EarSpy: Spying Caller Speech and Identity through Tiny Vibrations of Smartphone Ear Speakers

Ahmed Tanvir Mahdad
Texas A&M University
mahdad@tamu.edu

Cong Shi
New Jersey Institute of Technology
cs638@njit.edu

Zhengkun Ye
Temple University
zhengkun.ye@temple.edu

Tianming Zhao
University of Dayton
tzhao1@udayton.edu

Yan Wang
Temple University
y.wang@temple.edu

Yingying Chen
Rutgers University
yingche@scarletmail.rutgers.edu

Nitesh Saxena
Texas A&M Unviersity
nsaxena@tamu.edu

ABSTRACT

Eavesdropping from the user's smartphone is a well-known threat to the user's safety and privacy. Existing studies show that loudspeaker reverberation can inject speech into motion sensor readings, leading to speech eavesdropping. While more devastating attacks on ear speakers, which produce much smaller scale vibrations, were believed impossible to eavesdrop with zero-permission motion sensors. In this work, we revisit this important line of reach. We explore recent trends in smartphone manufacturers that include extra/powerful speakers in place of small ear speakers, and demonstrate the feasibility of using motion sensors to capture such tiny speech vibrations. We investigate the impacts of these new ear speakers on built-in motion sensors and examine the potential to elicit private speech information from the minute vibrations. Our designed system *EarSpy* can successfully detect word regions, time, and frequency domain features and generate a spectrogram for each word region. We train and test the extracted data using classical machine learning algorithms and convolutional neural networks. We found up to 98.66% accuracy in gender detection, 92.6% detection in speaker detection, and 56.42% detection in digit detection (which is 5X more significant than the random selection (10%)). Our result unveils the potential threat of eavesdropping on phone conversations from ear speakers using motion sensors.

KEYWORDS

Eavesdropping, ear, Motion Sensor

ACM Reference Format:
Ahmed Tanvir Mahdad, Cong Shi, Zhengkun Ye, Tianming Zhao, Yan Wang, Yingying Chen, and Nitesh Saxena. 2022. EarSpy: Spying Caller Speech and Identity through Tiny Vibrations of Smartphone Ear Speakers. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Eavesdropping on smartphones is always a well-known threat and a major security concern for users. Call recording is the most straightforward approach for an adversary to eavesdrop. However, smartphone operating systems are imposing restrictions on third-party apps for recording calls using microphones [3, 12], which thwarts most attacks relying on microphone access.

A possible workaround for adversaries can be extracting speech information from zero-permission motion sensors through a side-channel attack. It is a significant privacy concern that users are unaware of [7] but have been extensively investigated by researchers in the last decade. Researchers have reported potential eavesdropping prospects using motion sensors [1, 4, 13], keystrokes on touchscreens [35], stylus pen writing [18] and using external devices [5, 28]. Furthermore, eavesdropping through light sensors [6], gyroscope [19] are also reported in the literature.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

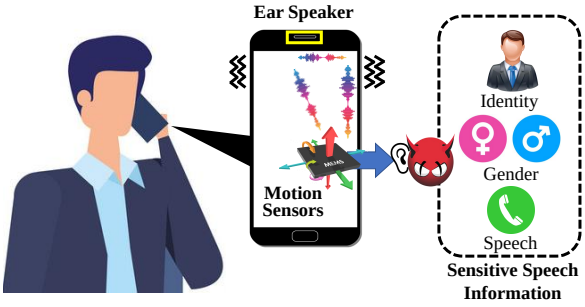


Figure 1: Overview of ear speaker eavesdropping.

Among the built-in sensors of smartphones, motion sensors are mostly known as vulnerable to eavesdropping. Adversaries leverage motion sensors to collect audio (e.g., voice conversation [4]), touch screen inputs [34], and even indoor locations [37]. Eavesdropping through motion sensors is straightforward, as adversaries do not need explicit permission to collect raw data from them.

An abundant amount of work has been done on eavesdropping attacks induced by vibration generated from phone loudspeakers (e.g., [1, 4]). However, very few works have been done on eavesdropping *ear speakers*, a built-in internal speaker in a smartphone that is used to listen to the conversation while the phone is held to the ear. Eavesdropping on the ear speaker is the most practical attack vector that can eavesdrop on phone conversations, as most people are not willing to expose sensitive speech, especially in public places. A few recent studies [5, 28] show that the vibrations produced by ear speakers can be captured using high-resolution wireless sensors placed close to the victim.

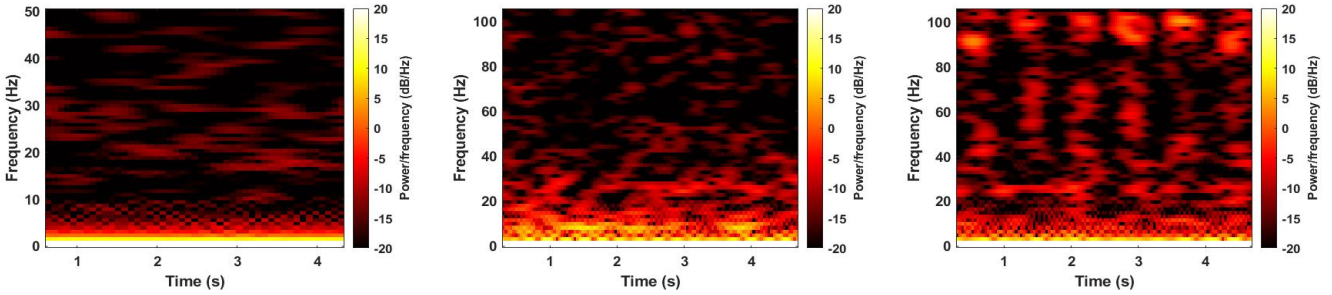
A natural question is that: Is it possible to eavesdrop on ear speakers using built-in motion sensors? Such an attack setting is highly practical due to the zero-permission property of motion sensors, which does not require placing or hacking any devices in the victim's environment. Previous studies did not find enough impact of ear speakers on the accelerometer (e.g., Figure 10 of [1]). However, we find that the audio quality of smartphone speakers continues to improve and evolve [10]. Following the trend, recent flagship smartphones contain stereo speakers, which requires placing two speakers at the top and bottom. In most cases, traditional small ear speakers are replaced by more prominent stereo speakers. As a result, phones with stereo speakers produce more sound pressure during conversations than phones with conventional ear speakers. In Figure 2, a comparative spectrogram analysis of two smartphones (Oneplus 7T contains stereo speakers, whereas OnePlus 3T does not) presents a noticeable difference in its vibration effect in a motion sensor (i.e., accelerometer) while playing a recording of the word "Zero" six times in five seconds interval. Figure 2a shows a spectrogram demonstrating the very low impact of ear speakers in the accelerometer in an older model phone (OnePlus 3T) where stereo speakers were not present. Figure 2b shows some impact on the accelerometer due to vibration induced by ear speakers of a newer model smartphone (i.e., OnePlus 7T) compared to Figure 2a. Figure 2c shows the spectrogram for loudspeakers of the OnePlus 7T with a clear view of word regions.

Based on these observations, we propose to analyze the accelerometer data and try to extract sensitive speech and speaker information from a speech played on ear audio. Although phone manufacturers use a larger and more powerful speaker at the top in place of ear speakers, during a phone conversation, the volume is controlled at a level so that users do not experience any discomfort. We use public speech datasets (e.g., Free Speech Data Set [14], JL-Corpus [17], emo-DB [11]) in our experiment, and our word region detection program can still detect more than 50% of "word region" from the raw accelerometer data. We extract time and frequency domain features, generate spectrograms, and use classical machine learning algorithms and deep learning techniques to examine if they can detect speech information (e.g., words, speakers, gender) from accelerometer data. Our analysis reveals that an adversary can successfully reveal Gender, Speaker, and speech information with reasonable accuracy (98% for Gender detection, 92% for speaker detection). An overview of the system is illustrated in Figure 1.

Our Contribution: We analyze the effect of vibration induced by ear speakers during a conversation, extract time-frequency domain features, and generate a spectrogram for each identified word region. We use classical machine learning and deep learning techniques to identify the speech, speaker, and gender of the caller by analyzing the features. Our contribution to this work is three-fold:

(1) ***Exploration of Eavesdropping Opportunity on Ear Speaker using Built-in Motion Sensors***: Eavesdropping from ear speakers is one of the most real-world and practical threats. Researchers have already explored this area [5, 28] in the previous literature that uses external radars/devices outside of the smartphone device. However, to the best of our knowledge, *EarSpy* is the first work that explores the eavesdropping opportunity on ear speakers using built-in motion sensors of recent smartphones with stereo speakers.

(2) ***Extraction of Word Regions and Features from Accelerometer Data***: Although we observe a little impact of ear speaker-induced vibration in accelerometer data, we are able to identify more than 45% word regions to analyze. We also extract time and frequency domain



(a) Spectrogram generated from accelerometer data of Oneplus 3T ear speaker (older model, no stereo speakers).

(b) Spectrogram generated from accelerometer data of Oneplus 7T ear speaker (newer model, with stereo speakers).

(c) Spectrogram generated from accelerometer data of Oneplus 7T loud speaker (newer model, with stereo speakers).

Figure 2: Spectrogram generated while playing word "Zero" for six times.

features and generate a spectrogram for each identified word region. We use these features and spectrograms to feed into classical machine learning and deep learning techniques to further analyze the accuracy of speech, speaker, and gender detection.

(3) ***Achieved Reasonable Accuracy in Detecting Speech and Speaker Specific Information***: We achieve high accuracy in detecting speech (56% accuracy) and speaker information (i.e., speaker (92% accuracy) and gender identification (98% accuracy)). Compared to audio and vibration domain (using loudspeaker) performance, this result is promising and reveals the real-world threat of voice conversation eavesdropping.

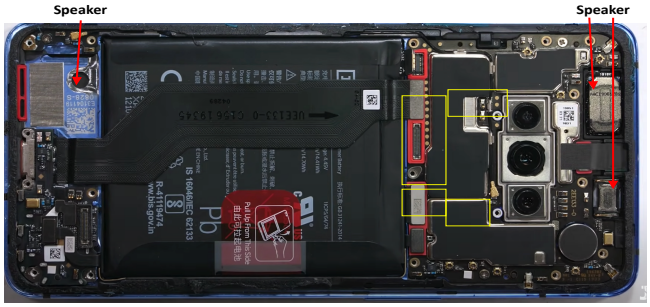


Figure 3: Teardown snapshot [16] of OnePlus 7T.

2 BACKGROUND

Ear Speakers on Smartphone. Ear speakers are designed to produce low-volume sound during phone conversation where the user places the phone against his ear in order to clearly hear the sound from the ear speakers. Fig 3 depicts the layouts of ear speakers on a typical phone model (i.e., Oneplus 7T). Specifically, the speaker at the bottom is typically the loudspeaker. The ear speakers of a smartphone is mounted on the top area of the smartphone's motherboard. Since the vibrations generated by the ear speaker are much weaker than the loudspeaker, therefore, a direct contact between the ear speaker and the user's ear is ideal for high-quality sound reception. The main reason why it helps is that sounds propagating among two solid surfaces (i.e., ear speaker and ears) are much better than no physical contact case (i.e., air as the intermediate medium).

Vibration Captured by Motion Sensor on Smartphone. Modern smartphones are equipped with highly sensitive motion sensors (i.e., accelerometer and gyroscope) that are designed for sensing phone vibrations. Existing studies [2, 26] have shown that the vibration of the phone body caused by the transmitted sound from the built-in speaker can be captured by the motion sensor. The basic principle is that the sound transmitted through the smartphone's body generates vibrations, and the motion sensor on that smartphone can capture those vibrations. More specifically, Spearphone [2] found that the accelerometer on the smartphone has a strong response to the sound frequency from 100Hz to 3300Hz. Moreover, they observed that sounds at different frequencies generate responses at the low-frequency points of the accelerometer, known as aliased signals. And it can be expressed using the equation as follows: $f_a = |f - N \cdot f_s|$, where f_a is the vibration frequency of the accelerometer, f is the sound frequency, f_s is the accelerometer sampling rate, and N can be any integer. This effect shows that the accelerometer can capture rich information in low-frequency aliasing signals since they are derived from the original sound at different frequencies. In addition, they also compared the frequency response of both accelerometers and gyroscopes and found that the accelerometer's response was stronger than the gyroscope's response in the frequency range 100Hz to 3300Hz. Therefore, we only adopt accelerometers in our experiments as well.

3 RELATED WORK

Direct speech sensing on external loudspeakers/human throats. Many research investigations have extended the search for speech eavesdropping from using a tampered/hidden microphone to radio frequency (RF) sensors, such as WiFi [29, 31], Ultra-Wideband [30], and mmWave signals [33]. For example, researchers have explored using WiFi signals to recover the sound of speaker devices [31] and capture mouth motions [29] for speech eavesdropping. These attacks rely on external and potentially customized/dedicated sensing devices around the human subjects for sensing, rendering these attacks cumbersome and less stealthy.

Indirect speech sensing based on vibrations. Speech eavesdropping is also shown feasible through sensing sound-induced vibrations using various types of sensors [8, 19, 20, 27, 36]. Gyrophone [19] first showcased the attack setup where a smartphone is placed on the same solid surface as a loudspeaker. The smartphone's gyroscope is then used to capture the surface vibrations induced by the speech playbacks of the loudspeaker. Recent studies further demonstrate the feasibility of such attacks through vibration sensing using lasers, high-speed cameras, and light sensors. For example, Davis et al. [8] utilizes a high-speed camera to capture video streams to recover vibrations from some room objects (e.g., a bag of chips). Nassi et al. [21] show that sound vibrations on lamps can be detected and recovered by using electro-optical sensors. These attacks are promising, but they require a loud sound volume of the loudspeaker (e.g., 70~110dB) or a close distance between the vibration surface and the loudspeaker to trigger surface vibrations. Differently, this work targets more realistic attack scenarios, where the sounds with a low sound volume of around 50dB are produced by the ear speaker of smartphones. Compared to these existing approaches, our attack is also more resilient to impacts of environments, such as the occlusion by walls and movements of nearby human subjects.

Speech eavesdropping on smartphone speaker. Instead of using external sensors, Spearphone [1] and AccelEve [4] recently demonstrated new eavesdropping attacks which derive speech based on the motion sensors on the same smartphone. The vibrations produced by the speaker can propagate through the motherboard and reach the motion sensors. With the motherboard as the vibration medium, it is more stable for the motion sensors to pick up speech vibrations. AccEar [13] takes one step forward to design a deep neural network to reconstruct audio signals from the motion sensor readings. The attacks show promising results, but they assume the sounds are relayed by built-in loudspeaker, which is audible to nearby people and is less likely to use for phone calls in public spaces (e.g., offices, conferences). Different from these prior works, our attack targets minute speech playback by an ear speakers, which is completely inaudible to nearby human subjects. Our attack is more devastating as users normally believe the confidentiality of the speeches played via ear speakers (e.g., one-time passwords, birthdays, and social security card numbers) has been enforced.

4 DESIGN AND IMPLEMENTATION

We design a system that uses motion sensor data from the user's smartphone induced by ear speaker vibrations. Our goal is to examine if the ear speakers cause distinguishable vibration patterns on motion sensor data. This section discusses the system design and tools used for the experiment in detail.

4.1 Feasibility Determination

Playing Voice Through Ear Speakers: We have used some third-party Android apps (Ear [23], Mobile Ear Speaker Earphone [25]) to play audio only through the ear speakers. We have used available, and widely known voice data sets (e.g., JL-Corpus [17]), FSDD [14], Emo-DB [11]) and played the audio through ear speakers.

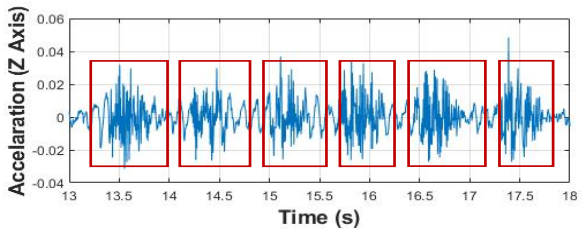
Impact of Powerful Ear Speakers of Smartphone: Ear speakers are designed to produce low-volume sound during a phone conversation. The volume level is set to an optimum level so that it can be comfortable to hear during a conversation in the handheld position. Smartphone speakers are evolving fast in the last decade [10], and the recent trend is to introduce stereo speakers with smartphones. Generally, stereo speakers are two speakers built into the top and bottom portions of phones. As such, manufacturers are designing phones with better quality speakers at the top, which is also used as ear speaker during phone conversations.

Although some phone manufacturers claim that their phone has stereo speakers, their designed top speakers are not as powerful as the bottom primary loudspeaker. We have analyzed some publicly available teardown videos [15, 16] and noticed some phone manufacturers use larger multiple speakers at the top to boost the audio quality (e.g., OnePlus 7T teardown at Figure 3) whereas others use smaller speakers (e.g., Google Pixel 5 [15]). Smartphone with larger and multiple speakers is more likely to generate more vibration than smaller ones.

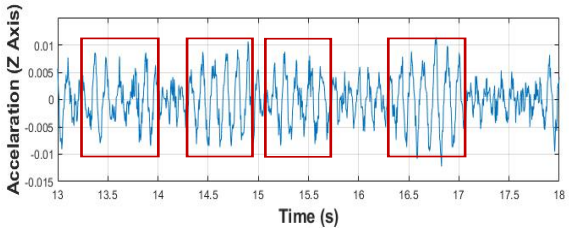
To test this hypothesis, we play a recorded word "Zero" six times in five second interval through ear speakers of a smartphone where large dual speakers are used (e.g., OnePlus 7T) and collect accelerometer readings. We extract the word region and generate a spectrogram for it. We did the same experiment with another phone with less powerful speakers (OnePlus 3T). From the generated spectrogram (Figure 2a,

and Figure 2b), it is evident that audio played with a larger and improved speaker will cause distinguishable vibration, unlike previous phones.

Choosing Accelerometer to Collect Vibration Data: We have already learned from previous literature [1] that the accelerometer performs better than a gyroscope to capture the vibration from the smartphone's internal speakers. So, in our experiments, we primarily focused on capturing accelerometer data.



(a) Example of identified word regions from acceleromter data after applying 8Hz high-pass filter on loudspeaker.



(b) Example of identified word regions from accelerometer data after applying 8Hz high-pass filter on ear speaker.

Figure 4: Comparison of identified word regions from the loudspeaker and ear speaker setting.

4.2 Word Region Identification

As we discussed earlier, we play speech audio using publicly available speech datasets and collect the vibration data from the accelerometer. We place the phone in the handheld position (i.e., the natural posture of a human during a phone conversation). As a result, body and hand movements add low-frequency noises to the accelerometer data. We place a high-pass filter during our analyses to eliminate the effect of low-frequency body and hand movement.

Ear audio creates a very small impact on the accelerometer. So, if we set a larger value as a high-pass filter cutoff frequency, important speech features will be lost. Zhang et el., in their work accelWord [36], also observed this challenge and did an information gain analysis to determine the optimum value. According to their analysis, if the cutoff value is greater than 2 Hz, then information gained with frequency domain features will reduce significantly. We did the information gain analysis on ear speaker data and found that even a value equal to or less than 1 Hz causes a significant amount of missing information. We collected all data from a single dataset in one go to avoid noise bias.

After that, we analyze the vibration of speech generated in raw accelerometer data. In previous literature, [1, 26], authors claimed that they found the most impact of vibrations are generated by phone speakers along the Z-axis. As we are working with ear speakers, we measure the impact of tiny vibrations along the X, Y, and Z axis. We observe that variance along the X, Y, and Z axis as $1.7029 * 10^{-6}$, $1.7029 * 10^{-6}$, and $1.946 * 10^{-4}$. It is obvious that this observation is in line with the observation of previous literature, which implies the Z axis gets more impact of vibrations compared to the X and Y axis.

We developed a program in MATLAB to analyze the accelerometer data and detect the word region. When a speech is played on the ear speakers, spikes can be noticed in the Z-axis value of the accelerometer. We present accelerometer data when the word "Zero" is uttered six times within 5 seconds timeframe in Figure 4 for the loudspeaker and ear speaker scenario. As the loudspeaker has a larger impact on accelerometer data, all individual word regions are visible (Figure 4a) and easy to detect. In contrast, ear speaker has a lower impact on accelerometer data (4b) and hence are hard to detect. In our presented example of Figure 4, where all six word regions are visible for the loudspeaker, only four word regions can be distinguished for the ear speaker. We observe that our program can automatically detect at least 45% of word regions from the raw accelerometer data and calculate time and frequency domain features (detailed discussion in the following subsection).

4.3 Tools Used for This Study

As we have discussed before, our primary goal is to analyze the accelerometer data while playing audio from ear speakers. For this experiment, we choose smartphones that have powerful/ multiple ear speakers. We have used OnePlus 7T and the OnePlus 9, which meet the requirements.

Both phones used for testing run on Android (Oneplus 7T runs on Android 11 while Oneplus 9 runs on Android 12). We have used a third-party Android app *Mobile Ear Speaker Earphone* [25] that runs a service that redirects all the output audio through ear speakers with default volume. We have also used another third-party app *Physics Toolbox Sensor Suite* [24] to collect accelerometer data while audio from datasets is played.

We have used a MATLAB program to analyze the accelerometer data and extract time and frequency domain features. To train the time and frequency features of data, we used *Weka* [32], which provides a collection of machine learning algorithms and essential analysis tools. We have

Table 1: Time and frequency domain features.

Time Domain Features	Frequency Domain Features
minfreq, maxfreq, meanfreq, standard deviation, variance, range, CV, skewness, kurtosis, quantile25, quantile50, quantile85, maeanCrossingRate	Energy, Entropy, Frequency Ratio, Irregularity K, Irregularity J, sharpness, smoothness, specCentroid, specStdDev, specCrest, specSkewness, specKurt

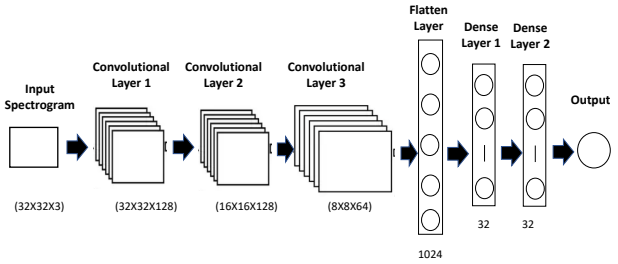


Figure 5: CNN model used for spectrogram-based image classifier.

also designed a Convolutional Neural Network (CNN) to analyze time and frequency domain data.

Using our developed MATLAB program, we also generate a spectrogram for each word region. We have also designed a CNN-based image classifier that can be fed with the generated spectrograms and classify them to detect gender, speaker, and speech.

4.4 Time-Frequency Domain Feature Analysis

We extracted time and frequency domain features using our developed MATLAB program. We use these features to train classical machine-learning algorithms using Weka. Initially, we checked with 40 different classifiers and found *RandomForest, RandomSubspace, and DecisionTables* are showing better performance than others. We use 80%/20% train/test split and 10-Fold cross-validation. We also use these time-frequency domain features to train our developed CNN model, and there we have also used 80%/20% train/test splits. The time-frequency domain features we have used for this experiment are listed in Table 1.

4.5 CNN Model Details

4.5.1 ***Spectrogram-based Image Classifier***. We use an image classifier for spectrogram analysis to take spectrogram inputs and then classify them.

Pre-processing: We prepare training and testing data from the generated labeled spectrograms as a Hierarchical Data Format version 5 (HDF5) file. Afterward, we convert the generated spectrogram into 128X128 images and prepare training and testing data by attaching appropriate labels.

CNN Details: In our designed model, there are three convolutional layers followed by three fully connected layers shown in Figure 5. The first convolutional layer takes 128X128 images and contains 128 filters. The second and third convolutional layer includes 128 and 64 filters, respectively. Each convolutional layer is followed by a ReLU function, a dropout layer with 0.2 rates, and a max-pooling layer (pool size (2X2)). After three convolution layers, we placed three fully connected layers. The first two layers reduce the size of the image to 128 and 64, respectively, with the ReLU activation function. The third layer comes up with the "softmax" activation function and changes the image size according to class size. The detailed CNN model is illustrated in Figure 5. We have used Root Mean Square Propagation (RMSProp) optimizer while training the model with spectrogram images.

4.5.2 ***Time-frequency Domain Feature based CNN Classifier***. We collect time-frequency domain features, write them into a CSV file and feed this data into our designed CNN model that classifies based on the time-frequency domain features.

Pre-processing: Our developed MATLAB program calculates time-frequency domain features with the label for each word region and generates a CSV file that contains all the information. After importing the CSV file, we check if there is any NaN (Not a Number) value on time and frequency domain features.

CNN Details: In our designed time-frequency feature-based classifier model, we use five convolutional layers followed by one dense layer (with softmax activation function). The first two convolutional layers contain 256 filters, the third convolutional layer has 128 filters, and the fourth and fifth one has 64 filters each. We have used dropout layers with a rate of 0.25 in the second and third convolution layers. We also used batch normalization in the second and third convolutional layers. Each of the convolutional layers used the ReLU activation function. Finally, we used a fully connected layer with a class size containing the softmax function. We have used Root Mean Square Propagation (RMSProp) optimizer while training the model with time and frequency domain features. An overview of the CNN model used for this purpose is depicted in Figure 6.

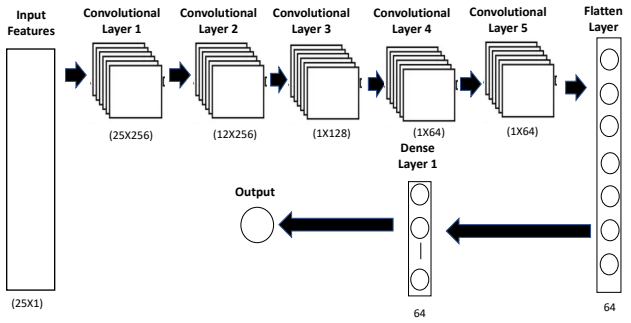


Figure 6: CNN model used for time and frequency domain feature Analysis.

5 EVALUATION

In this section, we discuss details about our experiment setup, dataset, and data collection methods. Most importantly, we discuss how our designed system can extract speech properties (i.e., speech, gender, and speaker information) from vibration induced by the ear speaker and evaluate the performance.

5.1 Experiment Setup

Data Collection Method: We use the natural handheld position of smartphone users that they use during a phone conversation in the experiment. As discussed earlier, we play the audio from the selected dataset, and a third-party app collects the accelerometer data at that time.

Dataset Selection: We use publicly available and well-known datasets to evaluate if our designed system can identify the speech information from the vibration induced by ear speakers. For gender and speaker detection, we use JL-Corpus [17], and emo-DB [11] datasets. The JL-Corpus dataset has 2400 utterances with four different speakers (two males and two females), whereas the emo-DB dataset has 535 utterances with ten actors (five males and five females). Actors use English in the JL-Corpus dataset and German in the emo-DB dataset as utterance language. For speech recognition, we use the digit dataset *Free Spoken Digit Dataset* [14] with the utterance of six(6) actors. Each actor utters each digit ten times (a total of 500 utterances per actor).

We have done a preliminary check on the datasets to examine the audio quality of the utterances. We have found that, in the FSDD dataset, three of the actor's data contain too much background noise or inconsistent volume during the recording. So, we removed these 1500 data and worked with only the remaining 1500 data in FSDD datasets. We have observed that data from emo-DB and JL-Corpus do not have similar problems. In addition to that, as the ear speakers produce low-volume audio output, the impact on the accelerometer is minimal. So, our word region detection program cannot identify all the word regions. However, it can detect 45% to 90% data, which is reasonable considering the low impact of vibration in the accelerometer.

Device Selection: As discussed earlier, our primary focus is to determine if ear speakers of recent smartphones that use stereo speaker feature and have powerful and multiple speakers on top are generating enough vibration on the motion sensor (i.e., accelerometer), so that, individual speech features can be identified. So, we measure the sound pressure level of each phone when they are playing the same audio from the FSDD dataset. From our experiment, we observe that only the *OnePlus 7T* and the *OnePlus 9* generate greater sound pressure than other phones (Oneplus 7T shows 42-46 dB, where OnePlus 9 shows 40-44 dB). So, we select these two phones for further experiments.

Posture Selection: In this work, our primary goal is to evaluate the risk of voice conversation in the phone through ear speakers. So, experiment data collectors keep their phones in the natural handheld posture while collecting data. All data are collected when data collectors sit on the chair. We have collected accelerometer readings for the whole dataset on one go to avoid human movement noise-induced bias on different classes.

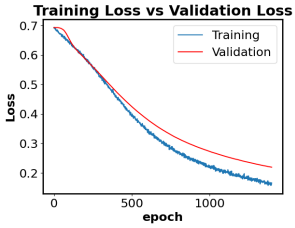
5.2 Data Collection Details

Audio File Preparation: We collect all dataset's audio files and sort them according to class in a folder so that it plays one class after another. Before starting the data collection, we played the audio file once to recheck if every audio file was playing correctly and note down the time when the audio file of a specific class was finished.

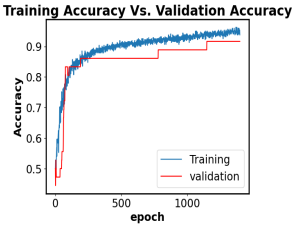
Tools Used in Data Collection: We have used *Physics Toolbox Sensor Suite* to collect accelerometer data. The OnePlus 7T and the OnePlus 9 phones' default sampling rates are 420 Hz and 520 Hz, respectively. We collect accelerometer data using this sampling rate. We have exported the collected data as a CSV file which is used for the MATLAB feature extraction program's input.

5.3 Feature Extraction

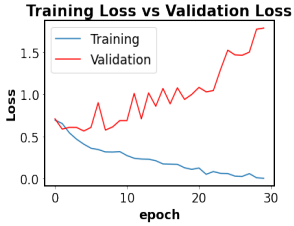
We have developed programs using MATLAB for time and frequency domain feature extraction and spectrogram generation. The first program takes the accelerometer data CSV file and detects each word region, and then extracts the time and domain frequency feature of each word region. This program writes and exports all time and frequency domain features in an external file and labels the data according to class. After that, we used Weka [32] to classify using classical machine learning algorithms.



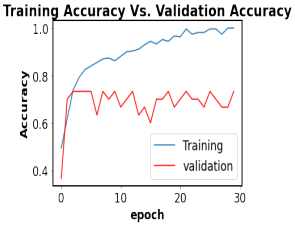
(a) Gender recognition training loss Vs. validation loss for Emo-DB dataset (time-frequency feature analysis).



(b) Gender recognition training accuracy Vs. validation accuracy for Emo-DB dataset (time-frequency feature analysis).



(c) Gender recognition training loss Vs. validation loss for Emo-DB dataset (spectrogram analysis).



(d) Gender recognition training accuracy Vs. validation accuracy for Emo-DB dataset (spectrogram analysis).

Figure 7: Gender recognition training and validation accuracy graph using different methods.

The Spectrogram generator (developed using MATLAB) can also detect the word regions and generate spectrograms for each word region. The generated spectrograms are labeled according to class. Later these generated spectrogram is used to feed into our developed CNN for further analysis.

5.4 Gender Recognition

Gender Detection: We use JL-Corpus [17], and emo-DB [11] dataset to evaluate if the caller's gender can be detected from the accelerometer data. We have used three methods to evaluate: (1) Classical machine learning algorithm with time and frequency domain features. (2) CNN with time and frequency domain features (3) CNN with generated spectrogram for each word region.

ML Algorithm with Time/Frequency Domain Features: For the emo-DB dataset, our detection program can detect 448 word regions among 535 original utterances for the OnePlus 7T and 300 word regions for OnePlus 9. We have extracted all detected word regions' time/frequency domain features. We have used "RandomForest", "RandomSubspace", and "Decision Table" as classifiers for our analysis. We have used 80/20 train/test split and 10-Fold cross-validation for our analysis.

For RandomForest Classifier, we have achieved 98.66% accuracy in classifying genders. Whereas for RandomSubspace, we have also achieved 98.66% accuracy, and for Decision Table, we have observed 98.21% accuracy for the OnePlus 7T. The detailed result is shown in Table 2 and Table 5. Similarly, for the OnePlus 9, we get 88.67%, 77.71%, and 84.67% accuracy for RandomForest, RandomSubspace, and Decision Table classifiers, respectively. The detailed result is shown in Table 2 and Table 6.

For the JL-Corpus dataset, our detection program can detect 1469 word regions among 2400 utterances. Here, we also have used RandomForest, RandomSubspace, and Decision Tree as classifiers for our analysis.

For RandomForest Classifier, we have achieved 78.62% accuracy in classifying genders. Whereas for RandomSubspace, we have also achieved 79.37% accuracy, and for Decision Table, we have observed 77.67% accuracy for the OnePlus 7T. Similarly, we got 77.71%, 74.20%, and 72.14% accuracy for the OnePlus 9. The detailed result is shown in Table 2, Table 5, and Table 6.

CNN with Time/Frequency Domain Features: As discussed, we have extracted time and frequency domain features for 448 word regions for emo-DB and 1469 word regions for JL-Corpus datasets. We have designed a CNN to classify time and frequency domain features (Details are described in Section 4.5.2).

We have used *binary_crossentroy* as loss function and *Root Mean Square Propagation (RMSProp)* as the optimizer and 80/20 split as train/test split in our analysis. We have achieved 95.55% for the emo-DB dataset and 75.71% accuracy for the JL-Corpus dataset for the OnePlus 7T. We also got 83.33% and 67.52% accuracy for the emo-DB and JL-Corpus datasets, respectively, for the OnePlus 9 device. Training loss Vs. validation loss and training accuracy Vs. validation accuracy charts are shown in Figure 7.

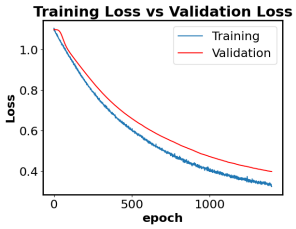
CNN with Spectrogram: We generate spectrograms for all word regions our algorithm has detected. We train our developed CNN model with the extracted spectrograms. We observed, at best 79.72% accuracy on analyzing, which is also lower compared to what we get from classical machine learning algorithms. We can see the loss and accuracy graph in Figure 7d and Figure 7c. Details results on gender detection are listed in Table 2.

5.5 Speaker Detection

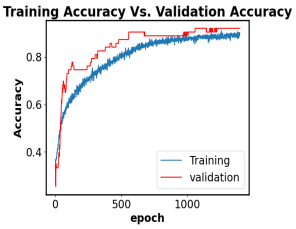
We use JL-Corpus [17] and FSDD (Free Spoken Digit Dataset) [14] dataset to evaluate if the speaker's identity can be detected from the accelerometer data. We use the same ML algorithms and CNN classifiers that we have used in gender detection.

Table 2: Gender recognition accuracy (random guess 50%).

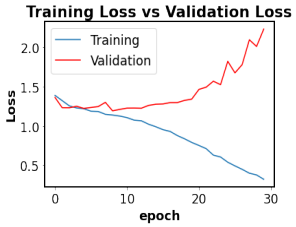
Method	Classifier	Data set	Accuracy (OnePlus 7T)	Accuracy (OnePlus 9)
Time and Frequency Domain Features	Random Forest	Emo-DB	**98.66%**	88.67%
		JL-Corpus	78.62%	77.71%
	Random Subspace	Emo-DB	98.66%	84.67%
		JL-Corpus	79.37%	74.20%
	Decision Table	Emo-DB	98.21%	84.67%
		JL-Corpus	77.67%	72.14%
	CNN	Emo-DB	95.55%	83.33%
		JL-Corpus	75.17%	67.52%
Spectrogram	CNN	Emo-DB	79.72%	69.69%
		JL-Corpus	70.10%	65.53%



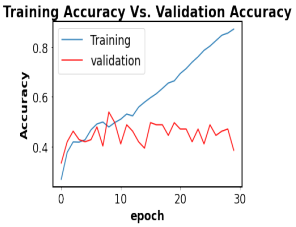
(a) Speaker recognition training loss Vs. validation loss for FSDD dataset (time-frequency feature analysis).



(b) Speaker recognition training accuracy Vs. validation accuracy for FSDD dataset (time-frequency feature analysis).



(c) Speaker recognition training loss Vs. validation loss for JL-Corpus dataset (spectrogram analysis).



(d) Speaker recognition training accuracy Vs. validation accuracy for JL-Corpus dataset (spectrogram analysis).

Figure 8: Speaker recognition training and validation accuracy graph using different methods.

ML Algorithm with Time/Frequency Domain Features: For the FSDD dataset, our detection program can detect 788 word regions among 1500 (1500 data was removed for inconsistent volume and background noise as discussed earlier) original utterances with three different classes for OnePlus 7T. For the OnePlus 9, the total number of extraction is 618. We extract all detected word regions' time/frequency domain features and use RandomForest, RandomSubspace, and Decision Tree as classifiers for our analysis. We use 80/20 train/test split and 10-Fold cross-validation similar to gender detection.

For RandomForest Classifier and FSDD dataset, we have achieved 91.24% accuracy in classifying genders for the OnePlus 7T device and 87.75% for the OnePlus 9. Whereas for RandomSubspace, we have also achieved 90.98% and 88.70% accuracy for these devices. For Decision Table, we have observed 90.22% accuracy for the OnePlus 7T and 88.23% accuracy for the OnePlus 9. The detailed result is shown in Table 3, Table 5, and Table 6.

Like previous analysis, for the JL-Corpus dataset, our detection program can detect 1469 word regions among 2400 utterances with four different classes. Here, we also have used the same classifiers to evaluate our results.

For RandomForest Classifier, we have achieved 64.60% and 61.50% accuracy in classifying genders for OnePlus 7T and OnePlus 9. Whereas for RandomSubspace, we have also achieved 64.32% and 59.86% accuracy, and for Decision Table, we have observed 63.03% and 55.72% accuracy for the OnePlus 7T and OnePlus 9 devices, respectively. The detailed result is shown in Table 3, Table 5, and Table 6.

CNN with Time/Frequency Domain Features: As discussed, we have extracted time and frequency domain features for 788 word regions with three different speakers for FSDD and 1469 word regions with four different speakers for JL-Corpus datasets. Our designed CNN is the same that we have used in gender detection.

We have used *categorical_crossentroy* as the loss function and *Root Mean Square Propagation (RMSProp)* as the optimizer and 80/20 split as the train/test split in our analysis. We have achieved 86.07% and 78.12% for the FSDD dataset and 60.20% and 57.73% accuracy for the JL-Corpus dataset for OnePlus 7T and OnePlus 9 devices. Training loss Vs. validation loss and training accuracy Vs. validation accuracy charts are shown in Figure 8.

CNN with Spectrogram: We generate spectrograms for all identified word regions and label them accordingly. After that, we fed generated spectrogram to our designed CNN-based image classifier. We collect data from two datasets (FSDD and JL-Corpus), and the CNN-based image classifier shows up to 45.23% accuracy for the JL-Corpus dataset. We observe that accuracy is much lower than in classical machine learning algorithms. Loss and accuracy analysis are illustrated in Figure 8d and Figure 8c.

Details results on gender detection are listed in Table 3.

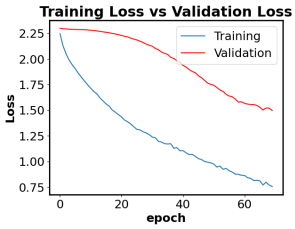
5.6 Speech Recognition

As a representative speech recognition dataset, we used the FSDD (Free Spoken Digit Dataset) [14] dataset containing audio records of three different actors uttering digits 0 (zero) to 9 (nine). We evaluate the impacts on the accelerometer and try to find out if every digit can be distinguished using raw accelerometer data. We have used the same ML algorithms and CNN classifiers that we have used in gender detection.

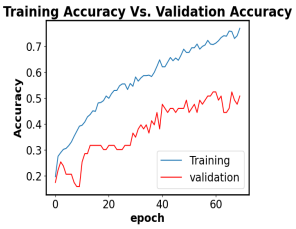
ML Algorithm with Time/Frequency Domain Features: For the FSDD dataset, our detection program can

Table 3: Speaker recognition accuracy.

Method	Classifier	Data set	Accuracy (Random Guess) (OnePlus 7T)	Accuracy (Random Guess) (OnePlus 9)
Time and Frequency Domain Features	Random Forest	FSDD	**91.24%** (33%)	87.75% (33%)
		JL-Corpus	64.60% (25%)	61.50% (25%)
	Random Subspace	FSDD	90.98% (33%)	88.70% (33%)
		JL-Corpus	64.32% (25%)	59.86% (25%)
	Decision Table	FSDD	90.22% (33%)	88.23% (33%)
		JL-Corpus	63.03% (25%)	55.72% (25%)
	CNN	FSDD	86.07% (33%)	78.12% (33%)
		JL-Corpus	60.20% (25%)	57.73% (25%)
Spectrogram	CNN	FSDD	35% (33%)	36.44% (33%)
		JL-Corpus	44.32% (25%)	45.23% (25%)



(a) Speech recognition training loss Vs. validation loss for FSDD dataset (time-frequency feature analysis).



(b) Speech recognition training accuracy Vs. validation accuracy for FSDD dataset (time-frequency feature analysis).

Figure 9: Speech recognition training and validation accuracy graph using CNN with time-frequency features.

detect 788 (OnePlus 7T)and 630 (OnePlus 9) word regions among 1500 (1500 data was removed for inconsistent volume and background noise as discussed earlier) original utterances with ten different classes. We have extracted all detected word regions' time/frequency domain features. We have used RandomForest, RandomSubspace, and Decision Tree as classifiers for our analysis, similar to gender and speaker analysis. We also use 80/20 train/test split and 10-Fold cross-validation similar to the previous analysis.

For RandomForest Classifier, we have achieved 54.46% accuracy in classifying genders. Whereas for RandomSubspace, we have also achieved 54.46% accuracy, and for Decision Table, we have observed 45.53% accuracy for the OnePlus 7T.

CNN with Time/Frequency Domain Features: As discussed, we have extracted time and frequency domain features for word regions extracted for both phones with ten different speakers for FSDD. Our designed CNN is the same that we have used in gender and speaker detection.

Similar to speaker analysis, we have used *categorical_crossentroy* as the loss function and *Root Mean Square Propagation (RMSProp)* as the optimizer and 80/20 split as the train/test split in our analysis. We have achieved 41.02% and 38.70% accuracy for the CNN analysis for OnePlus 7T and OnePlus 9 devices. Training loss Vs. validation loss and training accuracy Vs. validation accuracy charts are shown in Figure 9.

Table 4: Speech recognition accuracy (random guess 10%).

Method	Classifier	Data set	Accuracy (OnePlus 7T)	Accuracy (OnePlus 9)
Time and Frequency Domain Features	Random Forest	FSDD	53.59%	41.59%
	Random Subspace	FSDD	**56.42**%	38.99%
	Decision Table	FSDD	51.80%	33.33%
	CNN	FSDD	41.02%	38.70%

The detailed result is shown in Table 4, Table 5, and Table 6

5.7 Result Summary

Table 5: Detection performance of ML algorithm with time/frequency domain features for OnePlus 7T device.

Detection	Classifier	Data set	TP Rate	FP Rate	Precision	Recall
Gender	Random Forest	emo-DB	98.7%	1.3%	98.7%	98.7%
		JL-Corpus	78.6%	21.7%	78.8%	78.6%
	Random Subspace	emo-DB	98.7%	1.3%	98.7%	98.7%
		JL-Corpus	79.4%	21.0%	79.8%	79.4%
	Decision Table	emo-DB	98.2%	1.9%	98.2%	98.2%
		JL-Corpus	77.7%	22.5%	77.7%	77.7%
Speaker	Random Forest	FSDD	91.2%	4.6%	91.4%	91.2%
		JL-Corpus	64.6%	11.6%	66.3%	64.6%
	Random Subspace	FSDD	91.0%	4.7%	91.5%	91.0%
		JL-Corpus	64.3%	11.5%	67.5%	64.3%
	Decision Table	FSDD	90.2%	5.1%	90.4%	90.2%
		JL-Corpus	63.0%	11.9%	66.9%	63.0%
Speech	Random Forest	FSDD	53.6%	5.1%	52.2%	53.6%
	Random Subspace	FSDD	56.4%	4.8%	55.3%	56.4%
	Decision Table	FSDD	51.8%	5.4%	50.9%	51.8%

Gender Detection: After evaluation with two different datasets and three different methods, we found reasonable accuracy with the highest achieved accuracy of 98.6% in classical ML algorithm analysis for emo-DB datasets, which contains 10 different actors (5 male actors, 5 female actors). We have also evaluated with JL-Corpus dataset that contains 2400 utterances (we extracted 1469 word regions). JL-Corpus utterances are collected from 4 speakers (2 males and 2 females). We have achieved 79.37% accuracy on the JL-Corpus dataset using the classical ML algorithm in gender detection, which is also a reasonable accuracy. CNN analysis with time and frequency domain features is also in line with the accuracy we get using ML algorithms (95.55% accuracy with the FSDD dataset and 75.17%).

Spearphone [2] shows the highest 99% accuracy in gender detection using the motion sensor and loudspeaker of the smartphone. Considering the fact that ear speakers induced

much lower vibration on the motion sensor, the gender recognition accuracy is almost similar to loudspeakers, which is an interesting observation. Another work Face-mic [22], use face vibrations on AR/VR devices and achieves 96.81% accuracy in gender recognition. Compared to this work, we can say that vibration from ear speakers performs better in recognizing gender compared to face movements induced vibration.

Speaker Detection: We evaluated with two different datasets (FSDD and JL-Corpus dataset). Among them, the FSDD datasets we used contains 788 utterances with 3 different actors and show 91.24% accuracy using ML classifiers. The JL-Corpus dataset has 1469 utterances with four actors and shows the highest 64.60% accuracy in speaker detection, which is still two times greater than a random guess. CNN analysis with time and frequency domain features also shows similar accuracy here (86.07% for the FSDD dataset and 60.20% for the JL-Corpus dataset).

Using the same handheld scenario, spearphone [2] achieved 99% accuracy for one device in classifying ten speakers. On the other hand, we achieve, at best 91.24% accuracy in classifying three speakers. Although loudspeaker performance is better than ear speakers in this case, the observed accuracy reveals the potential vulnerability of identifying speaker-specific information from ear speakers just using built-in zero permission motion sensors. It also shows slightly lower but still good accuracy compared to face movement-induced vibration (Face-Mic [22]).

Speech Detection: To evaluate speech detection, we have used the digit dataset FSDD, where three actors utter ten different digits. We evaluate the time and frequency domain features with classical ML algorithms, which show the highest 56.42% accuracy. As there are ten different classes here, the accuracy still exhibits five times greater accuracy than a random guess, which implies that vibration due to the ear speaker induced a reasonable amount of distinguishable impact on accelerometer data.

Previous works also show lower accuracy in speech detection compared to gender and speaker detection. Spearphone [2]. shows 80% accuracy in recognizing digits. In work, mmSpy [5], which uses an external receiver to sense vibration from the ear speaker, can achieve 83% accuracy at 1 ft distance and 47.99% accuracy at 6 ft distance. Compared to a practical attack scenario (4-6 ft of distance from the phone), our result is promising in detecting speech digit data.

Other Performance Evaluation Metrics: We list down TP-rate, FP-rate, Precision, and Recall of our analysis using classical machine learning algorithm in Table 5 and Table 6.

TP Rate (True Positive Rate) indicates the rate of correctly classified elements. FP Rate (False Positive Rate) shows the rate of incorrectly classified elements for a particular class.

Table 6: Detection performance of ML algorithm with time/frequency domain features for OnePlus 9 device.

Detection	Classifier	Data set	TP Rate	FP Rate	Precision	Recall
Gender	Random Forest	emo-DB	88.7%	11.8%	88.7%	88.7%
		JL-Corpus	78.6%	21.7%	78.8%	78.6%
	Random Subspace	emo-DB	84.7%	15.4%	84.7%	84.7%
		JL-Corpus	79.4%	21.0%	79.8%	79.4%
	Decision Table	emo-DB	84.7%	16.7%	84.8%	84.7%
		JL-Corpus	77.7%	22.5%	77.7%	77.7%
Speaker	Random Forest	FSDD	87.8%	5.8%	87.9%	87.8%
		JL-Corpus	61.5%	13.2%	61.1%	61.5%
	Random Subspace	FSDD	88.7%	5.2%	89.1%	88.7%
		JL-Corpus	55.7%	15.5%	55.5%	55.7%
	Decision Table	FSDD	88.2%	5.4%	88.8%	88.2%
		JL-Corpus	59.9%	13.7%	59.6%	59.9%
Speech	Random Forest	FSDD	41.6%	6.8%	41.6%	41.6%
	Random Subspace	FSDD	39.0%	7.2%	39.1%	39.0%
	Decision Table	FSDD	33.3%	8.0%	33.6%	33.3%

FP Rate (False Positive Rate) shows the rate of incorrectly classified elements for a particular class. Precision indicates the proportion of correctly classified elements and all classified elements for a particular class. On the other hand, recall suggests the proportion of correctly classified elements and all the elements present in the class.

6 DISCUSSION AND FUTURE WORK

6.1 General Discussion

Sensor Rate Limit: To protect potentially sensitive information about users, if the app targets Android 12 (API level 31) or higher, the system has a limit on the refresh rate of data from certain motion sensors and position sensors. This data includes values recorded by the device's accelerometer. However, we performed gender classification by utilizing the emo-DB dataset and accomplished 90.97% accuracy at a 200 Hz sampling rate, which is still a high reasonable accuracy. In this case, the restriction for the sensor rate does not impact much on the eavesdropping threat.

Band-pass Filter for Human Movement: Currently, we are collecting all the data in one go. However, if we place a high-pass filter, the filter would still severely wipe out speech properties. For the ear speaker case, even if we placed a 1 Hz high pass filter, it reduces all speech properties. We conducted another test where we checked the accuracy when the phone was placed on a phone holder clamp, and no movement like the human hand gesture was involved but held in a handheld position. We achieved 97.3% for gender classification accuracy with the emo-DB dataset in the clamp testing. The results imply that human movement does not play any significant role in that. Thus, we do not remove any speech property by placing a high pass filter.

6.2 Limitation

Although recent smartphones use larger and more powerful ear speakers, they still reduce the volume at a reasonable level to ensure the comfort of the users during a phone conversation. As a result, they cannot generate a significant impact on raw accelerometer data. For this reason, our word region detection algorithm cannot detect a high percentage of the word uttered (it can detect 45% - 80% of words or speech in total). However, our result indicates that it is sufficient for the adversary to reasonably detect significant speech features (e.g., gender, speaker's identity, speech).

Impacts on the accelerometer due to the vibration of motion sensors are highly dependent upon the distance between the ear speaker and the motion sensor. It depends on the design of the smartphone's motherboard, which varies from manufacturer to manufacturer and even varies from model to model of the same manufacturer. So, our observed accuracy is not constant and can show slightly different results in terms of accuracy.

The data collected from the accelerometer can be noisy due to the hand and body movement of the user. However, as discussed in the previous subsection, even if we place 1 Hz high pass filter, it removes important speech features due to a very low impact on accelerometers by ear speakers. As such, removing low-frequency noises, in this case, is a challenge. However, we conducted another experiment where we emulated the handheld scenario with the phone attached to a clamp where there was no body-induced vibration present. We got 97.3% accuracy compared to 98.6% accuracy in our result, which implies there is a very low impact of noises in determining the accuracy.

6.3 Countermeasures

One of the potential countermeasures is to change the permission model of motion sensors so that third-party apps cannot record sensor data without the permission of users. Recently, Android has restricted sensor data collection without permission [9] for sampling rates beyond 200 Hz. However, this does not completely prevent silent eavesdropping using motion sensors. As discussed in the previous subsection, we have done another experiment on Gender detection where we collected all data at 200 Hz sampling rate instead of the default sampling rate (e.g., 420 Hz for OnePlus 7T and 520 Hz for OnePlus 9). We got 90.97% accuracy in gender detection compared to 98.6% accuracy we observed with the default sampling rate for OnePlus 7T phone. As such, this countermeasure cannot fully prevent the user from silent eavesdropping.

Smartphone manufacturers should be more careful about designing larger and more powerful ear speaker volume control. They should maintain the same sound pressure during phone conversation as previous generation phones ear speakers. Moreover, they should place the motion sensors in the proper position relative to the ear speaker so that the phone speaker's vibration impact can be minimized.

6.4 Future Works

Our work opens immense research opportunities on eavesdropping possibilities on ear speakers using smartphones' built-in sensors. As the ear speaker has very little impact on accelerometer data, it is always a challenge to effectively extract all word regions from it. Researchers can solve the challenge by proposing more efficient algorithms that increase detection rates. They also have the opportunity to design machine learning or deep learning techniques to achieve more accuracy in speech information extraction in general.

This research primarily focuses on speech recognition and some speech feature (e.g., speaker's identity, gender) eavesdropping from the ear speaker-induced vibration in an accelerometer. Researchers can work on other speech features (e.g., language) extraction from motion sensor data. They also have the opportunity to work on potential preventive and mitigating measures for eavesdropping from ear speakers.

7 CONCLUSION

This work focused on the unexplored area of eavesdropping possibility using smartphone ear speakers, especially with the device equipped with multiple powerful speakers that are used as ear speakers. We investigate the reverberation effect of ear speakers on a built-in accelerometer by extracting time-frequency domain features and spectrograms. We evaluate them using classical machine learning algorithms and our developed convolutional neural network (CNN) models. We found up to 98.6% accuracy on gender detection, up to 92.6% accuracy on speaker detection, and up to 56.42% accuracy on speech detection, which proves the presence of distinguishing speech features in the accelerometer data that the adversaries can leverage for eavesdropping. Our findings also open the opportunity for researchers to explore recently popular powerful ear speakers' potential risk factors.

REFERENCES

[1] S Abhishek Anand, Chen Wang, Jian Liu, Nitesh Saxena, and Yingying Chen. 2019. Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers. *arXiv preprint arXiv:1907.05972* (2019).

[2] S Abhishek Anand, Chen Wang, Jian Liu, Nitesh Saxena, and Yingying Chen. 2021. Spearphone: a lightweight speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 288–299.

[3] Android Authority. 2022. Google will kill call recording apps on Android for good starting May 11.

https://www.androidauthority.com/google-killing-call-recording-apps-3155610/.

[4] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based Practical Smartphone Eavesdropping with Built-in Accelerometer.. In *NDSS*.

[5] Suryoday Basak and Mahanth Gowda. 2022. mmSpy: Spying Phone Calls using mmWave Radars. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1211–1228.

[6] Supriyo Chakraborty, Wentao Ouyang, and Mani Srivastava. 2017. LightSpy: Optical eavesdropping on displays using light sensors on mobile devices. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2980–2989.

[7] Kirsten Crager and Anindya Maiti. 2017. Information leakage through mobile motion sensors: User awareness and concerns. In *Proceedings of the European Workshop on Usable Security (EuroUSEC)*.

[8] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Fredo Durand, and William T Freeman. 2014. The visual microphone: Passive recovery of sound from video. (2014).

[9] Google Developers. 2022. Sensors Overview. https://developer.android.com/guide/topics/sensors/sensors_overview.

[10] DXOMARK. 2021. 2000 to 2021: The evolution of smartphone audio playback. https://www.dxomark.com/evolution-of-smartphone-audio-playback/.

[11] Emo-DB. 2022. Berlin Database of Emotional Speech. http://emodb.bilderbar.info/start.html.

[12] Engadget. 2022. Google is banning third-party call recording apps from the Play Store. https://www.engadget.com/google-is-banning-third-party-call-recording-apps-from-the-play-store-093201443.html.

[13] Pengfei Hu, Hui Zhuang, Panneer Selvam Santhalingam, Riccardo Spolaor, Parth Pathak, Guoming Zhang, and Xiuzhen Cheng. 2022. AccEar: Accelerometer Acoustic Eavesdropping with Unconstrained Vocabulary. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1530–1530.

[14] Zohar Jackson. 2022. Free Spoken Digit Dataset. https://github.com/Jakobovski/free-spoken-digit-dataset.

[15] JerryRigEverything. 2019. OnePlus 7T Teardown. https://www.youtube.com/watch?v=_PySmMrihWI.

[16] JerryRigEverything. 2019. OnePlus 7T Teardown! - Is the Oreo Camera really needed? https://www.youtube.com/watch?v=_PySmMrihWI.

[17] Kaggle. 2022. JL Corpus. https://www.kaggle.com/datasets/tli725/jl-corpus.

[18] Yihao Liu, Kai Huang, Xingzhe Song, Boyuan Yang, and Wei Gao. 2020. MagHacker: eavesdropping on stylus pen writing via magnetic sensing from commodity mobile devices. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 148–160.

[19] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium (USENIX Security 14)*. 1053–1067.

[20] Ralph P Muscatell. 1984. Laser microphone. *The Journal of the Acoustical Society of America* 76, 4 (1984), 1284–1284.

[21] Ben Nassi, Yaron Pirutin, Tomer Galor, Yuval Elovici, and Boris Zadov. 2021. Glowworm Attack: Optical TEMPEST Sound Recovery via a Device's Power Indicator LED. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1900–1914.

[22] Cong Shi, Xiangyu Xu, Tianfang Zhang, Payton Walker, Yi Wu, Jian Liu, Nitesh Saxena, Yingying Chen, and Jiadi Yu. 2021. Face-Mic: inferring live speech and speaker identity via subtle facial dynamics captured by AR/VR motion sensors. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 478–490.

[23] Omega Centauri Software. 2022. Earpiece. https://play.google.com/store/apps/details?id=mobi.omegacentauri.Earpiece.

[24] Vieyera Software. 2022. Physics Toolbox Sensor Suite. https://play.google.com/store/apps/details?id=com.chrystianvieyra.physicstoolboxsuite.

[25] Move More Solutions. 2022. Mobile Ear Speaker Earphone. https://play.google.com/store/apps/details?id=com.sparkapps.mobileearphone.yp.

[26] Weigao Su, Daibo Liu, Taiyuan Zhang, and Hongbo Jiang. 2021. Towards Device Independent Eavesdropping on Telephone Conversations with Built-in Accelerometer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–29.

[27] Payton Walker and Nitesh Saxena. 2022. Laser Meager Listener: A Scientific Exploration of Laser-based Speech Eavesdropping in Commercial User Space. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 537–554.

[28] Chao Wang, Feng Lin, Tiantian Liu, Kaidi Zheng, Zhibo Wang, Zhengxiong Li, Ming-Chun Huang, Wenyao Xu, and Kui Ren. 2022. mmEve: eavesdropping on smartphone's earpiece via COTS mmWave device. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 338–351.

[29] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M Ni. 2014. We can hear you with Wi-Fi!. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. 593–604.

[30] Ziqi Wang, Zhe Chen, Akash Deep Singh, Luis Garcia, Jun Luo, and Mani B Srivastava. 2020. UWHear: through-wall extraction and separation of audio vibrations using wireless signals. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 1–14.

[31] Teng Wei, Shu Wang, Anfu Zhou, and Xinyu Zhang. 2015. Acoustic eavesdropping through wireless vibrometry. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 130–141.

[32] Weka. 2022. The Data Platform for AI. https://www.weka.io/.

[33] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. 2019. Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 14–26.

[34] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. 113–124.

[35] Jiadi Yu, Li Lu, Yingying Chen, Yanmin Zhu, and Linghe Kong. 2019. An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing. *IEEE Transactions on Mobile Computing* 20, 2 (2019), 337–351.

[36] Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. 2015. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 301–315.

[37] Huadi Zheng and Haibo Hu. 2019. Missile: A system of mobile inertial sensor-based sensitive indoor location eavesdropping. *IEEE Transactions on Information Forensics and Security* 15 (2019), 3137–3151.