BOSTON COLLEGE

MORRISSEY COLLEGE OF ARTS AND SCIENCES

Dissertation

PASSIVE ACOUSTIC EAVESDROPPING

by

HUIDONG HOU

B.S in Mathematics, Computer Science, Boston College, 2023
Chestnut Hill, MA

Submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Science in Computer Science

Spring 2023

Approved by

First Reader

Siddhartan Govindasamy, PhD
Professor, Department of Engineering

Second Reader

Donglai Wei, PhD
Assistant Professor, Department of Computer Science

Third Reader

Howard Straubing, PhD
Professor, Department of Computer Science

ACKNOWLEDGMENTS

The work for this projected originated in the Summer of 2022 when I first started working in Professor Govindasamy's lab. I am deeply grateful for all the guidance and suggestions that Professor Govindasamy provided throughout this journey. Without him, this project will probably not take place. I have learnt a lot working under his guidance, as an independent researcher, and life long lessons.

My sincere thanks go to my co-advisor, Professor Donglai Wei, from the Computer Science Department, who started my journey in machine learning. He always inspires me and gives me invaluable life lessons beyond academic questions which I feel I am indebted to.

In addition to expressing my heartfelt appreciation to my advisors who has supported me throughout this project, I would like to specifically extend my deepest gratitude to Professor Howard Straubing for his invaluable contributions in reviewing this paper. As part of the requirements for this year's Scholars of College Project, Professor Straubing graciously agreed to invest his time and expertise to meticulously evaluate my work. His keen eye for detail and thorough examination helped identify numerous grammatical errors, thereby significantly enhancing the overall quality and coherence of the paper.

Lastly, I would like to convey my gratitude to my friend Jason Adhinarta who provided suggestions and feedback for the model, and everyone I met in the journey. Working on this project was probably one of my best memories during my undergraduate study at Boston College.

PASSIVE ACOUSTIC EAVESDROPPING

HUIDONG HOU

Boston College, Morrissey College of Arts and Sciences, Spring 2023

Advisor: Donglai Wei, Assistant Professor

ABSTRACT

The goal of this project is to demonstrate a potential eavesdropping hazard associated with 60GHz wireless communications, where acoustic signals can be detected by an eavesdropper by observing the wireless communications signals. Unlike recent works which use radar waveforms to perform eavesdropping by ranging, our system uses binary-shift-keying (BPSK) signals by exploiting variations in the multipath environment between a transmitter and receiver caused by vibrations due to an acoustic excitation. As such this system demonstrates a *passive* eavesdropping hazard, where an eavesdropper can passively observe the wireless communications signal transmitted by a legitimate source to listen in on acoustic signals, without requiring the eavesdropper to transmit any signal of its own. Using a custom deep learning frame work using pytorch, we have proved that we could obtain an accuracy of 30% classifications of accuracy for words.

CONTENTS

LIST OF FIGURES

CHAPTER 1
Experimental Design

1.1 BACKGROUND

In today's interconnected world, the need for privacy and security in communication is of paramount importance. As technology continues to evolve, so do the methods of eavesdropping, often becoming more sophisticated and harder to detect. Additionally, wireless communications systems are becoming very widely used in the world, making most indoor environments full of various wireless communications signals. The presence of these signals creates the potential for passive acoustic eavesdropping, a surreptitious technique that exploits vibrations of objects due to sound propagation to intercept hidden conversations, by observing small changes in the wireless communications signals due to movements of objects. Recently, there have been several works that demonstrate acoustic eavesdropping using high frequency, mmWave signals. For example, Hu et al. (2022b) uses mmWave radar and reconstructs the acoustic signals using a Generative Adversarial Network (GAN), Zheng et al. (2022) uses the piezoelectric effect of a special material placed in the vicinity of the attackee, and Hu et al. (2022a) uses a radar signal to observe the vibrations on the headset of a mobile phone to eavesdrop on speakers using headsets with mobile phones. All of these works use radar signals which are transmitted by the attackers themselves, and as such are active eavesdropping systems. The presence of the attacker could be detected by detecting the radar waveform.

Additionally, Wei et al. (2015) proposes several techniques for acoustic eavesdropping, including using a constant sine wave to reflect off objects as well as a

2.4GHz wifi-based eavesdropping system. However, the wifi based system in that work is for the specific case of of observing changes in the transmitted signal of a phone while a user is speaking in to it.

In contrast to the above works, this work considers a mmWave, specifically 60GHz, wireless communications signal which is reflected of objects near a speaker. Since this system uses a wireless communications signal, the source of the signal does not have to be the eavesdropper, but it could be from a legitimate signal transmitted by a device owned by the atackee themselves. In other words, the system we propose is a passive system, which does not require the attacker to transmit any signal. Hence, this system demonstrates an eavesdropping hazard associated with wireless communications signals.

1.2 SET UP

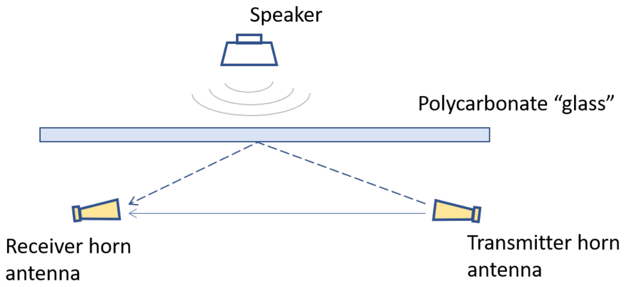


Figure 1.1: Conceptual Diagram of Eavesdropping System

Figure 1.1 illustrates the system used to demonstrate the security hazard in this work. The transmitter antenna sends a binary shift keying signal modulated at a frequency f_1. This signal can be described using the following equation

$$x(t) = m(t)\cos(2\pi f_1 t) \tag{1.1}$$

where $m(t)$ represents the data that is being transmitted. For simplicity, our experiment only used rectangular pulses, while in practical systems, other pulses such as a raised cosine may be used. Figure 2 provides an example of $m(t)$ that is being transmitted.

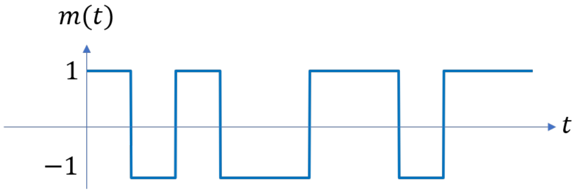


Figure 1.2: Example $m(t)$

While the transmitter antenna sends the signal, the speaker will transmit acoustic signals(in our experiment we used sine wave for synchronization followed by English words). The transmitted acoustic signal will cause the vibration of the poly-carbonate "glass", thus impact the reflected path the signal travels from the Transmitter antenna to the Receiver antenna.

In addition to the reflected path that the transmitter signal travels through, which is influenced by the poly-carbonate board, the transmitter signal will also travel in a direct path, or the shortest distance path, to the Receiver Antenna. The receiver observes the superposition of these two signals.

1.3 EXPERIMENTAL DESIGN

We conducted the experiment in two stages: using sine waves and words. During the summer of 2022, with the current physical setup illuistrated in 1.3, we initiated the Universal Software Radio Peripheral device(USRP),after the sine wave started playing from the speaker. Figure 1.3 will show a graphical representation od the design of the setup. Note that transmitter digital signal is what is made

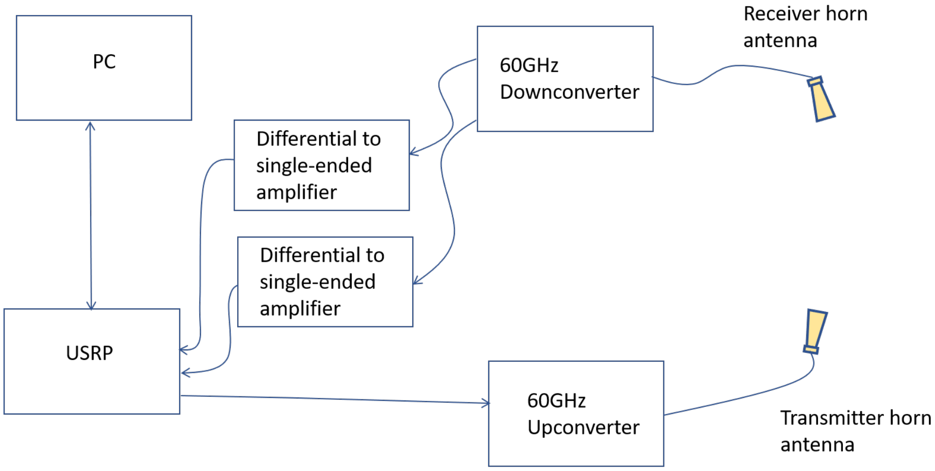
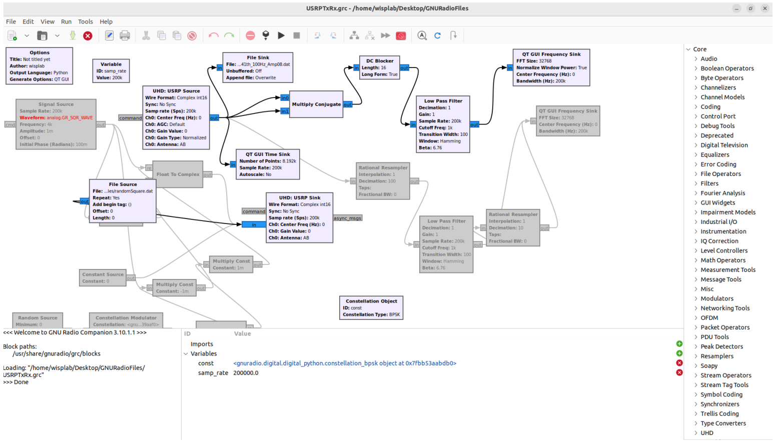


Figure 1.3: Physical Setup of the Experiment

available by our Universal Software-defined Radio (USRP) system, along with the GNURadio Software that is responsible for generating the signal. The computer interfaces with a Ettus Research USRP X310 which transmits a low frequency BPSK signal ($m(t)$) to the upconverter board which transmits a high frequency signal through a transmitter horn antenna (Eravant SAR-2309-15VF-R2). This signal is received by another horn antenna (Eravant SAR-2309-15VF-R2) which is then downconverted to lower frequencies using 60GHz downconverter. The resulting signals $y_I(t)$ and $y_Q(t)$ are differential signals which have to be converted to single-ended signals by two differential-to-single-ended amplifiers (Maxim Semiconductor MAX4444EVKIT). These signals are then fed into the USRP x310 which then collects that data and transmits to it to the PC. The USRP is controlled using a software called GNURadio, and the 60GHz transmitter and receiver boards are controlled using Analog Devices SOC Transmitter/Receiver software. For illustration purposes, screen shots of the software are shown in Figures 1.4, 1.5, 1.6

The USRP board generates the base-band(i.e, low frequency communication signal) given by $m(t)$ presented in 1.2. We used Analog Devices 60 G Hz EK1HMC6350 board up converter to convert the signal before transmission. The receiver, a 60G Hz downconverter(Receiver module of Analog Devices EK1HMC6350), converts the high frequency signal $y(t)$ into the low frequency signals $y_I(t)$ and $y_Q(t)$, which are then fed into USRP and processed digitally on the computer. A more detailed explanation will be presented in Section 1.4. Note that the 60GHz up and down-converters are separate devices with separate clocks and are not synchronized with each other. This is a key difference compared the existing works that use radar whereby the transmitter and receiver are on the same device and have synchronized clocks.



Options
Title: Not titled yet
Author: wisplab
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 200k

File Sink
File: ...41th_100Hz_Amp08.dat
Unbuffered: Off
Append file: Overwrite

DC Blocker
Length: 16
Long Form: True

QT GUI Frequency Sink
FFT Size: 32768
Normalize Window Power: True
Center Frequency (Hz): 0
Bandwidth (Hz): 200k

Signal Source
Sample Rate: 200k
Waveform: analog.GR_SQR_WAVE
Frequency: 4k
Amplitude: 1m
Offset: 0
Initial Phase (Radians): 100m

UHD: USRP Source
Wire Format: Complex int16
Sync: No Sync
Samp rate (Sps): 200k
Ch0: Center Freq (Hz): 0
Ch0: AGC: Default
Ch0: Gain Value: 0
Ch0: Gain Type: Normalized
Ch0: Antenna: AB

Multiply Conjugate

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 200k
Cutoff Freq: 1k
Transition Width: 100
Window: Hamming
Beta: 6.76

QT GUI Frequency Sink
FFT Size: 32768
Center Frequency (Hz): 0
Bandwidth (Hz): 200k

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 200k
Autoscale: No

Float To Complex

Rational Resampler
Interpolation: 1
Decimation: 100
Taps:
Fractional BW: 0

File Source
File: ...les/randomSquare.dat
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

UHD: USRP Sink
Wire Format: Complex int16
Sync: No Sync
Samp rate (Sps): 200k
Ch0: Center Freq (Hz): 0
Ch0: Gain Value: 0
Ch0: Antenna: AB

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 200k
Cutoff Freq: 1k
Transition Width: 100
Window: Hamming
Beta: 6.76

Rational Resampler
Interpolation: 1
Decimation: 10
Taps:
Fractional BW: 0

Multiply Const
Constant: 1m

Constant Source
Constant: 0

Multiply Const
Constant: -1m

Constellation Object
ID: const
Constellation Type: BPSK

Random Source
Minimum: 0

Constellation Modulator
Constellation: <gnu...39aaf0>

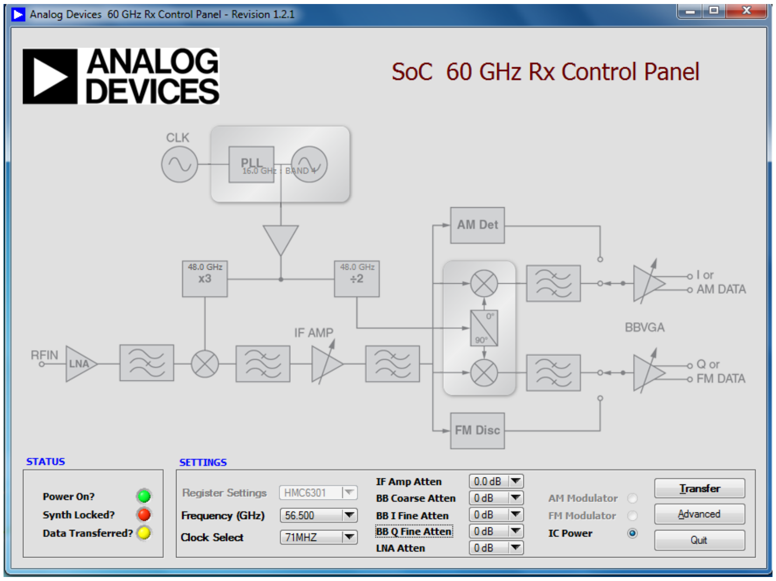
```
<<< Welcome to GNU Radio Companion 3.10.1.1 >>>

Block paths:
    /usr/share/gnuradio/grc/blocks

Loading: "/home/wisplab/Desktop/GNURadioFiles/
USPRTxRx.grc"
>>> Done
```

ID	Value
Imports	
Variables	
const	<gnuradio.digital.digital_python.constellation_bpsk object at 0x7fbb53aabdb0>
samp_rate	200000.0

Figure 1.4: GNURadio Control

Upon multiple experimental testings, we have found out that the system with Intermediate frequency (IF) attenuation of 5.2dB and RF attenuation of 15.0 dB set via software can minimize the clipping effect on the transmitter due to large signal amplitude.

In the first stage of the experiment, we used sine waves as the acoustic signal transmitted from the speaker. The USRP was initiated almost simultaneously when the sine wave started playing. It's almost impossible to perfectly synchronize both the speaker and the RF transmitter to start transmitting acoustic or radio


Analog Devices 60 GHz Rx Control Panel - Revision 1.2.1

SoC 60 GHz Rx Control Panel

STATUS

Power On? 🟩
Synth Locked? 🟥
Data Transferred? 🟨

SETTINGS

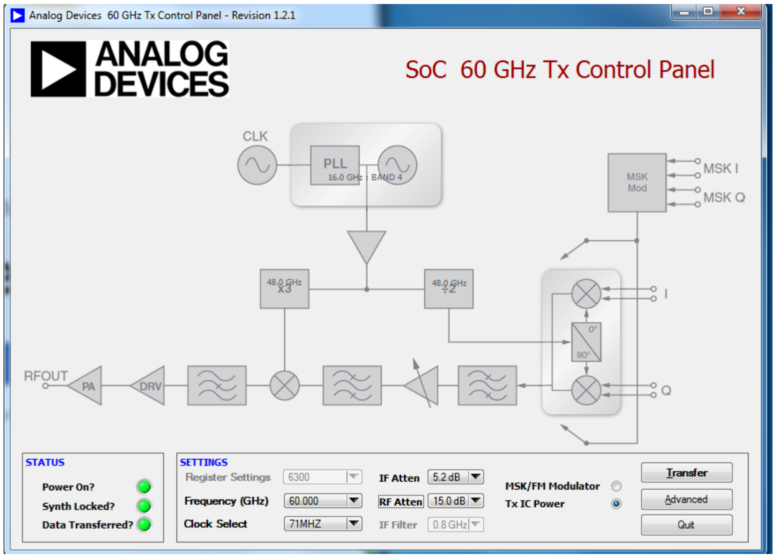
Register Settings HMC6301
Frequency (GHz) 56.500
Clock Select 71MHZ
IF Amp Atten 0.0 dB
BB Coarse Atten 0 dB
BB I Fine Atten 0 dB
BB Q Fine Atten 0 dB
LNA Atten 0 dB
○ AM Modulator
○ FM Modulator
● IC Power

Figure 1.5: Analog Device board for Receiver

frequency signal simultaneously due to the inevitable delay within the USRP hardware.

The first stage experiment result confirmed our hypothesis: we observed a peak at the frequency matching the frequency acoustic sine wave in the RF signal. See figure 1.7 which illustrates a clear peak at 220Hz in the frequency-domain plot of the RF signal when a 220Hz acoustic signal was played. The peak will be more distinct for lower frequencies, consistent with the observation that the poly-carbonate glass will oscillate with greater amplitude at lower frequencies.

The second stage of the experiment is to use the transmitted radio frequency to predict words that are played from the speaker. Details of this discussion will be present in Chapter2. The goal of this experiment is to demonstrate the possibility of passive acoustic eavesdropping on nearby conversions without being detected. In addition, radio frequency graphs are collected in this experiment to train a Machine Learning Model that could be used to predict the associated words that were



Analog Devices 60 GHz Tx Control Panel - Revision 1.2.1

SoC 60 GHz Tx Control Panel

STATUS

Power On? ●

Synth Locked? ●

Data Transferred? ●

SETTINGS

Register Settings: 6300

Frequency (GHz): 60.000

Clock Select: 71MHZ

IF Atten: 5.2 dB

RF Atten: 15.0 dB

IF Filter: 0.8 GHz

MSK/FM Modulator ○

Tx IC Power ●

Figure 1.6: Analog Device board for Transmitter

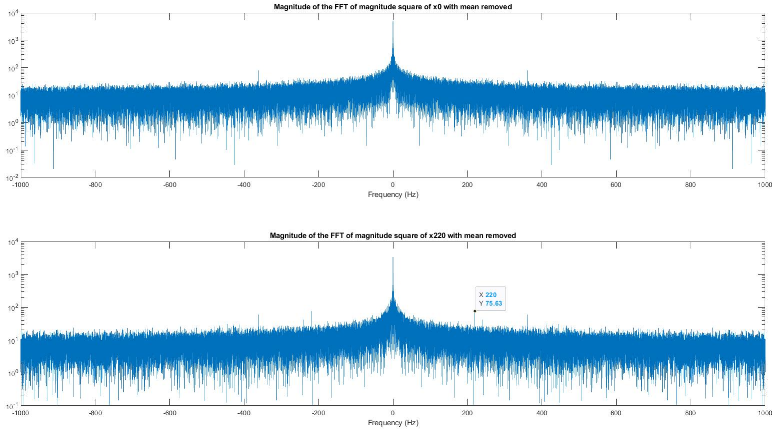


Figure 1.7: Frequency Domain Plot of Radio Frequency Signal with no signal (top graph) and 220Hz acoustic signal (bottom graph) present.

played during the collection of the RF signals.

The USRP is set to sample the signal at a rate of 200K Hz(i.e, 200K data points written into the data file per second when the USRP is operating), as this sample rate is easily translatable to different frequencies to minimize the extraneous signal that is outside the bandwith of human speech, and is a frequency supported by the USRP.

1.4 MATHEMATICAL ANALYSIS

The direct and reflected signals as received by the receiver antenna are given as follows, respectively.

$$y_d(t) = A_d m(t) \cos(2\pi f_1 t) \tag{1.2}$$

$$y_r(t) = A_r m(t) \cos(2\pi f_1 (t + d(t))) \tag{1.3}$$

Here, we assume a negligible propagation delay on the direct path and that the propagation delay on the reflected path is $d(t)$. Further, let A_d and A_r be the attenuation of the signals on the direct and reflected paths respectively. Note that in the presence of an acoustic signal, the glass vibrates, changing the delay on the reflected path $d(t)$ as a function of the acoustic signal $a(t)$. Hence, $d(t)$ contains information regarding the speech signal, and is the quantity of interest in this work. Note that the signal captured by the receiver is the sum of the direct and reflected paths. Denoting this signal by $y(t)$, we have

$$y(t) = A_d m(t) \cos(2\pi f_1 t) + A_r m(t) \cos(2\pi f_1 (t + d(t))) \tag{1.4}$$

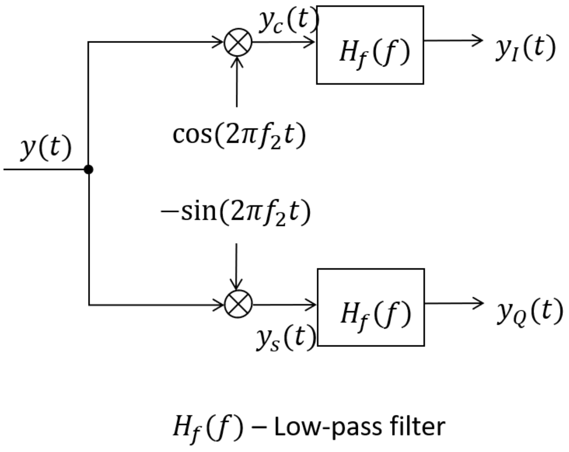


Figure 1.8: Quadrature Demodulator

The receiver applies a quadrature demodulator to the received signal, which is illustrated in the block diagram in Figure 1.8. Note that the frequency used at the receiver f_2 is supposed to equal the frequency at the transmitter f_1. However, since the transmitter and receiver are on separate devices, there is a difference between what each device thinks is a particular frequency. Hence, we assume that $f_2 \neq f_1$, but are close to each other. We note that this will not be the case for radar based systems as those systems use the same transmitter and receiver device.

The signals $y_I(t)$ and $y_Q(t)$ are sampled and converted to digital form with $y_I(t)$ being the real part and $y_Q(t)$ being the imaginary part of a digital signal. Note that this digital signal is what is made available by our Universal Software-defined Radio (USRP) system. In order to simplify the discussion, we shall describe the analysis in continuous-time (i.e. analog signal), by defining

$$w(t) = y_I(t) + jy_Q(t) \tag{1.5}$$

where $j = \sqrt{-1}$ is the unit imaginary number.

Applying the trigonometric identity

$$\cos\theta\ \cos\varphi = \frac{1}{2}\cos(\theta - \varphi) + \frac{1}{2}\cos(\theta + \varphi)$$

yields

$$\begin{aligned} y_c(t) &= \frac{1}{2}A_d m(t)\cos(2\pi(f_1 - f_2)t) + \frac{1}{2}A_d m(t)\cos(2\pi(f_1 + f_2)t) \\ &+ \frac{1}{2}A_r m(t)\cos(2\pi(f_1 - f_2)t + 2\pi f_1 d(t))) + \frac{1}{2}A_r m(t)\cos(2\pi(f_1 + f_2)t + 2\pi f_1 d(t))) \end{aligned} \tag{1.6}$$

Note that the terms $\frac{1}{2}A_d m(t)\cos(2\pi(f_1+f_2)t)$ and $\frac{1}{2}A_r m(t)\cos(2\pi(f_1+f_2)+2\pi f_1 d(t)))$ are signals of high frequency as they represent signals that are modulated by cosine waves with frequencies of $f_1 + f_2$. These signals are removed by the low-pass filter in Figure 1.8, resulting in

$$y_I(t) \approx \frac{1}{2}A_d m(t)\cos(2\pi(f_1 - f_2)t) + \frac{1}{2}A_r m(t)\cos(2\pi(f_1 - f_2)t + 2\pi f_1 d(t))) \tag{1.7}$$

The expression above is an approximation because in practice, the low-pass filter will not be ideal and thus will not be able to perfectly remove the high frequency components. However, we shall assume that the approximation is exact moving forward.

By a similar analysis and applying the trigonometric identity $\sin\theta\ \cos\varphi = \frac{1}{2}\sin(\theta + \varphi) + \frac{1}{2}\sin(\theta - \varphi)$, we find the following

$$y_Q(t) \approx \frac{1}{2}A_d m(t)\sin(2\pi(f_1 - f_2)t) + \frac{1}{2}A_r m(t)\sin(2\pi(f_1 - f_2)t + 2\pi f_1 d(t))) \tag{1.8}$$

Next, consider

$$\begin{aligned}
y_I^2(t) &= \frac{1}{4}A_d^2(m(t))^2\cos^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2 m^2(t)\cos^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \\
&\quad + \frac{1}{2}A_d A_r m^2(t)\cos(2\pi(f_1-f_2)t)\cos(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.9) \\
&= \frac{1}{4}A_d^2\cos^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\cos^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \\
&\quad + \frac{1}{2}A_d A_r\cos(2\pi(f_1-f_2)t)\cos(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.10)
\end{aligned}$$

where the last line is due to the fact that $m(t)$ is either always 1 or -1. Applying $\cos\theta\,\cos\varphi = \frac{1}{2}\cos(\theta-\varphi) + \frac{1}{2}\cos(\theta+\varphi)$ again leads to

$$\begin{aligned}
y_I^2(t) &= \frac{1}{4}A_d^2\cos^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\cos^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \\
&\quad + \frac{1}{4}A_d A_r\cos(2\pi f_1 d(t))) + \frac{1}{4}A_d A_r\cos(4\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.11)
\end{aligned}$$

Now consider

$$\begin{aligned}
y_Q^2(t) &= \frac{1}{4}A_d^2\sin^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\sin^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \\
&\quad + \frac{1}{2}A_d A_r\sin(2\pi(f_1-f_2)t)\sin(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.12)
\end{aligned}$$

Then, we apply $\sin\theta\,\sin\varphi = \frac{1}{2}\cos(\theta-\varphi) - \frac{1}{2}\cos(\theta+\varphi)$ to the previous equation to get

$$\begin{aligned}
y_Q^2(t) &= \frac{1}{4}A_d^2\sin^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\sin^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t))) \\
&\quad + \frac{1}{4}A_d A_r\cos(2\pi f_1 d(t))) - \frac{1}{4}A_d A_r\cos(4\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.13)
\end{aligned}$$

Now consider the following

$$\begin{aligned}|w(t)|^2 &= y_I^2(t) + y_Q^2(t) \qquad (1.14)\\ &= \frac{1}{4}A_d^2\cos^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\cos^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t)))\\ &\quad + \frac{1}{4}A_dA_r\cos(2\pi f_1 d(t))) + \frac{1}{4}A_dA_r\cos(4\pi(f_1-f_2)t + 2\pi f_1 d(t)))\\ &\quad + \frac{1}{4}A_d^2\sin^2(2\pi(f_1-f_2)t) + \frac{1}{4}A_r^2\sin^2(2\pi(f_1-f_2)t + 2\pi f_1 d(t)))\\ &\quad + \frac{1}{4}A_dA_r\cos(2\pi f_1 d(t))) - \frac{1}{4}A_dA_r\cos(4\pi(f_1-f_2)t + 2\pi f_1 d(t))) \qquad (1.15)\\ &= \frac{1}{4}A_d^2 + \frac{1}{4}A_r^2 + \frac{1}{2}A_dA_r\cos(2\pi f_1 d(t))) \qquad (1.16)\end{aligned}$$

where we have applied the identity $\sin^2\theta + \cos^2\theta = 1$.

Further

$$|w(t)| = \sqrt{\frac{1}{4}A_d^2 + \frac{1}{4}A_r^2 + \frac{1}{2}A_dA_r\cos(2\pi f_1 d(t)))} \qquad (1.17)$$

Hence, $|w(t)|$ is a function of $d(t)$, which is in turn a function of the acoustic signal $a(t)$. Note that $|w(t)|$ is not a function of the data that is transmitted($m(t)$), and impacts of discrepancy between f_1 and f_2, has been removed by the processing. We additionally highlight that even though $d(t)$ is extremely small, it is multiplied by f_1 which is 6×10^{10}. Hence, an extremely small change in $d(t)$ can result in a small, but appreciable change in $|w(t)|$. Hence, the quantity $|w(t)|$ can be used as features to detect the speech signal.

To get a sense of the range of values of $d(t)$, we consider that $d(t)$ is the difference in propagation delay between the direct path between the transmitter and receiver and the reflected path. The difference in distance between the paths is on the order of 100 μms Hu et al. (2022b). The delay is given by the distance divided by the speed of light

$$100 \times 10^{-6} \times \frac{1}{3} \times 10^{-8} = 3.3 \times 10^{-13}\ s \tag{1.18}$$

When multiplied by $f_1 = 60 \times 10^9$, we have

$$f_1 d(t) = 3.3 \times 60 \times 10^9 \approx 0.02 \tag{1.19}$$

While this is a small number, as shown subsequently in this paper, we are able to detect these speech signals.

CHAPTER 2
Signal Identification

2.1 INITIAL DATA PROCESSING

The goal of this section is to provide a more detailed explanation of the data collection and analysis process in the Stage two of the Experiment presented in (section 1.4)

For simplicity, we have only used spoken digits to prove our claim that such a passive eavesdropping attack could be possible. We have selected the digits from the MNIST audio dataset, and standardized each audio digit file to repeat only once. We have combined the standardized audio digit file, each with a fixed length, into one single file. We collected approximately 80 such files, consisting approximately 9600 words in total.

As described in the last chapter(1.3), we cannot assume] synchronization between the start of the acoustic signal and the received RF signal. Using the result from the previous stage of the experiment reported in chapter 1.3, the system has a sensitive response to low-frequency sine waves. Hence, we created a 0.5 second long 100 Hz sine wave followed by a 1 second break before playing the digits from speaker, to synchronize the received signal with the transmitted signal.

The sine wave in Figure 1.7 suggested that a distinctive peak appears at 220Hz frequency. We use the cross-correlation technique to identify the start time of the received signal], by cross correlating the received RF signal with a reference 100Hz sine wave. Using cross correlation provides an accurate estimation of the start point when the USRP started receiving the transmitted signal. The data samples are collected at a frequency of 200kHz by the USRP. Since human speech is typically limited to a few kHz, we have reduced the collected sample by a factor of 20, resulting in a signal sampled at 10kHz. Note that since speech signals are typically a few kHz in bandwidth, and the *Nyquist frequency* of a 10kHz sampled signal is 5kHz, most of effects of the speech signal will be preserved in the sampled radio signal.

We used Matlab to perform analysis on the collected data sample. Figure 2.1 demonstrates the plot of one segment of the received RF signal after the processing. For an audio file consisting of 120 words, 120 distinctive peaks appear on the radio frequency time domain plot. A partial plot of the radio frequency plot is attached here to illustrate.

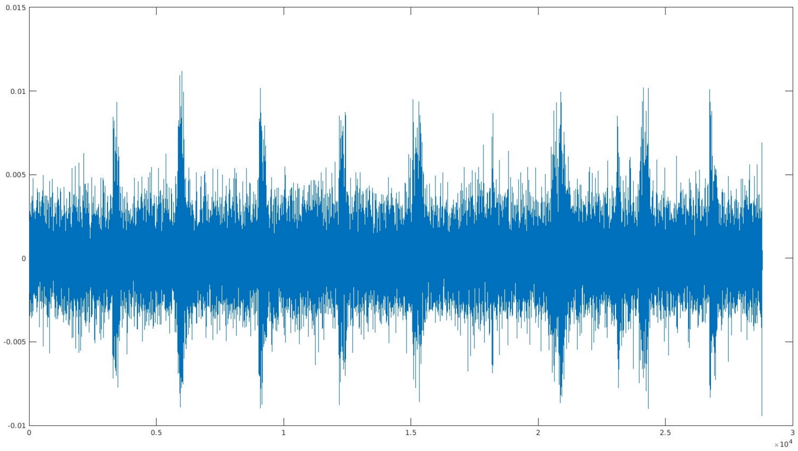


Figure 2.1: Selected part of the collected radio frequency signal

As presented in figure 2.1, this audio sample consisted of 10 words corresponding to 10 peaks reflected in the graph.

Direct classification performed on such data did not obtain a very high accuracy. Possible explanations for the low accuracy will be presented in the next section.

2.2 DATA REPRESENTATION FOR THE MACHINE LEARNING MODEL

The given task is a supervised learning task: we input a processed radio frequency signal of a word, and the output is a label associated with the word. The audio files in our work consist of 120 words each, which are standardized in length. So, we can segment the processed RF signal (after synchronization) to match each word in the audio file.

Upon analysis with the original time domain radio signal, using CNN and other feed-forward neural networks, the accuracy is approximately 20%. There are a few reasons we believe that cause the low accuracy:

- The transmitter 60GHz board is designed for evaluation purposes for short experiments. however, under our setting, we have used it to transmit and receive the signal for longer periods of time than the board was originally designed for. Such use could potentially cause the hardware to be unreliable, i.e, the signal transmitted/received could be inaccurate.

- When choosing words from the MNIST audio dataset, we realized that certain words are low in volume which are difficult to detect. To increase the volume of such words, we have normalized the power of all the audio signals. While normalized audio signals have higher volume, the noise present in the signal will also be amplified. Hence, the increased volume may not necessarily result in significantly improved accuracy.

Before training, we plotted approximately 10 randomly selected collected radio frequency graphs corresponding to the audio digit “seven” collected to examine the features of the signal. Approximately seven of the samples have a relatively distinct peak in the time domain graph, and the remaining plots do nor have distinctive peaks that help identify the words. This sample reflects the fact that some of the audio signals may not appear distinctly in the time domain signals.

The other approach to the preparing the dataset is to convert the collected radio frequency signals to spectrograms If the reader is not familiar with the idea of spectrograms, please see Appendix A for an overview of spectrograms before proceeding. There are a few benefits of using spectrogram representations:

- Representation which includes time and frequency: Spectrograms provide a visual representation of a signal's time-frequency content, which is useful for analyzing speech signals. In contrast, a raw RF graph shows the signal's amplitude or power as a function of time, which can be more challenging to interpret in terms of speech content.

- Better Visualization: Spectrograms allow for better visualization of speech content by representing the signal in terms of its frequency components. They make it easier to identify and analyze different phonemes, pitch, and other speech characteristics compared to directly observing an RF time-domain signal. This is particularly beneficial in our setting because while each audio digit is limited to the same length, the audio may not start at the beginning of the assigned period. Such a sample would result in a peak as seen in 2.1, at a different position in segment.

- Enhanced Signal Processing: Spectrograms enable the use of advanced signal processing techniques such as filtering, feature extraction, and noise reduction. These methods can improve speech recognition performance by making it easier to identify and differentiate words and phonemes.

- Easier Pattern Recognition: Spectrograms can help in the identification of re-

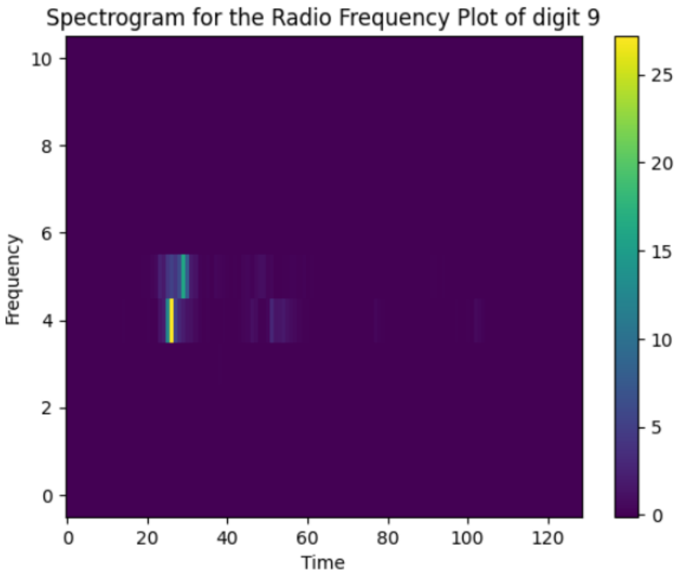


Figure 2.2: Sample spectrogram corresponding to a word collected in the experiment.

curring patterns in speech signals, such as formants and harmonics. These patterns are crucial for understanding and recognizing different speech sounds. In contrast, it can be more challenging to identify such patterns in a time-domain RF graph.

Figure 2.2 is an example of the spectrogram of the radio frequency data collected. As suggested, the presence of word is more easy to see, hence train.

2.3 MODEL CONSTRUCTION

The original attempt on developing the model focuses training on the time domain signal, but as presented in the last section, the time domain signal which only shows amplitude and time, is not as easily trained to. Hence we used a spectrogram approach to help achieve a higher accuracy. The model design for the spectrogram classification has been carefully constructed to maximize accu-

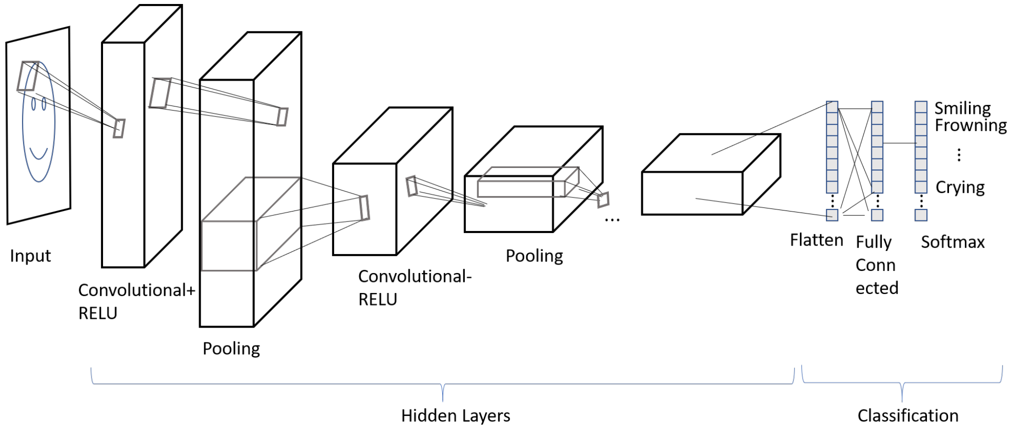


Figure 2.3: Example CNN Architecture model

racy while minimizing overfitting. The architecture consists of a combination of convolutional layers and fully connected layers, each with specific attributes and configurations designed to optimize the model's performance.

The four convolutional layers are the backbone of the model, as they are responsible for extracting the most relevant features from the input spectrograms. These layers are followed by batch normalization, which helps to stabilize and accelerate the training process by reducing internal covariate shift. The use of the ReLU activation function ensures that the model remains computationally efficient, as it introduces non-linearity while reducing the likelihood of vanishing gradients. Furthermore, max pooling operations are employed to reduce the spatial dimensions of the feature maps, thereby decreasing the computational complexity and aiding in the extraction of robust features. Figure 2.3 is an example of CNN Architecture. The model we deployed consisted of 4 hidden Convolutional Layers.

The number of filters for each CNN layer increases with depth, starting at 64 and progressing to 128, 256, and finally 256 again. This configuration allows the model to capture increasingly complex patterns and details from the input data. The kernel size is set to 4 for the first CNN layer, while the remaining layers utilize a kernel size of 2. This design choice enables the initial layer to capture a broader range of features before subsequent layers focus on finer details. A model attached in the figure will demonstrate the drawing.

Following the convolutional layers are six fully connected layers(the classification layers in 2.3), which serve to combine the extracted features and make the final classification decision. The output sizes of these layers decrease progressively from 4096 to 10, reflecting the narrowing focus of the model towards the final classification task. The last layer has 10 output nodes, corresponding to the 10 different digits being classified.

Dropout operations are included after each fully connected layer, except for the output layer, to mitigate overfitting. Overfitting occurs when a model becomes too specialized to the training data, impairing its ability to generalize to new, unseen data. By applying dropout with varying probabilities (0.4, 0.4, 0.6, 0.4, and 0.4), the model becomes more robust, as it is forced to learn multiple representations of the data. This design choice was made in response to observations of overfitting during the training process.

We are still exploring possibilities with using pretrained network such as the ResNet18, DeepSpeech Model from Pytorch. However, we refrained from using the pretrained model as the model goal was designed for images or acoustic signal identification, while the dataset we collected, does not fall into either category.

The validation accuracy at its highest, approached 30%. This accuracy is not as high as we desire. However there are a few approaches that we are considering to make the model more accurate. One drawback with the current dataset is that over $\frac{2}{3}$ of the dataset was collected under when the hardware was not fully reliable, and has suffered from performance degradation.

Even through the accuracy is relatively low, it is significantly higher than chance (10%), which indicates that an attacker can obtain some information regarding the conversation that is taking place. Therefore, approaches to mitigate such attacks should be explored, such as varying the amplitude of the wireless communication signal randomly by a small amount, in such a manner that the impact on the communications signal is not significant, yet it provides protection to users from acoustic eavesdropping. We speculate that a whole direction of research focused on systematically reducing the risk of such attacks can be created.

2.4 FUTURE WORK

This system has only considered capturing reflected signals. We hypothesize that signals transmitted through a vibrating surface such as a window will undergo refraction that changes over time in a way that is related to the acoustic signal causing the vibration. Such a hypothesis has not really been tested by other researchers. As an extension of our project, we would like to see whether it is possible to use the refraction of the signal through a vibrating glass window to demonstrate the possibility of a similar attack.

While the system accuracy has not approached a very desirable result, with a test accuracy of 30%, it still remains possible to test the data with some other pre-trained model, such as DeepSpeech. We refrained from using a pre-trained model on speech signal as our data is not an acoustic signal, but if the data collection is performed using a continuous source of acoustic signals, such as audio books, we may be able to achieve a relatively high rate of classification.

2.5 CONCLUSIONS

We have demonstrated an eavesdropping hazard caused by mmWave wireless communication signals. While the model did not achieve an accuracy as high as we expected from the radio frequency signal, the accuracy is significantly higher than chance. Therefore, we have demonstrated that an eavesdropper could obtain acoustic information from wireless communications signals, unlike previous literature that either use radar waveforms, or communication wave forms under a very specific setting. Improvements in the system hardware as well as potentially more good quality data will need to be collected, as deep learning in general requires large dataset to train on, to further improve the system.

Given the extensive use of wireless communication in the world, we believe that this work highlights the need to consider security and risks posed by wireless communication waveform signals, beyond the traditional security risks of malicious attempts at intercepting communication data.

APPENDIX A
Additional Concepts

A.1 SPECTROGRAMS OVERVIEW

Spectrogram is a visual representation of the frequency spectrum of a signal as it evolves over time. It is commonly used in the analysis of audio signals, such as speech, music, or environmental sounds. Spectrograms display the frequencies present in a signal on the vertical axis, time on the horizontal axis (or vice-versa), and the amplitude or energy of each frequency component is represented by color or intensity. The figure 2.2 presents an example of spectrogram that is deployed in the experiment.

Spectrograms can be generated by following the steps in the following:

1. Divide the signal into overlapping segments or frames: The input signal is divided into short, overlapping segments or frames, usually using a windowing function such as the Hanning or Hamming window. This is done to isolate small portions of the signal for analysis and to minimize the impact of discontinuities at the edges of each frame.

2. Compute the Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) for each frame: The DFT or its computationally efficient variant, the FFT, is applied to each frame to convert the signal from the time domain to the frequency domain. This transformation produces a set of complex numbers representing the amplitude and phase information for each frequency component in the frame.

3. Calculate the magnitude spectrum: The magnitude spectrum is derived by

taking the magnitude of each complex number in the DFT or FFT result. This represents the amplitude or energy of each frequency component in the frame.

4. Convert the magnitude spectrum to a logarithmic scale (optional): Often, the magnitude spectrum is converted to a logarithmic scale (such as decibels) to better match the human perception of sound and to compress the wide dynamic range of the amplitude values.

5. Construct the spectrogram: The magnitude spectra (or their logarithmic representations) for all frames are combined into a 2D matrix or image, with time on the horizontal axis, frequency on the vertical axis, and the amplitude or energy of each frequency component represented by color or intensity.

BIBLIOGRAPHY

Hu, P., Li, W., Spolaor, R., & Cheng, X. (2022a). mmecho: A mmwave-based acoustic eavesdropping method. In *2023 IEEE Symposium on Security and Privacy (SP)*, (pp. 836–852). IEEE Computer Society.

Hu, P., Ma, Y., Santhalingam, P. S., Pathak, P. H., & Cheng, X. (2022b). Milliear: Millimeter-wave acoustic eavesdropping with unconstrained vocabulary. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, (pp. 11–20). IEEE.

R Core Team (2017). R: A Language and Environment for Statistical Computing. https://www.r-project.org/.

Sharpsteen, C., & Bracken, C. (2016). *tikzDevice: R Graphics Output in LaTeX Format*. https://cran.r-project.org/package=tikzDevice.

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., & Despouy, P. (2017). *plotly: Create Interactive Web Graphics via 'plotly.js'*. https://cran.r-project.org/package=plotly.

University of Florida (2000). Useful Pharmacokinetic Equations. http://pharmacy.ufl.edu/files/2013/01/5127-28-equations.pdf (Accessed:2017-01-25).

Wei, T., Wang, S., Zhou, A., & Zhang, X. (2015). Acoustic eavesdropping through wireless vibrometry. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, (pp. 130–141).

Wickham, H. (2007). Reshaping data with the reshape package. *J Stat Softw*, *21*(12).

Zheng, K., Wang, C., Shu, Z., & Lin, F. (2022). Speech acquisition and recovery based on piezoelectric effect in the mmwave band. In *2022 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, (pp. 174–176). IEEE.