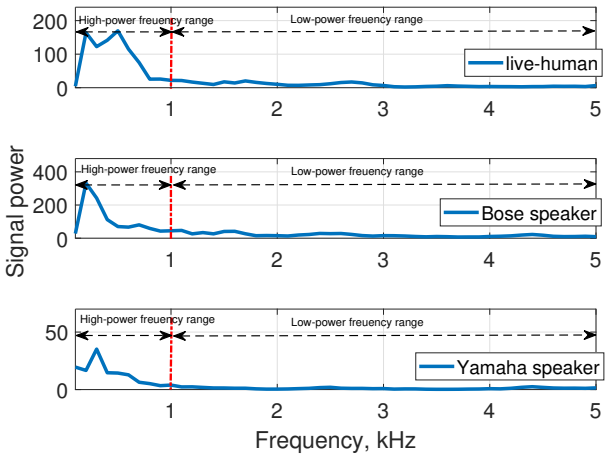

Figure 6: Distribution of normalized spectral power over frequencies where the total frequencies in audio signal are up to 8kHz and $W = 10$.

Figure 6 shows that there is no clear distinguishable pattern between live-human voices and voices replayed through loudspeakers (Bose and Yamaha) in lower frequencies above 1kHz. That is, the spectral power of those signals is mainly concentrated at frequencies below 1kHz. Therefore, we also need to focus on the features to

[7] FV stands for feature vector.

reflect their relative differences in higher frequencies below 1kHz. Interestingly, in the case of voices replayed using loudspeakers, there is only one single peak in the frequency range below 1kHz while there is more than one peak for live-human voices (see Figure 6). Therefore, the number of peaks in the frequency range below 1kHz can also be used as an important feature for classification. Please refer to Section 4.3 where we have discussed the reason power peaks in spectral power density.

Algorithm 3 HighPowerFrequencyFeatures

Input: $< pow >$ and ω
Output: $FV_{HPF} = \{N_{peak}, \mu_{peaks}, \sigma_{peaks}\}$.

1: Find peaks from $< pow >$ and store the discovered peaks $\{(peak_1, loc_1), \cdots, (peak_n, loc_n)\}$ as S_{peak} ▷ n is the number of peaks discovered in $< pow >$
2: $T_{peak} = \omega \cdot max(peak_1, \cdots, peak_n)$
3: **for** each $peak_i$ in S_{peak} from $i = 1$ to n **do**
4: **if** $peak_i < T_{peak}$ **then** remove $peak_i$ from S_{peak}
5: N_{peak} = the number of peaks in S_{peak};
6: μ_{peak} = the mean of the locations of peaks in S_{peak}
7: σ_{peak} = the standard deviation of the locations of peaks in S_{peak}

Algorithm 3 describes the procedure for computing signal's higher power frequency features (FV_{HPF}). In $< pow >$, we first identify peaks and their locations (see the step 1 in Algorithm 3). Our peak selection criterion T_{peak} has a nice property that it automatically scales itself with respect to audio signal's spectral power density values. For example, for low/high powered voice signal, the T_{peak} is computed accordingly, as shown in the step 2 of Algorithm 3. From our experiments on both the datasets, we observe that detected peaks from live-human voice samples and replayed samples show different characteristics when we set $\omega = 0.6$. However, ω needs to be set such that high power frequency feature set play its role in detecting voice replay attacks. We set a threshold to filter out insignificant peaks by multiplying $max(S_{peak})$ by a given weighting factor ω where $0 \leq \omega \leq 1$ (see the step 2, 3 and 4 in Algorithm 3).

For FV_{HPF}, we first count the number of peaks in S_{peak} and store the count of peaks as N_{peak} (see the step 5 in Algorithm 3); the mean and standard deviation of locations of the discovered peaks are sequentially computed and stored them as μ_{peaks} and σ_{peaks}, respectively (see the step 6 and 7 in Algorithm 3).

5.2.4 Linear prediction cepstrum coefficients: Linear predictive coding based approaches are most widely used in speaker recognition and verification, speech coding, speech synthesis, speech recognition, and etc. In particular, LPC coefficients are often used to detect voice replay attacks [7].

The basic idea behind LPC is that the current speech sample can be approximated as a linear combination of previous samples. The predictor coefficients for a voice sample is computed by minimizing the sum of squared differences between the actual speech samples and linearly predicted ones. We use the auto-correlation method with Levinson-Durbin algorithm [26] to compute the LPCC feature vector FV_{LPC}.

5.3 Decision making

In the third stage of Algorithm 1, we construct a classifier with the feature sets computed in Section 5.2 to detect voices replayed through loudspeakers. Instead of manually constructing rules for a comprehensive logic of decision making, we opted to utilize a machine learning-based classifier to develop Void in a systematic fashion. The description of our machine learning model is as follows:

- **Feature set:** The four feature vectors $FV_{LPF}, FV_{LDF}, FV_{HPF}$, and FV_{LPC} are combined to constitute a feature set for machine learning algorithm. The total number of feature is 167 where the detailed summary of FV_{HPF} and FV_{LDF} feature vectors is given in Appendix B.
- **Classifier:** We employed support vector machine (SVM) as the classifier for two class (live-human and replayed). We tested Void on various models including generative (Gaussian mixture model) and discriminative (SVM and kNN) models for evaluation . We believe that generative models such as Gaussian mixture model is not suitable for Void because they rely on generating feature sets values based on the observed values. The generated features might not accurately represent our feature sets. In our experiments, we rely on SVM which is light-weight and quick in decision making, and therefore is a good choice for real time applications. An SVM classifies dataset by finding the best hyperplane that separates all feature points of one class from those of the other class. Moreover, it is capable of deducing linear relations between the the cross-correlation values that define the feature vector.

6 DATASETS

This section describes human voice samples and replayed samples we collected using multiple recording and playback devices, and under varying conditions. We also explain a publicly available database of replayed voice samples [7] that we used for evaluation.

6.1 Data collection and demographics

For voice sample data collection, we prepared 10,000 different real-world voice commands that are actually understood by an existing voice assistant implementation. The voice commands were mixed in lengths (ranging between 2 to 5 seconds) and command types (e.g., setting alarms, calling contacts, opening emails, etc.). We then recruited participants from two different user pools.

For the first data collection, we recruited 100 participants from a large IT company, asking each participant to say 100 different voice commands from the prepared list. Two recording devices, Samsung Galaxy S8 and Apple iPhone 8, were used to record all voice and both devices being placed about 20 centimeters away from each participant. About 53% of the participants were male, equally covering voice frequency ranges from both males and females [13]. Most of the participants were in the 40-49 (13%), 30-39 (62%), and 20-29 (25%) age groups.

We explicitly informed the participants that the purpose of experiments is to collect voice samples to develop and evaluate a voice liveness detection solution. Ethical perspective of our research was validated through an institutional review board (IRB) at a university.

To generate replay attack dataset, we replayed all 10,000 collected voice samples in an lab environment under various conditions as described below:

- **With or without background noise:** We replayed all voice samples during the day when there were people using the office to record replay attacks with natural background noise. We also replayed all voice samples late at night when the office was not being used to record replay attacks with minimal background noise – ambient environmental noises such as central AC and computer CPU noises were still present though.
- **Distances between the target device and speech source:** Distances between the target device (used to record replayed voice samples) and the speech source (replaying device) could affect the detection accuracy – this is because the proposed approach relies on spectral power features and power patterns could be affected by varying distances. Hence, we recorded replayed voice samples using three identical devices, Samsung Galaxy S8, which were located 10 centimeters, 1 meter, and 2 meters away from each speech source.
- **Cross-dataset training:** We trained and tested voice samples collected from one specific set of participants, target (recording) devices, and speech sources (replaying devices) using 10-fold cross validation for both our dataset and the publicly available dataset.
- **Speech source types:** We also used 10 different types of smartphone built-in speakers, and 4 different types of loudspeakers as shown in Appendix C to replay recorded voice samples (speech sources). Each loudspekaer was different in terms of the number of sound channels supported, brand, price (some were high-end speakers), and electrical power.

For the second data collection, we recruited 20 participants from a university, and asked each participant to say 20 different commands from the prepared command list. 14 participants were graduate students, and their age ranged from 25 to 36. 16 participants were male. We used a slightly different human voice record setting: each participant was asked to say a command 1 meter away from the recording devices (multiple laptops in this case), and repeat the same command 3 meters away from the same recording devices. Replaying human voice samples and recording them were performed as described above.

We merged the two datasets, and used the merged dataset for our evaluations.

6.2 ASVspoof dataset

In addition, we also used an online replay attack database called the "ASVspoof 2017 dataset" [7, 27]. It has been created to facilitate a competition to detect genuine human voice samples from samples reproduced using laptops and smartphones' built-in speakers as well as standalone speakers. It is a large database containing voice samples collected from 179 replay attack sessions, played back with 125 unique replay configurations – such configurations vary in recording, playback, and background environment settings. A session consists of sets of voice samples that were recorded under the same replay configuration. Moreover, voice samples were collected from numerous environments, including a balcony, bedroom, canteen, home, office, and open lab space. The details of both dataset are presented in Table 1 for comparison.

Table 1: Dataset description.

Item	Detail	Our dataset	ASVspoof
Data	Samples	229,991	18,016
	Training ratio	10-fold CV	10-fold CV
	Participants	120	42
Devices	Replay	14	26
	Recording	15	25
Replay configurations		33	125
Speakers	Built-in	10	8
	Standalone	5	18

7 EVALUATION

This section presents our evaluation results, including the attack detection accuracy and time taken to train models and classify attacks.

7.1 Setup details

As for data collection, we followed the procedures and experimental conditions described in Section 6, and used a sampling frequency of 44.1kHz to record all human voice samples. All of the built-in speakers and standalone speakers used for replaying recorded voice samples are listed in Appendix C.

We used SVM as our classification algorithm as it is known for its robust performance when a given dataset is large. SVM is also light-weight, and fast in classifying input data, making it suitable for real time applications like Void. We first performed 10-fold cross validation on the collected datasets, and applied SVM with linear kernel to train the classification models.

All experimented were conducted on a desktop PC equipped with Intel(R) Core(TM)i5-6500 3.2 GHz CPU, and 16 GB of main memory. 64-bit Windows 10 operating system was installed on it. For computing EER, we used the Bosaris toolkit [14], which was suggested in the ASVspoof Challenge [7].

We use four possible performance metrics as shown in Table 2. "True acceptance" (TA) and "true rejection" (TR) refer to correctly detecting live-human and loudspeaker, respectively. "False acceptance" (FA) is when loudspeaker is classified as live-human, and "false rejection" (FR) occurs when live-human is classified as loudspeaker. To measure the performance, we rely on the standard automatic speaker verification metrics, which are false acceptance rate (FAR) and false rejection rate (FRR). A replay attack is a success if Void classifies it as a live-human voice. We also present EERs, which are values for which the proportion of FAR is equal to the proportion of FRR. An authentication system with 100% accuracy would have EER of zero.

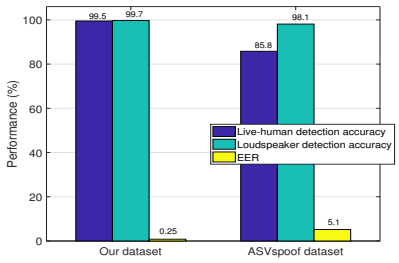

Figure 7: Overall performance results of replay attacks launched from built-in speakers in portable devices (e.g., smartphones and laptops) and high-quality standalone speakers on both datasets.

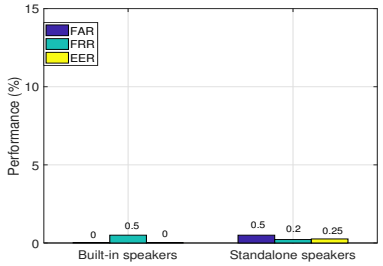

Figure 8: Our dataset: overall attack detection rate when attack launched using smartphones and high-quality standalone speakers.

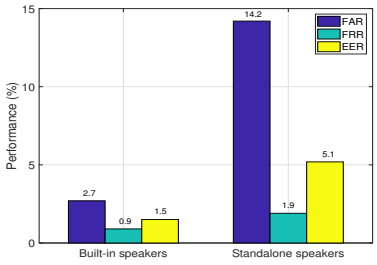

Figure 9: ASVspoof dataset: overall attack detection rate when attack launched using smartphones and high-quality standalone speakers.

Table 2: Decision outputs.

	Accept	Reject
Live-human	True Acceptance	False Rejection
Replay attack	False Acceptance	True Rejection

7.2 Overall performance

We present the overall performance of Void in detecting voice replay attacks based on the performance metrics described in Table 2. Figure 7 shows the performances of Void on both datasets (ours and the ASVspoof dataset). From this figure, we can see that the performances for classifying all built-in and standalone speaker voice samples together: for our dataset, live-human voice detection accuracy is 99.5%, and voice replay attack detection accuracy is 99.7%. As a result, the EER for our dataset is below 1% (see Figure 8).

As for the ASVspoof dataset, Void achieves 85.8% accuracy for detecting live-human voices, and 98.1% accuracy for detecting replay attacks. The EER on the ASVspoof dataset is 5.1%, which is larger than that of our own dataset but is still a competitively low error rate. The reason for this performance degradation is due to the fact that there is a much larger number of replay configurations (125) in the ASVspoof dataset being tested (see Table 1).

7.2.1 Performance on our dataset. Figure 8 shows the Void performance on our own dataset in two parts: one for built-in speakers and another for standalone speakers. The FAR for detecting replay attacks originating from built-in speakers and standalone speakers are 0% and 0.5%, respectively. The FRR for built-in speakers and standalone speakers are 0.5% and 0.2%, respectively. The EER for built-in speakers and standalone speakers are 0% and 0.25%, respectively. The EER for standalone speakers is still very low considering the variances we introduced in the dataset (multiple recording and replaying devices, varying distances, with and without background noises and so on.). Since spectral power distribution of standalone speakers is similar to that of live-human, linearity degree features (FV_{LDF}) alone are not sufficient. The reason Void still achieves high accuracy is due to the additionally used feature vectors (FV_{LPF}, FV_{HPF}, and FV_{LPC}).

7.2.2 Performance on ASVspoof dataset. Figure 9 shows the Void performance for ASVspoof dataset in two parts. Even with the large variations in replay configurations present in that dataset, Void still performed well for built-in speakers, achieving 1.5% EER. The EER for standalone speakers was 5.1% due to larger FAR and FRR results.

7.2.3 Time taken for training and attack detection. Table 3 summarizes the time taken for training, feature extraction, and testing (classifying) a single voice sample from the ASVspoof dataset. We compare Void's time measurements against a constant Q cepstral coefficients (CQCC)-based approach, which is used as a baseline approach in 2017 ASVspoof challenge [7]. Feature extraction time ('Extraction' in Table 3) measures the time taken to extract all features from a voice sample, and testing time measures the time taken to finish classifying a given voice sample.

Table 3: Space used for training ASVspoof dataset [7], and average training and testing time (Numbers in parentheses indicate standard deviations).

Complexity	Feature	Void	CQCC [7]
Time (sec.)	Extraction	274.16 (0.24)	577.94 (0.31)
	Training	3.5 (0.013)	929.97 (0.34)
	Testing	0.18 (0.025)	0.63 (0.026)
Space	# Features	167	25,560
	# Training data	3,008,672	139,121,550
Accuracy	EER	5.1%	23.8%

Compared to CQCC, Void is significantly faster in all aspects. Void also uses significantly less space with respect to feature vector size and training data size. Those observations clealry indicate that Void is a highly efficient and fast solution, making it more competitive to be deployed on existing voice assistant servers.

7.3 Effects of variances

In this section, we analyze the effects of three key variances – distances between target device and attack device, gender, and cross data training – on the performance of Void. We trained the

Void model with 16,000 voice samples from our dataset: 8,000 live-human samples and 8,000 replayed samples. We randomly selected 8,000 replay attack samples from a total 219,872 attack samples. All attack samples were replayed through the standalone speakers listed in Appendix A.

Table 4: Performance of Void under diversity.

Diversity	Dimension	Samples	Detection	Accuracy
Distance	15cm	3,996	3,963	99.1%
	130cm	3,995	3,979	99.5%
	260cm	3,997	3,971	99.3%
Gender	Male	7,995	7,945	99.3%
	Female	7,996	7,959	99.5%
Cross data	Scenario 1	4,000	3,436	85.9%
	Scenario 2	5,996	4,802	80.1%
	Scenario 3	7,994	6,573	82.2%

7.3.1 Sound source distances. To analyze the effects of varying distances between attacker's devices and target device, voice samples were replayed using three different distances: 15cm, 130cm, and 260cm. We replayed a total of 3,996 voice samples – each loudspeaker was used to replay around 1,000 voice samples. Accuracy results are presented in Table 4. Regardless of varying distances, Void achieved over 99% accuracy in detecting replay attacks – demonstrating that distance variations do not really affect the performance of Void. Note, we did not try distances that are too far away from target devices (e.g., 10 meters) since attackers would then have to use very loud volumes, which would be noticed by people nearby.

7.3.2 Gender. Since the fundamental frequency characteristics of male and female voices are quite different (e.g., females have typically higher fundamental frequencies than males) [20, 21], the power distribution patterns may vary between males and females. To analyze effects of changing gender, we randomly selected 7,995 and 7,996 replayed voice samples from 30 male and 30 female participants, respectively. Then we measured Void's performance on each gender group. Again, the accuracy results shown in Table 4 indicate that gender variances did not really influence our detection accuracy.

7.3.3 Cross data training. For cross data training, we trained Void on one set of participants and playback devices, and evaluated the performance of Void using another set of participants and playback devices.

For the training dataset, we used voice samples replayed through specifically the Bose speaker from 10 male participants. For testing, we considered the following three scenarios: In scenario 1, we launched replay attacks with samples collected from 10 female participants whose voices were played back using a standalone Logitech speaker. In scenario 2, we launched attacks with voice samples collected from 20 female participants whose voices were played back using two standalone speakers, VMODA and Logitech. In scenario 3, we launched attacks with voice samples collected from 20 female and 10 male participants whose voices were played back using three standalone speakers, Yamaha, VMODA, and Logitech. Again, the performance of Void on those samples is presented in Table 4. In all three scenarios, the detection accuracy decreased substantially to about 80–86%. Similar to other machine learning based approaches, the performance of Void could be significantly downgraded when we use a training dataset consisting of small number of loudspeakers. This limitation is discussed further in Section 8.2.

7.4 Detecting hidden and inaudible voice commands

We also experimented with hidden and inaudible voice command attacks. This section presents the performance of Void against those two attacks.

7.4.1 Performance against hidden voice commands. Hidden voice commands refer to commands that can not be interpreted by human ears but are understood and processed by target devices (see Section 2). Void should perform well against such hidden commands as it relies on spectral power features that are difficult to manipulate. Moreover, hidden voice commands add more frequencies to an original voice sample during obfuscation, which would increase the overall signal power linearity (see Section 5.2.2). We recorded hidden voice command samples using the black-box attack methods demonstrated in [25], and with 1,250 samples from our own dataset, and 13,306 samples from the ASVspoof dataset (see Table 5).

Table 5: Performance of Void against adversarial voice attacks.

Attacks	Dataset	Samples	Accuracy (%)
Hidden	Our dataset	1,250	99.6
	ASVspoof 2017	13,306	96.05
Inaudible	ultrasonic speaker	5,000	93.3

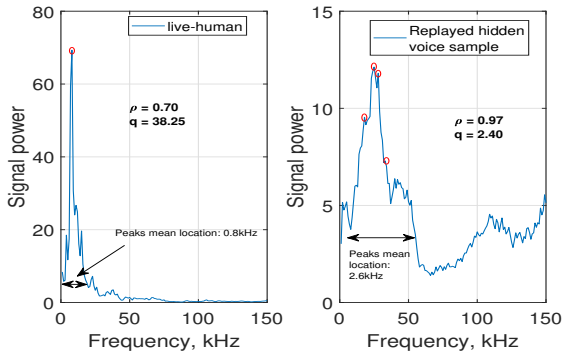

Figure 10: Power spectrum and spectral features of voice sample *Artificial intelligence is for real.* for live-human (left) and hidden voice sample (right) over the range of frequencies.

Figure 10 compares signal power distributions for live-human voice and hidden voice command crafted with a phrase "Artificial

intelligence is for real." The original command is shown on the left, and the obfuscated hidden command is shown on the right. This obfuscated command was replayed through a loudspeaker. From the signal power distribution shown on the right, we can clearly observe signal power linearity scores that indicate a replay attack. Unlike the live-human case in which the power distribution over the range of frequencies shows non-linear behavior (mostly concentrated below 2 kHz), the linearity coefficients for the hidden voice samples show linear behavior (i.e., ρ: 0.97 and q: 2.40). The high power frequencies location is also different, which is another clue for detecting a replay attack.

The hidden voice command attack detection accuracy measured with our own dataset and ASVspoof dataset were 99.6% and 96.05%, respectively (see Table 5).

7.4.2 Performance against inaudible voice commands. Inaudible voice command attack involves playing an ultrasound signal with spectrum above 20kHz, which would then be inaudible to human ears. Inaudible voice commands are played through ultrasonic speakers. Due to the non-linear behavior of hardware – microphones in this case – the received voice signals are shifted to lower frequencies (down-modulation) with much lower power. Figure 11 compares the signal power over the audible frequency range for live-human (left) and inaudible voice sample (right) replayed through Jameco Valuepro 40TR12B-R ultrasonic speakers, which were also used in [16]. To evaluate the performance of Void against inaudible voice attacks, we modulated voice commands on higher frequencies; i.e., each command was modulated using amplitude modulation (AM) between 23kHz and 34kHz with a 1kHz gap. Once modulated, voice signals were transmitted using Jameco Valuepro 40TR12B-R Ultrasonic Sensor Set. We used 5,000 samples from the ASVspoof dataset for evaluation.

From the two graphs, we can clearly see that due to the down-modulation effects from the non-linearity characteristics of the hardware (microphone) used, the sum of power in inaudible sample signal is much lower than that of the live-human sample. Also, the linearity in power distribution and the high power frequencies are obvious indicators of a replay attack.

As summarized in Table 5, Void achieves 93.3% detection accuracy for inaudible voice attacks.

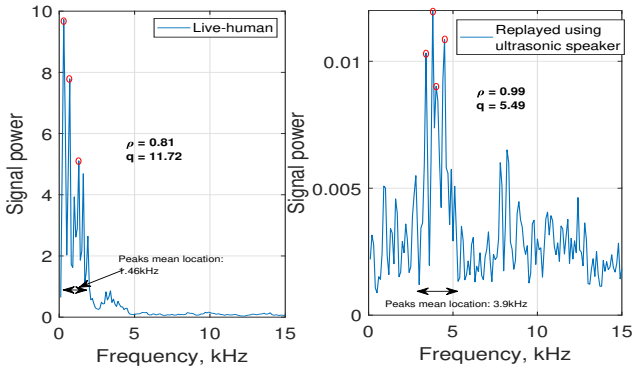

Figure 11: Power spectrum and spectral features of a voice sample for live-human (left) and inaudible voice sample (right) over the range of frequencies.

8 DISCUSSION

8.1 Effectiveness of Void

Our evaluation results show that Void can effectively detect voice replay attacks (see Section 7.2). For two different datasets consisting of 229,991 and 18,016 voice samples, Void achieved detection accuracy of over 99% and 98% with less than 1% and 5.1% EER, respectively. Void is also resilient against adversarial attacks with hidden voice commands and inaudible voice commands, achieving 96% and 93% accuracy for the two types of attacks, respectively (see Section 7.4).

Because Void exploits two different characteristics in spectral power patterns of voice signals, which can vary significantly based on the type of loudspeakers, the performance of Void can also be affected by loudspeaker types. Overall, Void is more effective in detecting replay attacks launched through built-in speakers in portable devices. For example, in the ASVspoof dataset, Void is capable of achieving 1.5% EER for built-in speakers while the EER for standalone speakers increases to 5.1% (see Section 7.2).

Under various controlled conditions about distance and speaker's gender, Void still produced reliable performance results. For example, when a replay attack is launched with varying distances between the victim device and the attack device, the overall distribution of spectral power over frequencies is not significantly effected. Hence, Void achieved accuracy of more than 99% with distance (see Section 7.3). This is because Void exploits the relative changes of spectral power pattern rather than the absolute values of power density. However, as we can see the experiments results for cross data training, the performance of Void could be significantly degraded when using a small training dataset. We surmise that unexpected patterns of voice signals generated by various high-quality standalone speakers lead to a reduction in the effectiveness of Void.

Compared with the existing machine learning based approaches (e.g., CQCC), Void is lightweight and very fast in detecting replay attacks. In our implementation, the number of features used in Void is 167 while the number of features used in CQCC is 25,560. Moreover, Void took only 3.5 seconds for training time and 0.18 seconds for testing time (on average) while CQCC took 929.97 seconds for training time and 0.63 seconds for testing time (see Section 7.2). Therefore, Void might be considered to be incorporated into a voice assistant app in smartphones because of its relatively lower computational overhead. In this case, we can significantly reduce the workload on the server because the voice assistant app can locally discard unwanted voice commands (e.g., suspicious voice commands) before delivering them to the server.

8.2 Limitations of Void

Since Void is based on just spectral power analysis to detect replay attacks and the liveness of genuine users, some intentional perturbation in a particular frequency range may affect the performance of Void. For instance, when a recorded command is played through a built-in speaker in a portable device, and at the same time, a human attacker can also start uttering some random phrases or words.

In this case, the characteristics of built-in speaker can be compensated by the human attacker's voice, which may affect the accuracy of Void. Even though a replayed voice command includes a linear power density pattern in the frequency range of 100Hz-8kHz, the human attacker's speech may generate a non-linear power density value such that most power density lies in higher frequencies below 1kHz, which makes it more difficult to detect a difference between the device owner's live voices and replayed voices. To mitigate such sophisticated voice replay attacks with a human attacker's live-voice speech, another defense mechanism (e.g., speaker-recognition algorithm) can be additionally introduced. Note that developing such a defense mechanism is beyond the scope of this paper.

Unsurprisingly, most machine learning based approaches including Void cannot easily detect new and unexpected samples which were not included in the training dataset. Therefore, the performance of Void may be degraded in real environments exposed to combinations of numerous types of recording and plackback devices for voice replay attacks. To avoid the performance degradation of Void, we will consider two strategies commonly applied in machine learning techniques: (1) We can increase the size of training dataset. Surely, this idea is not new, but we claim that Void has the advantage of learning new training samples compared with the existing methods because the training cost of Void is much lower than other classification methods. (2) The performance of Void against new devices might be improved with deviation-based outlier detection methods by building a model of live-human voice samples and detecting deviations from the normal model in the observed data. The key features used in Void can still be used to develop effective deviation-based outlier detection methods.

9 RELATED WORK

Recent studies have demonstrated that voice assistants on smartphones are insecure and prone to various forms of voice presentation attacks [6, 8, 9, 18, 23] – such attacks "present human characteristics to the biometric capture subsystem in a fashion that could interfere with the intended policy of the biometric system" [4]. In practice, however, *replay attack* is the most serious security threat to voice-as-biometric authentication systems because this attack can simply be implemented by recording a victim's voice command and playing it later.

Most sophisticated attacks were also introduced to hide the attack attempts themselves. Carlini et al. [24, 25] presented an attack called hidden voice commands to generate *mangled* voice commands that are unintelligible to human listeners but which are interpreted as commands by devices. Zhang et al. [16] extended this attack to make voice commands completely inaudible by modulating voice commands on ultrasonic carriers (e.g., over 20 kHz) to achieve inaudibility.

Inaudible voice attacks [16, 17] are improving over time as potential attack on voice assistants by exploiting loophole in hardware non-linearity. To overcome the limitation of short distances (within about *5ft*) in previous studies [16, 17], Roy et al. [18] demonstrated the feasibility of launching such attacks from longer distances (i.e., within *25ft* range) by multiple ultrasonic speakers. They stripe segments of the voice signal across multiple speakers placed in separated space. Moreover, they developed a defense system against to detect inaudible voice command attacks by analyzing the properties of inaudible voice samples.

There are many different approaches to detect machine-generated voice attacks. Zhang et al. [5] proposed user's articulatory gesture-based liveness detection (looking at precise movement of articulators like lips and tongue) but their approach is only applicable to scenarios where a speaker's mouth is physically near a smartphone's microphone. Similarly, Chen et al. [9] leveraged magnetic fields emitted from loudspeakers to detect replay attacks. Their approach, however, requires users to utter a passphrase while moving smartphones through a predefined trajectory around sound sources. Feng et al. [8] proposed a voice authentication system that uses a wearable device, such as eyeglasses, earphones/buds, and necklaces – collecting a user's body surface vibrations, and matching it with voice signals received by a voice assistant through a microphone. Although their approach is capable of achieving around 97% accuracy, they rely on an additional hardware user always has to carry on. Therefore, detecting replay attacks remain challenging without any extra hardware support.

Recently, an extensive study (through the 2017 ASVspoof challenge [7]) analyzed the performance of machine learning-based state-of-the-art methods for distinguishing human voice samples from sound samples replayed through a loudspeaker. According to their experimental results, the equal error rates (EER) of the methods were varied from 6.73% to 45.55%. The winning solution fused four models including three deep learning models (two CNNs and one RNN) and one SVM model [28]. However, combining multiple models and/or using the deep learning approach generally requires heavy computational resources.

This motivated us to develop a high-performance model without requiring heavy computational resources. Void was designed with a small number of key features to detect machine-generated voice impersonation attack attempts.

10 CONCLUSION

Void analyzes the spectral power patterns of voice signals to accurately detect voice replay attacks. Compared with existing methods that make heavy use of data-driven techniques and classification features, our solution runs on a lightweight and efficient classification algorithm with a small number of features (167), and does not require any additional hardware.

Our experiments, conducted on several (and diverse) voice datasets, showed that Void can achieve detection accuracy of about 99% (with less than 1% EER) for replay attacks launched through portable devices with built-in speakers and high-quality standalone loudspeakers. Moreover, Void is resilient to hidden [24, 25] and inaudiable voice command attacks [16–18], respectively. Void achieves an accuracy of 96% in detecting hidden voice commands, and an accuracy of 93.3% in detecting inaudible voice commands.

As part of future work, we plan to deploy Void in a practical real-world environment and further investigate the performance of Void with a massive set of voice commands.

REFERENCES

[1] Google Smart Lock, https://get.google.com/smartlock/, 2017.
[2] Lenovo voice unlock, [Online:] *https://www.techinasia.com/baidu-lenovo-voice-recognition-android-unlock*, 2012.

[3] Wechat Voiceprint, [Online:] *http://blog.wechat.com/2015/05/21/voiceprint-the-new-wechat-password/*, 2015.

[4] ISO/IEC 30107-1:2016, "Information technology – Biometric presentation attack detection," in *ISO/IEC Information Technology Task Force (ITTF)*, 2016.

[5] L. Zhang, S. Tan, J. Yang, "Hearing Your Voice is Not Enough: An Articulatory Gesture Based Liveness Detection for Voice Authentication", in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[6] A. Janiki, F. Alegre, and N. Evans, "An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks", in *Security and Communication Networks*, pp. 3030-3044, 2016.

[7] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection", in *Proceedings of the 18th INTERSPEECH*, 2017.

[8] H. Feng, K. Fawaz, and K. G. Shin, "Continuous Authentication for Vicoe Assistants", in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017.

[9] S. Chenyz, K. Reny, S. Piaoy, C. Wang, Q. Wangx, J. Weng, L. Suy, and A. Mohaisen, "You Can Hear But You Cannot Steal: Defending against Voice Impersonation Attacks on Smartphones", in *Proceedings of IEEE 37th International Conference on Distributed Computing Systems*, 2017.

[10] A. Liptak, "Amazon's Alexa started ordering people dollhouses after hearing its name on TV", [Online:] *https://www.theverge.com/2017/1/7/14200210/amazon-alexa-tech-news-anchor-order-dollhouse*, 2017.

[11] D. Luo, H. Wu, and J. Huang, "Audio recapture detection using deep learning", in *Proceedings of the 3rd IEEE China Summit and International Conference on Signal and Information Processing*, 2015.

[12] P. Dicomo, "Understanding Speaker Frequency Response", [Online:] *http://www.ecoustics.com/articles/understanding-speaker-frequency-response/*

[13] Sound Engineering Academy, "Human Voice Frequency Range", [Online:] *http://www.seaindia.in/blog/human-voice-frequency-range/*

[14] "BOSARIS Toolkit," [Online:] *https://sites.google.com/site/bosaristoolkit/*

[15] J. Gago, J. Balcells, D. GonzÁlez, M. Lamich, J. Mon, and A. Santolaria, "EMI susceptibility model of signal conditioning circuits based on operational amplifiers", in *IEEE Transactions on Electromagnetic Compatibility*, vol. 49, no. 4, pp. 849-859, 2007.

[16] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, W. Xu, "DolphinAtack: Inaudible Voice Commands", in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[17] Song, Liwei and Mittal, Prateek, "POSTER: Inaudible Voice Commands", in *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[18] N. Roy, S. Shen, H. Hassanieh, R. R. Choudhury, "Inaudible Voice Commands: The Long-Range Attack and Defense", in *Proceedings of 15th USENIX Symposium on Networked Systems Design and Implementation*, 2018.

[19] G. Militano, "Understanding Speaker Specifications and Frequency Response", [Online:] *http://novo.press/understanding-speaker-specifications-and-frequency-response*, 2011.

[20] I. R. Titze, "Principles of Voice Production", in *Prentice Hall*, 1994.

[21] R. J. Baken, "Clinical Measurement of Speech and Voice", in *Taylor & Francis*, 2000.

[22] Bose Corporation, "SoundLink: Mini Bluetooth, speaker II", [Online:] *https://www.bose.com/en_us/products/speakers/portable_speakers/soundlink-mini-ii-bundle.html*.

[23] Saurabh Panjwani and Achintya Prakash, "Crowdsourcing Attacks on Biometric Systems", in *Proceedings of the 10th Symposium On Usable Privacy and Security*, 2014.

[24] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition", in *Proceedings of the 9th USENIX Workshop on Offensive Technologies*, 2015.

[25] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden Voice Command", in *Proceedings of the 25th USENIX Security Symposium*, 2016.

[26] P. Castiglioni, "Levinson-durbin algorithm", in in *Encyclopedia of Biostatistics*, 2005.

[27] ASVspoof, [Online:] *https://datashare.is.ed.ac.uk/handle/10283/2778*

[28] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, V. Shchemelinin, "Audio replay attack detection with deep learning frameworks", in *Proceedings of the 18th INTERSPEECH*, 2017.

A EXAMPLE OF CUMULATIVE DISTRIBUTION OF SPECTRAL POWER DENSITY

An example of the cumulative distribution of spectral power density is represented in Figure 12. In this example, we can see that around 70% of the overall power lies in the frequency range below 1kHz in the live-human voice. However, in the loudspeaker case, the cumulative distribution of spectral power density is almost linearly increasing, and 70% of the total power lies at the frequency range of about 4kHz. Thus, we can clearly see that there exist unique patterns in cumulative distribution of spectral power density to distinguish live-human voices from voices replayed through built-in speakers on smartphones.

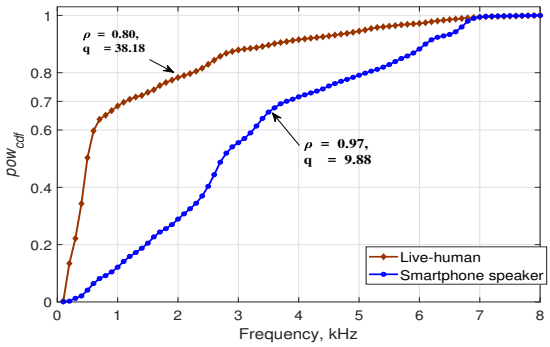

Figure 12: Cumulative distribution of spectral power density over frequencies where the total frequencies in audio signal are up to 8kHz and $W = 10$.

B FEATURE VECTORS AND DESCRIPTION

There are different characteristics in signal power distribution between low-quality speakers (such as built-in speakers in portable devices) and high-quality standalone speakers. To detect replay attacks from any kind of speakers, we choose to use signal power linearity degree features for low-quality speakers (see Table 6) and higher power frequencies features for high-quality speakers (see Table 7).

Table 6: Summary of signal power linearity degree features. The measure of linearity of the power plays an important role to detect live-human and loudspeaker; since in case of loudspeaker, the power over frequency range is highly linear where in live-human case, it is highly skewed.

Signal's Power Linearity Scores	Symbol
Cross-correlation coefficients	ρ
Quadratic curve-fitting coefficients	q
$FV_{LDF} = \{\rho, q\}$	

Table 7: Signal's High Power Frequencies Features Vector.

Signal's High Power Frequencies Features	Symbol
#peaks in high-power frequencies	N_{peaks}
Relative frequencies corresponding to *peaks*	μ_{peaks}
Standard deviation of high power frequency location	σ_{peaks}
$FV_{HPF} = \{N_{peaks}, \mu_{peaks}, \sigma_{peaks}\}$	

C DEVICE INFORMATION AND THE LIST OF VOICE COMMANDS IN THE EXPERIMENT

To validate our system, we used various devices. For attack, we played recorded voice commands through built-in speakers in 14 different portable devices including 10 smartphones and 4 laptops, and 4 different high-quality standalone speakers, respectively (see Table 8). For recording devices, we used 3 different laptops and 2 different smartphones (see Table 9).

Table 8: List devices used for replay attack.

	Maker	Model
Smartphone	Galaxy A8	A810S
	Galaxy A5	SM-A500x
	Galaxy Note 8	SM-N950x
	Galaxy S8	SM-G950
	Galaxy S8	SM-G955N
	Galaxy S9	SM-G960N
	iPhone SE	A1662
	iPhone 6S Plus	A1524
	iPhone 5S	A1519
	LG V20	V20 F800
Standalone	Bose (Bluetooth)	SoundTouch 10
	V-MODA (Bluetooth)	REMIX-BLACK
	Logitech (2.1 Ch.)	Z623
	Yamaha (5.1 Ch.)	YHT-3920UBL
	Ultrasonic speakers	Valuepro 40TR12B-R

Table 9: List of devices used for recording voice commands.

Maker	Model
Samsung Notebook	NT910S3T-K81S
Samsung Notebook	NT200B5C
Macbook Pro	A1706 (EMC 3163)
Galaxy S8	SM-G955N
Galaxy S9	SM-G960N