上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Intelligent IOT Project Report

Project Name: Canceling Inaudible Voice Commands Against Voice Control Systems

Teammate: Hanwen Xu、Xingyuan Chen、Xiuyuan Chen、Yijie Gao、Liujia Yang

Group Work Sheet

Name	Student ID	Work Item	E-mail
Hanwen Xu	519030910405	Ultrasound Attack Realization, Ultrasound Defense Realization, Frequency Hopping Attack Realization, CNN Defense Realization, Write Report, Middle Presentation PPT, Pre-defense Presentation PPT, Pre-oral defense, Final Presentation PPT, Final Oral Defense	dctnorin@sjtu.edu.cn
Xingyuan Chen	519030910414	Ultrasound Attack Simulation, Ultrasound Defense Simulation, Ultrasound Attack Realization, Ultrasound Defense Realization, Frequency Hopping Attack Realization, CNN Defense Realization, Write Report, Final Presentation PPT, Final Oral Defense	February25th@sjtu.edu.cn
Xiuyuan Chen	519030910411	Ultrasound Attack Simulation, Ultrasound Defense Simulation, Ultrasound Defense Realization, CNN Defense Realization, Frequency Hopping Attack Realization, Final Presentation PPT, Write Report	khhuiyh@sjtu.edu.cn
Yijie Gao	519030910392	Frequency Hopping Attack Simulation, Ultrasound Defense Realization, CNN Defense Realization, Write report	gaoyijie@sjtu.edu.cn
Liujia Yang	519030910406	Frequency Hopping Attack Simulation, CNN Defense Realization, Write report	yangliujia1008@sjtu.edu.cn

Abstract

This project aims to achieve the goal that attack down voice assistant on the phone using ultrasound and defense the phone avoiding attacking by ultrasound. Moreover, our group proposed a new method to defense the ultrasound attack by Convolution Neural Network. We generate ultrasound attack signal by modulating the attack command to high frequency with AM modulation, then broadcast this signal through ultrasonic probe to the phone. The voice assistant on the phone will detect the attack command and do as the command asks. To defense this attack, our group set up a system by sending guard signal constantly to the phone through ultrasonic probe. After that we apply LMS algorithm to adapt the filter parameters to eliminate the original attack command. To attack the phone with this guard system, we realize a frequent-hopping attack system to break through the guard system. Our group propose a new guard system by utilizing MFCC feature of audio and using a light-weighted CNN to extract latent feature to discriminate difference between attacking signal and original human sound. This paper divide into three parts. First of all, we describe the basic theory of attacking system and defensing system. Next, we show our progress and results in detail. At last, we discuss the problem we met in experiment and give a brief idea of what to be done in the future.

Key words: Ultrasound Attack, LMS, frequent-hopping, MFCC, CNN

Content

1. Background Introduction

With the development of artificial intelligence and natural language processing technology, the sound has become an important way for human-computer interaction, and the voice in smartphones has a broader application prospect. Voice assistants can provide convenient services for humans in many scenarios, people no longer need to handle mobile phones with both hands, but through the way to speak information, thereby using the corresponding functions of mobile phones. It can be seen that the existence of voice assistants has greatly improved the convenience of people's lives.

However, the voice assistant also has a corresponding safety hazard while making us live. In the top of the International Information Security World, California, the "Network and Distributable System Security Conference", disclosed an ultrasonic attack method called "surfing attack", and the attacker can transmit the table or other solid contact of the phone. The ultrasonic wave carrying the command information is to achieve the effect of the control voice assistant.

In fact, more than "surfing attack", when the ultrasound signal carrying information enters the microphone, the microphone's nonlinearity makes the ultrasonic signal self-consuming, thereby converting the high frequency signal to a low frequency signal, and retains the original information. Further, it is identified by the voice assistant. The entire process is not heard, which has a great threat to the user's privacy security. The goal of this experiment is to achieve ultrasonic signal attacks, and design the corresponding defense programs, and finally implement frequency hopping attacks on the basis of defense.

2. Principle Description

In this section, we will introduce the three principles: attack, defense and frequency hopping involved in the experiment.

2.1 Introduction to Attack Principle

2.1.1 Microphone

Microphone is an energy converter that converts sound signals into electrical signals。In the 20th century, microphones developed from resistance conversion to inductive and capacitive conversion. A large number of new microphone technologies gradually developed, including aluminum tape, moving coil and other microphones, as well as capacitive microphones and electret microphones widely used at present.

2.1.2 Mobile Microphone

Mobile phone microphone is mainly composed of diaphragm and amplification circuit. When the sound enters the microphone, the vibration of the sound will be transmitted to the diaphragm of the microphone, pushing the magnet inside to form a changing current, realizing the conversion from sound signal to electrical signal.

The amplifier circuit is composed of complex circuit structure and a large number of electrical components. We simplify it and divide it into three parts: amplifier, low-pass filter and analog-to-digital converter. Because the strength of the previously converted electrical signal is very weak, it is necessary to amplify the signal through an amplifier, and then filter the part that cannot be heard by human ears through a low-pass filter. Finally, an analog-to-digital converter is used to convert the analog signal into digital signal. As for how to deal with this digital signal, that is the working principle of the speaker, which is not within the scope of this experiment.

2.1.3 Nonlinear Distortion of Amplifier

The nonlinearity of circuit means that the ratio of output to input signal is a variable, that is, the relationship between output and input is not a straight line with fixed slope. For a simple single-stage amplifier, its nonlinearity is related to the size of the input signal and the DC bias point.

When the input is a sinusoidal signal, the output waveform is not an ideal sinusoidal signal due to the nonlinearity of the amplifier, resulting in distortion. This distortion caused by the nonlinearity of the amplifier parameters is called nonlinear distortion, and the relationship between input and output can be expressed by Taylor series expansion.

$$y(t) = a_1 x(t) + a_2 x(t)^2 + a_3 x(t)^3 + \cdots$$

When the input frequency is very small, the nonlinearity of the amplifier can be ignored, the output and input can be regarded as a linear relationship, and the gain is a_1. As the input frequency increases, the nonlinearity of the amplifier must be taken into account. At this time, the expression needs to include not only the primary term of the input, but also the quadratic term or even the higher-order term.

2.1.4 Attack Voice Assistant with Ultrasonic Signal

Using the nonlinearity of the amplifier in the microphone described above, the following steps can be carried out in sequence to realize the attack of ultrasonic signal on voice assistant:

Record a normal human voice signal $v(t)$

Am modulate $v(t)$ to modulate the signal to high frequency band (beyond the frequency range that human ears can receive). This step needs to consider the frequency response characteristics of the microphone. If the modulation frequency band is too high, the signal will be seriously attenuated.

$$F\big((v(t)+1)\cos(2\pi f_1 t)\big) = V(f-f_1) + \delta(f-f_1)$$

The high-frequency signal passes through the amplifier. According to the nonlinear principle of the amplifier, this process is equivalent to self-convolution of the modulated high-frequency signal in the frequency domain.

$$[V(f-f_1) + \delta(f-f_1)] * [V(f-f_1) + \delta(f-f_1)] = 2V(f) + \delta(f) + \cdots$$

The convoluted signal is passed through a low-pass filter and only the frequency band below 20kHz is reserved: $2V(f) + \delta(f)$

The signal is transformed from frequency domain to time domain by inverse Fourier

transform to obtain the signal recognized by mobile phone

$$attack(t) = F^{-1}\big(2V(f) + \delta(f)\big) = 2v(t) + 1$$

The attack signal retains the information in the original signal $v(t)$, so that the voice assistant can recognize the semantic information in the original signal by receiving the attack signal, so as to realize the whole attack process. The third step to the fifth step is equivalent to the signal processing process of the mobile phone microphone. The human ear cannot hear the relevant sound during the attack.

2.2 Introduction to Defense Principle

2.2.1 Ultrasonic attack review

The key step to the success of the attack is that the AM modulated signal will be self convoluted with itself through the microphone. This process will move the high-frequency signal to the low-frequency band, and then convert the obtained low-frequency signal into a time-domain signal by inverse Fourier transform. It can be seen that this low-frequency signal is the key to complete the attack, so how to eliminate it is the core of the next defense process.

2.2.2 Defense Principle

While transmitting the AM modulated signal to the microphone, the mobile phone terminal continues to send a defense signal. We choose the defense signal as a sinusoidal signal with fixed frequency. The modulated high-frequency signal and sinusoidal signal enter the microphone at the same time. According to the nonlinear principle, the two signals will convolute with each other. Instead of moving the high-frequency signal to low frequency as before, they will be transferred to a special frequency band. The schematic diagram is as the figure below:

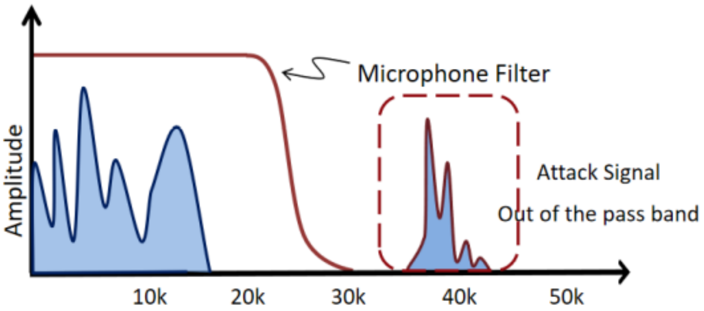


Fig.1 Defense Principle

According to the figure, we can see that the frequency band lower than 20kHz is the result of self-convolution of high-frequency signals, including attack signals. We must filter this section of signals. The signal near 40kHz is an attack signal formed by the convolution of high-frequency signal and sinusoidal signal. We need to do more processing on this signal.

The specific method is to make the whole signal undergo a high pass filter, so as to extract the signal in the frequency band where the high-frequency signal and sinusoidal signal convolute each other (near 40KHz), and LMS adaptive filter this signal to the baseband attack signal, so as to finally eliminate the attack signal.

2.2.3 Adaptive Filtering: LMS (Least Mean Square)

The input signal $x(n)$ generates the output signal $y(n)$ after passing through the parameter adjustable digital filter. It is compared with the expected signal $d(n)$ to form the error signal $e(n)$. The filter parameters are adjusted through the adaptive algorithm to minimize the mean square value of $e(n)$.

Adaptive filtering can automatically adjust the filter parameters at the current time by using the results of the filter parameters obtained at the previous time, so as to adapt to the unknown or time-varying statistical characteristics of signal and noise, so as to realize the optimal filtering. In essence, it is a Wiener filter that can adjust its transmission characteristics to achieve the best. Its detailed model is shown in the figure below:

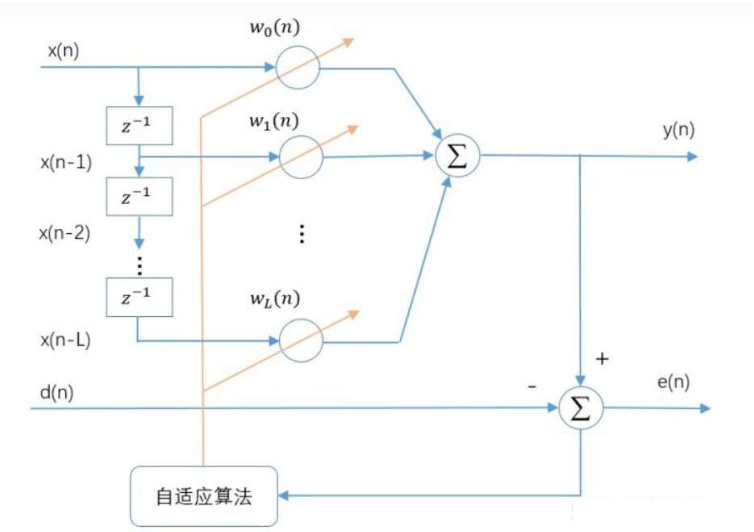


Fig.2 LMS model

According to the model diagram, the LMS process is as follows:

① Initialize filter coefficient $W(0)$and model input $X(0)$

② For each new input sample $x(n)$, calculate the output signal $y(n)$

③ Using the expected output $d(n)$, the error signal $e(n)$ is calculated to obtain the gradient ∇:

$$\nabla(n) = -2e(n)X(n)$$

④ Update filter coefficient $W(n)$ using LMS update formula:

$$W(n+1) = W(n) + 2\mu X(n)e(n)$$

⑤ Return to step ② until the end, the output sequence and error sequence can be obtained.

2.3 Introduction to Frequency Hopping Principle

2.3.1 LMS Review

Through the previous introduction, it can be seen that LMS adaptive filtering is the key to defense. In this algorithm, the adaptive filter adopts the gradient descent method to realize the convergence of the loss function. Record the number of iterations as μ, λ_{max}

represents the maximum eigenvalue of the autocorrelation matrix of the input signal. In order to converge, the two shall meet:

$$0 < \mu < \frac{1}{\lambda_{max}}$$

2.3.2 Frequency Hopping Attack

According to the above inequality, we find that once it does not hold, it will lead to unsuccessful convergence and defense failure. Therefore, we propose frequency hopping attack, using single chip microcomputer to segment the attack signal according to the frequency, and then am modulate each segment, so as to make the carrier frequency of the attack signal jump between multiple values. This obviously increases the number of iterations. when μ does not meet the above inequality, the adaptive filter will not converge, the defense will fail, and finally realize the frequency hopping attack.

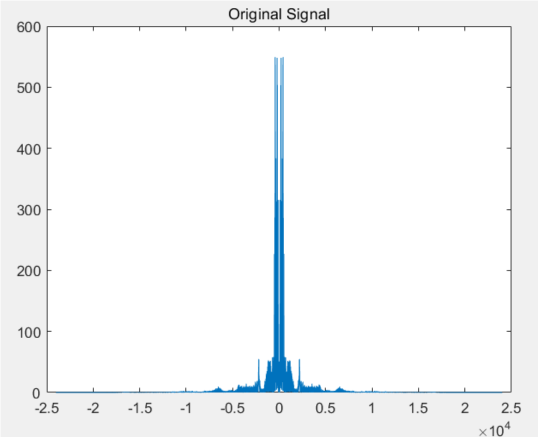
3. MATLAB Simulation

In this section, we will show the MATLAB simulation experiments of attack, defense and frequency hopping.

3.1 Attack Simulation Experiment

According to the attack principle, the original signal needs to be AM modulated to the high frequency band. At this time, the frequency response characteristics of the microphone mentioned earlier need to be considered: when the microphone receives the signal, it will attenuate to a certain extent; In the high frequency band, the attenuation increases with the increase of signal frequency. Therefore, if the attack signal frequency is too high, it will lead to excessive attenuation and affect the attack effect. Therefore, the frequency of AM modulation signal should not be too high.

In the simulation experiment, the modulation frequency is selected as 24khz, and the experimental results are as follows:



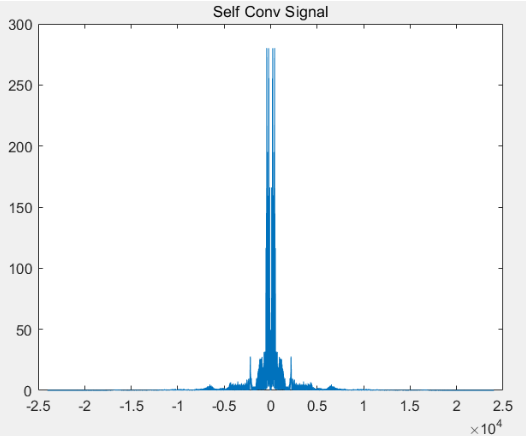


Fig.3 Attack Simulation Experiment

The recorded human voice is taken as the original signal (its frequency domain is shown on the left of Fig. 3-1), and the attack signal is obtained through: a) 24khz AM modulation; b) Frequency domain self-convolution of analog microphone characteristics; c) Low pass filtering. Finally, the self-convolution signal is obtained (its frequency domain is shown on the right of Fig. 3-1). It is not difficult to find that after self-convolution, the attack signal is basically consistent with the original signal, which can explain the effectiveness of the attack.

3.2 Defense Simulation Experiment

According to the above and the introduction of the paper, the frequency of attack signal should not be too high, and 20kHz ~ 45khz is more appropriate. Therefore, two sinusoidal signals with frequencies of 20kHz and 40KHz can be selected as the defense signal. After the amplifier, the defense signal and attack signal can be convoluted with each other to limit the convolution signal containing the attack signal to 10kHz ~ 20kHz. By detecting whether there is a high-power signal in this frequency band, we can judge whether there is an attack signal and carry out subsequent removal. Here, in order to simplify the operation and conform to the subsequent system hardware implementation, we have selected 10kHz defense signal and 23khz attack signal. The frequency domain diagram and time domain diagram are as follows:

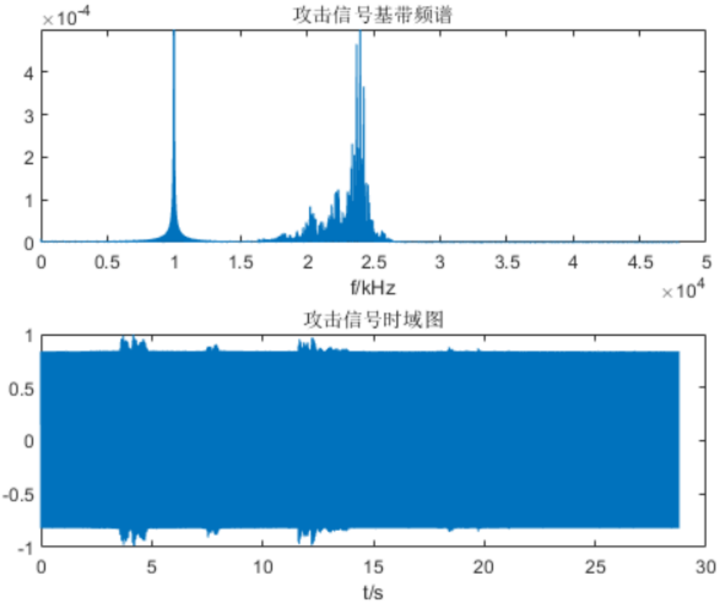


Fig.4 10kHz Defense Signal and 23khz Attack Signal

In the frequency domain diagram, we can clearly see the respective frequency bands of our defense signal and attack signal. Next, we simulate the nonlinear process of sound passing through the microphone, that is, the results of self convolution of two signals and mutual convolution of two signals are as follows:

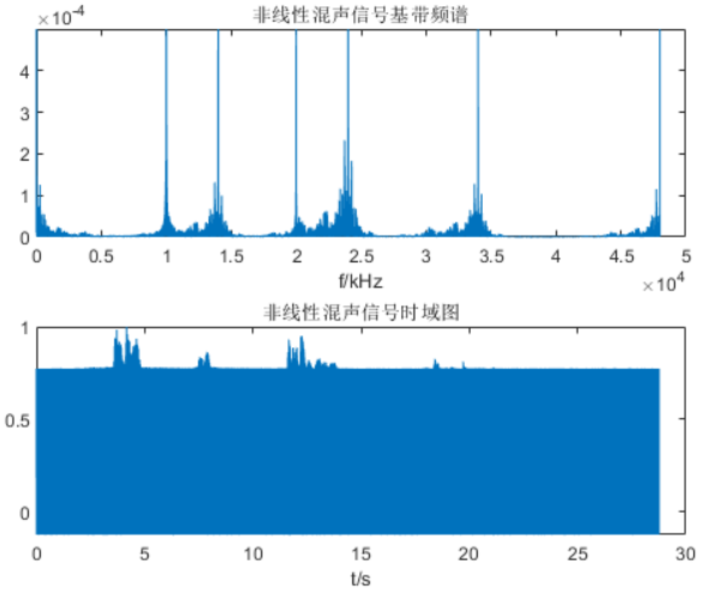


Fig.5 Nonlinear Mixed Signal

From the simulation results, the baseband part (< 10kHz) is the result of the self-convolution of the attack signal and the core instruction part of the attack mobile phone voice assistant. At the same time, a high peak appears at the frequency of 13khz, which is the signal formed after the convolution of the defense signal and the attack signal, and it is also the subsequent breakthrough. At this time, it is necessary to self convolute the 13khz signal, extract the attack signal to be filtered, leave the baseband attack signal, and carry out low-pass filtering on the whole signal. Finally, to remove the attack signal, it is necessary to operate on the baseband signal. The extracted attack signal and baseband signal are as follows:

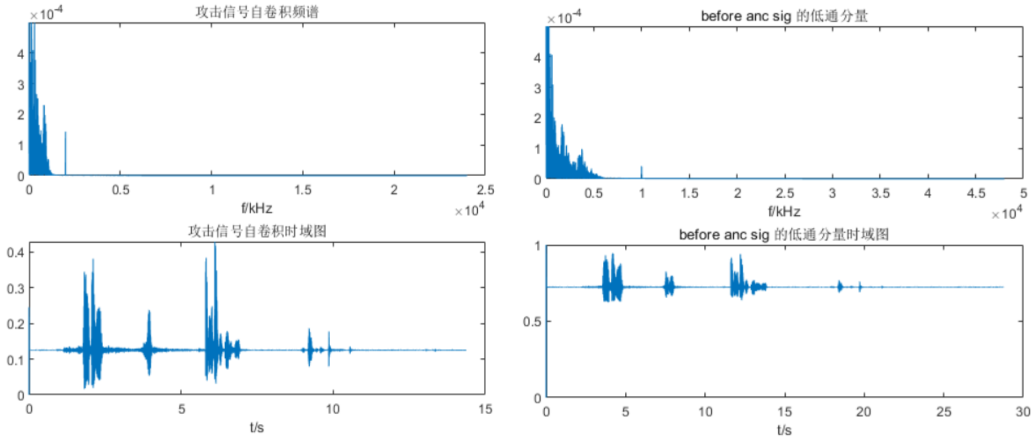


Fig.6 Attack Signal and Baseband Signal

The overall shape of the extracted attack signal is roughly similar to that of the baseband attack signal. In the time domain, each semantic segment is also retained to a certain extent, but there will be corresponding power loss in the extraction process, resulting in a small overall power of the attack signal, which will affect our subsequent removal to a certain extent. After the attack signal is obtained, it is removed by adaptive filtering. The results are as follows:

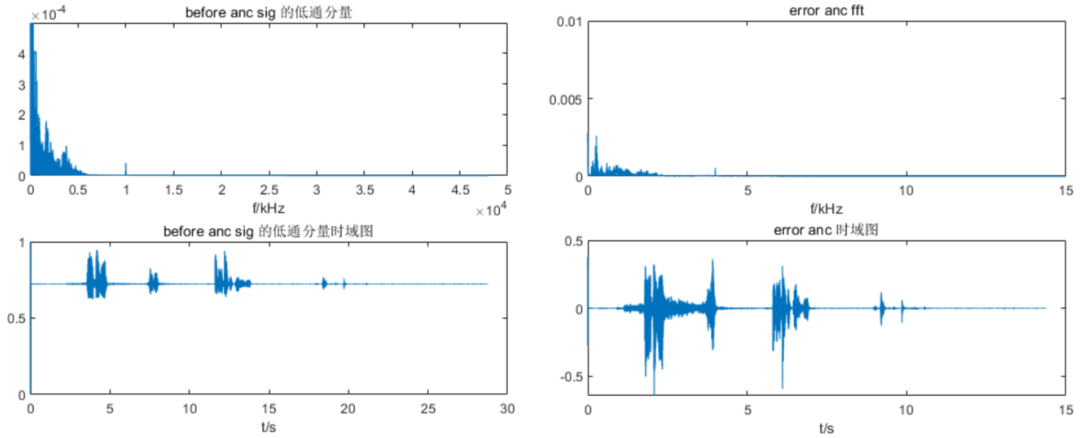


Fig.7 Signal Before and After Adaptive Filtering

On the left is the signal before adaptive filtering, and on the right is the signal after filtering. It can be found that the high-power part of the original base band attack signal in the right figure has been removed, and this part contains a lot of semantics; Reflected in the time domain diagram, the semantic of attack instructions has been destroyed. Playing the signal represented in the right figure, it is found that the attack command has been almost inaudible, indicating that the simulation effect of defense is good.

3.3 Frequency Hopping Simulation Experiment

According to the principle of frequency hopping attack, in AM modulation, the original signal needs to be periodically divided into several segments in time domain, and each segment is modulated with different frequencies. In the simulation experiment, the time domain segment length is 133ms, and the modulation signal frequencies are 24khz, 32kHz and 40KHz respectively. The time domain and frequency domain images of frequency hopping attack signal are as follows:

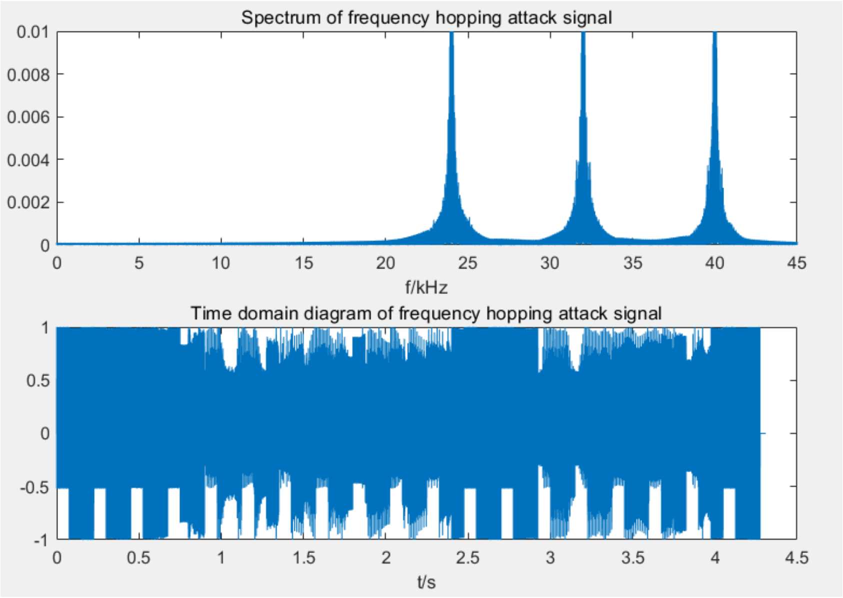


Fig.8 Frequency Hopping Simulation Experiment

The three center frequencies in the frequency domain diagram are the frequencies of three groups of different AM modulation; The sawtooth amplitude change in the time domain diagram is the pulse caused by switching frequency, which reflects the result of different frequency modulation in time domain.

4. Hardware System Construction and Display

Experiment equipment:

ultrasonic probes, low frequency power amplifiers, DuPont lines, computers, headphone to Dupont cable and power converter.

Experimental Situation:

Experimental location: quiet environment is sufficient

Attack target: OPPO A9 (Xiaobu voice assistant)

Attack command: open WeChat

4.1. Attack System

The attack system adopts the attack principle explained in Section 2 for the ultrasonic signal attack on cell phone voice assistant, which is implemented by hardware based on MATLAB simulation. In this system, the equipment to be used are: ultrasound probe, low frequency power amplifier, headphone to duplex cable, power converter, and computer.

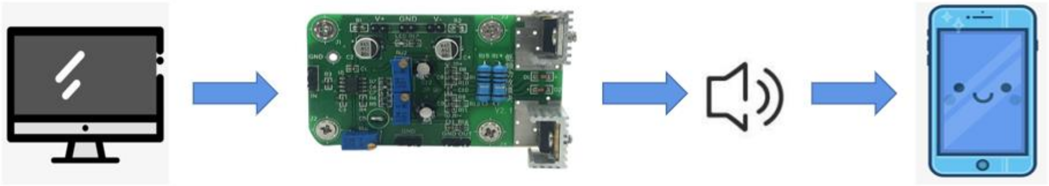


Fig.9 Attack system implementation process:

① Use MATLAB software on the computer to modulate the voice signal. First, input the voice signal, go through low-pass filtering, and use AM to modulate it to the ultrasonic frequency band, and then pass the band-pass filtering to play the sound.

② Use a USB headphone converter and headphone to Dupont cable to connect the sound signal played by the computer to a low frequency power amplifier. The low-frequency amplifier is rated at 25v and is powered using a power converter.

③ The signal from the low-frequency power amplifier is connected to the ultrasound probe, and then the cell phone is used to receive it for the experiment.

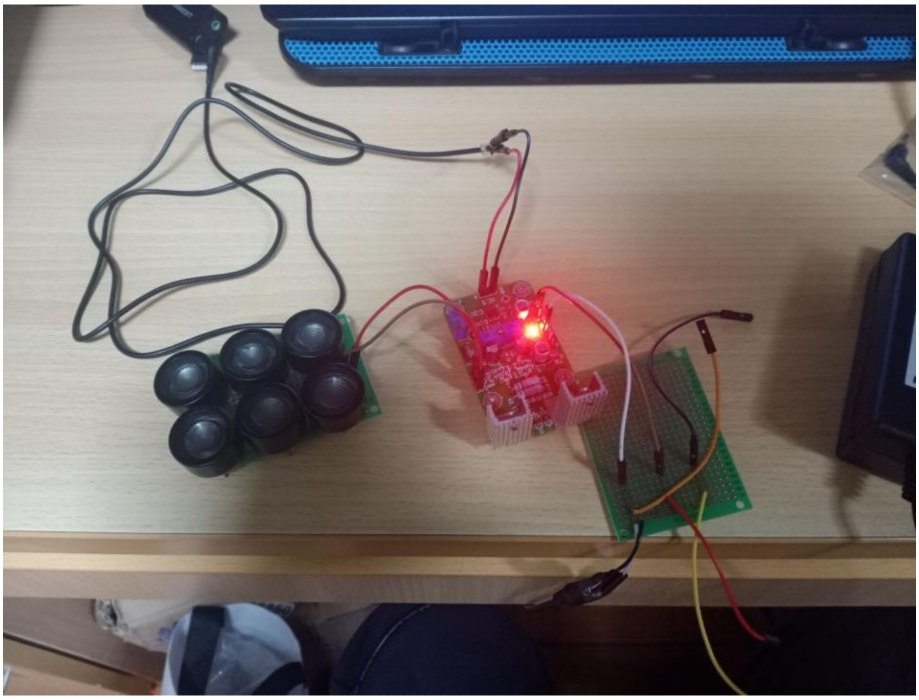


Fig.10 Attack system hardware picture

Results:

The attack distance is roughly about 50cm, restoring the attack distance of the previous senior.

The success rate tends to be extremely high at 100% within the attack distance, but it decreases as the attack distance increases beyond the range value.

Experimental summary:

The success rate of the attack is related to the angle of the phone and the ultrasound probe. This should be due to the ultrasound probe array is a 2 * 3 arrangement, the sound waves of each probe will overlap, if they can not be received at a specific angle, the effect will be very poor. Subsequent experiments using a single ultrasound probe. For this time, the success rate of arbitrarily angle of the phone remained unchanged, indicating that the summary rule is correct.

Different voice assistants or cell phones may have different effects. Previously used Huawei's voice assistant Xiao Yi for testing, the results were not satisfactory.

The experiment process needs to keep the environment quiet, if the environment is noisy, the experiment effect is very poor.

4.2 Defense system

The defense system takes the principle of eliminating ultrasound attacks as described in Section 2, and is implemented using hardware based on MATLAB simulation implementation. In this system, the equipment to be used are: ultrasound probes (two sets), low-frequency power amplifiers (two), headphone to Dupont cables, power

converters, and computers.

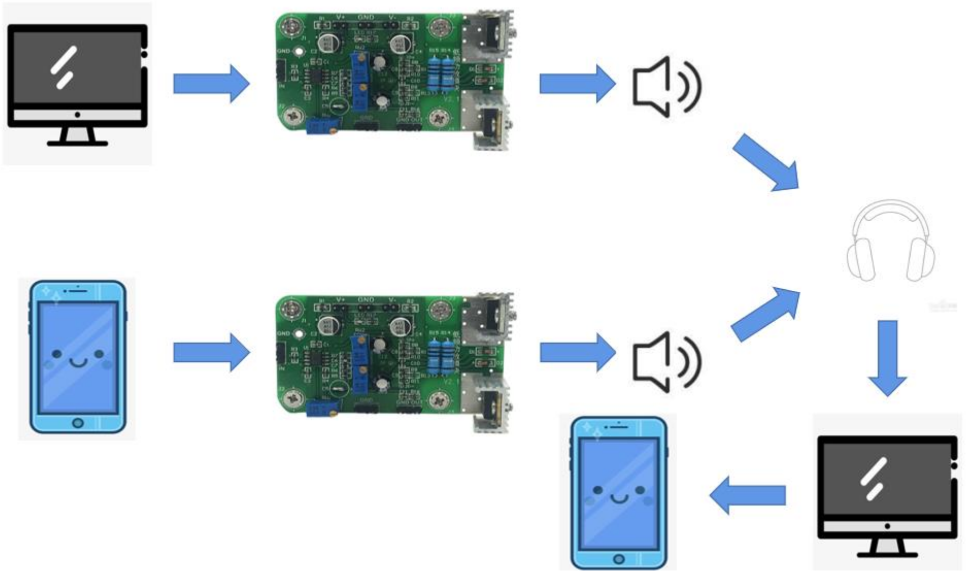


Fig.11 Defensive system implementation process:

① Play the attack signal in the computer, this operation is the same as the attack system operation.

② Play the defense signal in the cell phone, the pre-recorded 10khz defense signal will be played using the cell phone, the playback process is the same as the operation of the attack system.

③ Connect the computer to a headset for recording audio, input the recorded audio into the computer, run the defense program to filter out the attack instructions, and then play it to the phone, but the phone cannot be recognized.

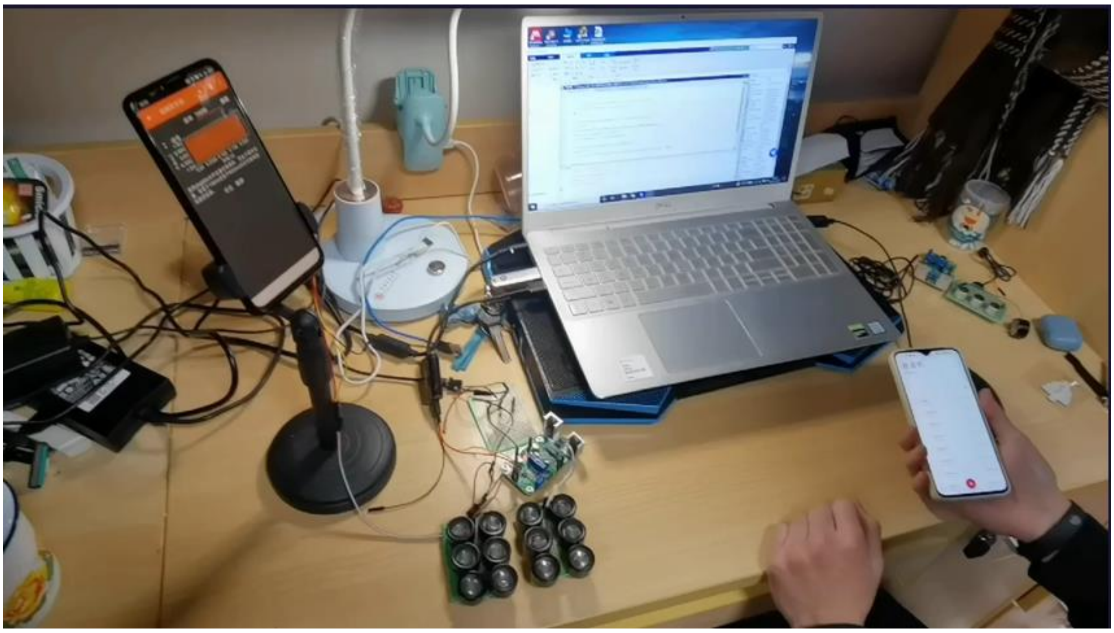


Fig.12 This figure is defense system. The phone on the stand plays the guard sound, and ultrasound probe plays attack commands to the target phone, which is on the hand. The

attack command is as same as in the attack part.

Experimental procedure:

In order to show the success of the defense, we must first make sure that the attack signal can be successfully attacked by using the command "Open WeChat", if the WeChat can be opened correctly, then the attack is successful. After determining the success of the attack, the attack signal is recorded into the computer together with the defense signal, and then played out through the MATLAB code. After three or more experiments, if there is no success, the defense is effective.

Results:

The attack part was successfully achieved. The signal outgoing after the program processing, after more than three trials, the cell phone can not respond to this, indicating that the defense signal is effective.

4.3 Frequency hopping system

The frequency hopping system takes the principle of frequency hopping ultrasonic attacks as described in Section 2, and is implemented using hardware based on MATLAB simulation implementation. In this system, the equipment to be used are: ultrasound probe, low frequency power amplifier, headphone to Dupont cable, power converter, and computer.

Frequency hopping system implementation process.

The implementation of the frequency hopping system is essentially the same as the implementation of the attack system, differing only in the procedure used in MATLAB.

Experimental results of frequency hopping system:

Since the operation of the frequency hopping system is the same as that of the attack system, the hardware schematic of the frequency hopping system is not posted separately.

The experimental environment, the target phone, the target voice assistant, and the instructions are the same as in the attack and defense systems.

Experimental procedure:

First of all, we need to prove whether the frequency hopping attack is successful or not, we will put the cell phone close to the ultrasonic probe to receive the frequency hopping attack signal. When we hear the "di di di" sound, it means that the frequency hopping attack can run successfully. Secondly, it is necessary to prove that the frequency hopping attack can break the defense system, the attack signal of the frequency hopping attack will replace the attack signal in the defense system for testing, if it can break the defense signal means that the frequency hopping attack is completely successful.

Results:

The ultrasonic probe showed a "drip-drip-drip" sound when playing a frequency hopping signal, but the probability of successful attack on the cell phone was not high. Because of the low probability of success, it was difficult to put it into the test of the defense system, and for this reason, no defense test was conducted.

5. Convolution Neural Networks for Defense

5.1 feature extraction

In our innovation point, we propose a binary classifier to solve this problem based on the task of distinguishing attack signal from normal signal. But for a classification task, how to quantify and represent an indefinite length of speech signal with feature vector is particularly important.

Generally, in any Automatic Speech recognition system, the first step is to extract features. In other words, we need to extract the identifying components of the audio signal and throw away the rest of the clutter. Knowing how speech is made helps us a lot in understanding speech. People produce sound through the vocal tract, and the shape of the vocal tract Determines what kind of sound it makes. The shape of vocal tract includes tongue, teeth and so on. If we can accurately know this shape, we can accurately describe the resulting phoneme. The shape of the channel is shown in the envelope of the short-time power spectrum of speech. MFCCs is a feature that accurately describes this envelope.

MFCC(Mel-Frequency Cepstral Coefficients)： Mel-Frequency is proposed based on the auditory characteristics of human ear, and it has a nonlinear corresponding relationship with Hz frequency. MFCC is the Hz spectrum characteristics calculated by using this relationship between them. There are mainly the following steps: pre-weighting, frame segmentation, windowing, fast Fourier transform (FFT), Mel filter banks, discrete cosine transform (DCT).

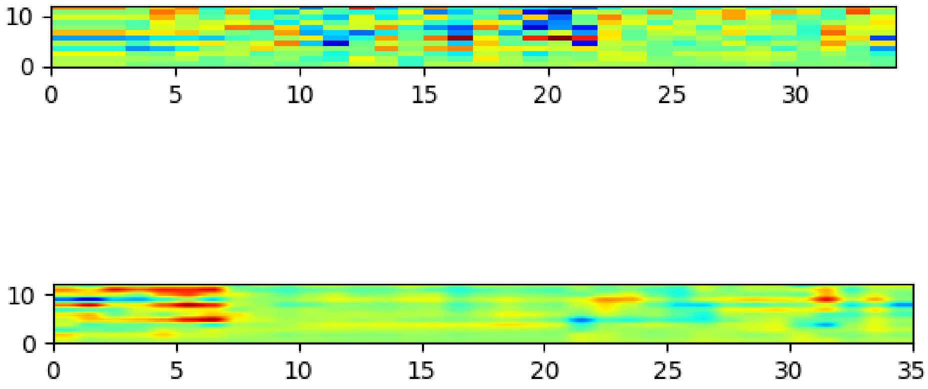


Fig.13 The figure above is Normal voice signal MFCC features and figure below is abnormal voice signal MFCC features：

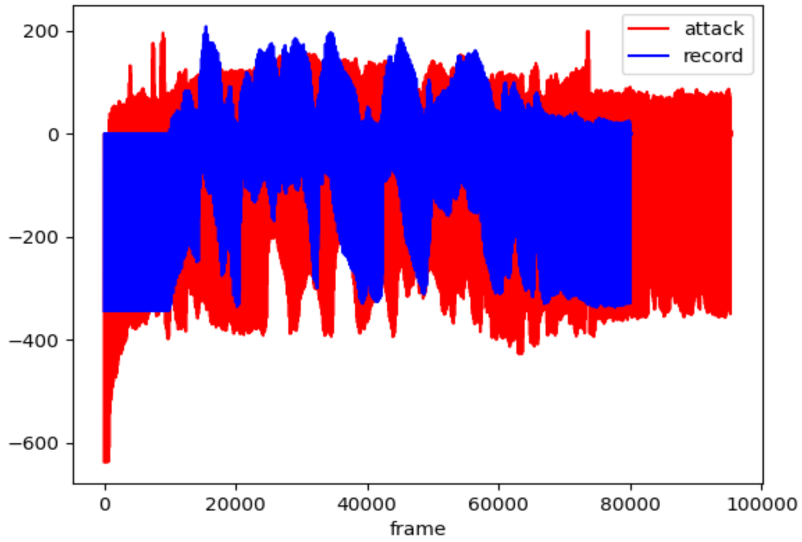


Fig.14 Contrast the MFCC features of normal voice signals and attack voice signals

5.2 Method

Convolutional Neural Networks (CNN) is a variant of multi-layer perceptron (MLP). Developed from earlier work on the visual cortex of cats by biologists Huber and Wessel. The cells of the visual cortex have a complex structure. These cells are very sensitive to subregions of the visual input space, which we call the receptive field, and in this way tiled over the entire visual field. These cells can be divided into two basic types, simple cells and complex cells. Simple cells maximally responded to marginal stimulus patterns within the receptive field. Complex cells have a larger receptive field and are locally invariant to stimuli from the exact location.

Generally, neurocognitive machines contain two types of neurons, namely, the sampling element that undertakes feature extraction and the anti-deformation convolution element. The sampling element involves two important parameters, namely, receptive field and threshold parameter. The former determines the number of input connections, while the latter controls the response degree to feature sub-patterns. Convolutional neural network can be regarded as a generalized form of neural cognitive machine, which is a special case of convolutional neural network.

In essence, convolutional network is a mapping between input and output. It can learn a large number of mapping relations between input and output without any precise mathematical expressions between input and output. As long as it is trained with known patterns, the network has the mapping ability between input and output pairs. The convolutional network performs tutor training, so its sample set is composed of vector pairs of the form (input vector, ideal output vector).

In our method, the input of the model is the MFCC feature of a piece of speech and the output is the logical value representing whether it is normal. The most common CNN structure, convolution layer + batch normalization + activation function with down-sampling

(MaxPool), is adopted to realize the network structure.

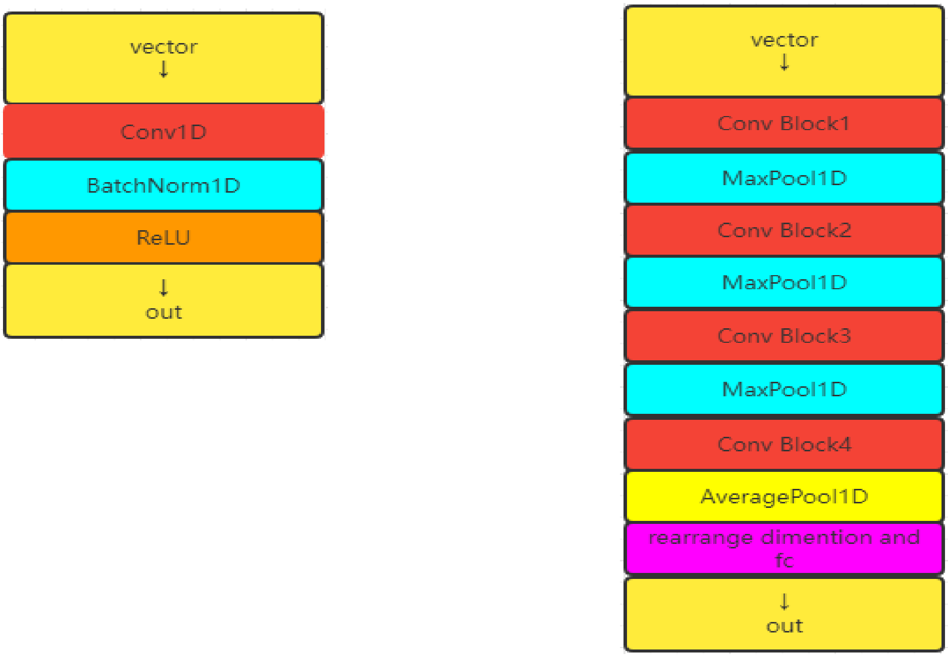


Fig. 15 The left figure is Conv block structure and the right one is model structure

6. Discussion

In paper Dolphin Attack: Inaudible Voice Commands, the authors proposed to use the ultrasonic carrier to carry information to attack the mobile phone voice assistant, so that the mobile phone voice assistant can be awakened without the mobile phone user's knowledge, and the attack can be realized. Our group also reproduced some of the attack scenarios in the paper according to the description of the paper, but during the experiment, our team found that there are problems in the laboratory, such as short attack distance, limited attack angle, strict requirements for attack scenarios, and fewer mobile phones with successful attacks. In this regard, our group put forward some conjectures and experiments to explain.

6.1 Ultrasound Problem

As for the reason for the short attack distance, our team believes that the attack signal is based on ultrasonic waves, but ultrasonic waves are prone to rapid attenuation in the air, which makes the signal strength of the input mobile phone microphone insufficient, and the audio strength carried by the mobile phone is weaker, making the mobile phone harder to perceive the attack voice, making the attack difficult to achieve. As the formula shows below:

$$A_x = A_o * e^{-afx}$$

A_o is the amplitude at the ultrasound probe, a is the attenuation coefficient, f is ultrasonic frequency, x is the distance between diaphragm and ultrasound probe, and A_x is the amplitude on the diaphragm. We can see that the higher the frequency and the greater the distance will result in a smaller magnitude of the received response. And the attenuation coefficient of ultrasound in the air is 0.09. We can find this problem by comparing our experiment with that of the previous seniors. The power amplifier used by the previous seniors can support 50 times the amplification efficiency, so that the farthest attack they can reach is about 50cm, while the power amplifier used in this experiment has a magnification of 30 times, so that the farthest attack can reach about 30cm if we don`t do any changes. We try to increase the amplitude of the digital signal so that the amplitude of the analog signal increases accordingly. Finally, the farthest attack distance we can reach is 50cm. We can see that in Fig.16. Compared with the experimental equipment of the previous seniors, it can be seen that the impact of the change in attack distance caused by the rapid attenuation of ultrasonic waves is more obvious.

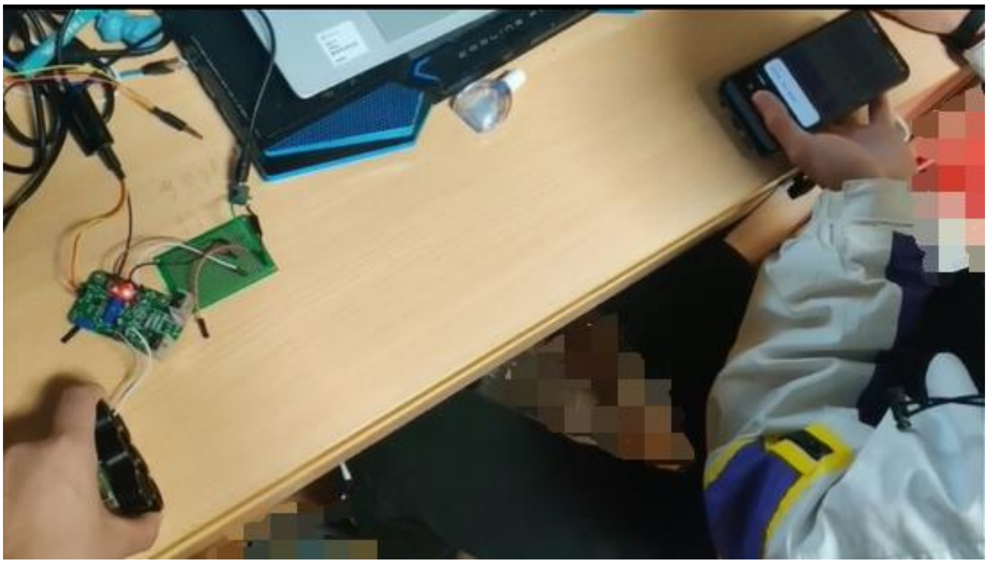


Fig. 16 The furthest distance attack system can reach is about 50cm.

Our group believes that the reason for the limited attack angle range is due to the multipath effect caused by the use of multiple ultrasound probes to generate attack signals. Due to the high frequency and short wavelength`s property of ultrasonic waves, the phase shift generated after a slight angular deviation is huge, resulting in obvious multipath effects. During the experiment, we combined multiple sets of ultrasonic probes to compare with a single ultrasonic probe, and found that a single ultrasonic probe would produce a more stable attack effect. The mobile phone can receive a wider angle, but it will also lead the problem of insufficient attack distance. Therefore, we believe that there will indeed be a certain multipath effect problem when the ultrasonic probe group is attacked.

6.2 Mobile Phones Update

The current mobile phones are updated faster, and both the hardware and software parts of the radio equipment have been improved. This could lead to the problem that there are fewer successful attacks on mobile phones. In the process of implementing the attack, it is very important to use the nonlinearity of the power amplifier of the mobile phone

microphone in the high frequency part. However, due to the upgrading of hardware, a good linear amplification and fitting can be performed in the amplifier part, and the amplitude of the nonlinear effect is greatly reduced, making it difficult for the attack signal to be moved to the low-frequency part by nonlinearity, which leads to the failure of the attack. When conducting the experiments, we attacked cell phones of nearly one to four years, and recorded the attack signals with a tape recorder. It can be clearly found that the more recent the mobile phone produced, the better the effect of resisting the attack, and the earlier the mobile phone produced can clearly hear the command of the attack signal. We used Huawei P50 (produced in 2021), Honor V20 (produced in 2019), OnePlus 7 (produced in 2019), OPPO A9 (produced in 2018). After the attack, we found that except Huawei P50, which cannot be broken, other mobile phones can be easily attacked. At the same time, modern mobile phones are basically equipped with dual microphones for noise reduction. One microphone receives background noise, and the other microphone receives the main signal. By detecting the audio part where the difference between the background noise and the main signal is 5 to 6 dB, the background noise is filtered out from received signal. This also limits the attack angles of attack system because it is easy to detect as background noise.

6.3 Complex Scenario

The attack scenario is limited because our team found that the attack signal received by the mobile phone was obviously doped with high-frequency signals after recording the attack audio, while the low-frequency signals had low strength and were easily distorted, resulting in inconsistent semantics. clear. This makes it easier for the voice assistant to detect the surrounding background noise with clear semantics and high intensity due to the distortion of the attack signal itself, although the mobile phone has the ability to filter out some background noise in a noisy environment. In a quiet scene, the signal source is single, and it is easier for the voice assistant to detect the voice carried in the attack signal. This weakness is also the starting point of our team's defense this time. Since the real voice and the voice of the attack signal are very different and very easy to distinguish, our team modeled it as a two-class problem and used machine learning methods to conduct real voice. Detection and defense with the identification of attack signals.

7. Future Work

For the idea of adding voiceprint to defend against attacks proposed by the previous group, we also implemented the attack. We hacked the latest version of XiaoBu voice assistant on OnePlus 7 using ultrasonic attack. This version of the voice assistant has voiceprint recognition technology, which needs to record the user's voiceprint in advance for comparison and recognition. During the attack, we found that the voice assistant can be well recognized and awakened by using the attack signal modulated by the user's voice. But the voice assistant cannot be woken up with other people's voice commands. So we come to the conclusion that to a certain extent, voiceprint recognition technology can make the voice

assistant not be broken by other people's voice. However, if the attacker uses speech synthesis technology, under the premise of the collected language corpus of some users, he can also synthesize the user's voice to attack, so in essence, the task of defense is still not well completed. At the same time, we believe that future work can perform speech synthesis and ultrasonic attack to make the speech attack technology more perfect and mature.

In addition, the CNN classification network implemented by our group this time is a lightweight network, and the entire network parameter size is 96KB. We also sampled vocal recordings and attack recordings as our training set during training. However, due to the limited time, the data set we collected also has certain problems. For example, the frequency of the attack signal is relatively simple, and only the carrier frequency of 21KHz is used for training; at the same time, there are few accents and the recording environment is simple (basically in a quiet environment). For the frequency hopping signal, there is no recording and sampling, etc., so we think that the next step we can expand the training data set, so that the network model can fit more real environments, and more able to achieve defense classification goal. At the same time, we have verified the real-time and effectiveness of the software running on the computer, and we can consider porting the system to the mobile phone in the next step. Adapted to mobile phones for attack defense.

8. Acknowledgement

Our group is very appreciated to Prof. Xiaohua Tian for the course project guidance and experimental equipment, which enabled us to gain a lot of hardware knowledge and Internet of Things knowledge in the course, which increased our ability and knowledge. We are also very grateful to the course teaching assistants for their help and assignments. At the same time, we are also very grateful to the previous seniors who personally guided us to carry out the experimental operation and helped us quickly get started on this topic! Finally, we are very grateful to the members of the team for their hard work, and only have the final good results!

9. Appendix

9.1 Ultrasound Attack Code

```
%% Clear Variables and Load Signal
clc; close all;
clear;
[call_signal,upsample_fs] = audioread('wechat.m4a');
call_signal = call_signal(:,1);
sound(call_signal,upsample_fs);
%% Plot Orignal Singal FFT
```

```
freq = abs(fft(call_signal));
Y = fftshift(freq);
f = (-length(freq(:,1))/2:length(freq(:,1))/2-
1)*(upsample_fs/length(freq(:,1)));
plot(f,Y);
title('Orignal Signal')
xlabel('Freq/Hz')
%% AM
fc = 24000;
high_freq=resample(call_signal,96000,upsample_fs)+1;

x = modulate(high_freq,fc,96000,'am');
freq = abs(fft(x));
Y = fftshift(freq);
f = (-length(freq(:,1))/2:length(freq(:,1))/2-
1)*(96000/length(freq(:,1)));
plot(f,Y);
xlabel('Freq/Hz');
title('AM modulate');
audiowrite('40khz.wav',x,48000);
%% Plot Signal After AM
freq = abs(fft(x));
n = length(x);
Y = fftshift(freq);
f = (-n/2:n/2-1)*(48000/n);
plot(f,Y);
sound(x,upsample_fs);
%% Self Convolution and Plot its FFT
x_2 = x .* x ;
freq = abs(fft(x_2));
n = length(x);
freq(1)=0;
[m,index] = max(freq);
freq(index) = 0;
% Y = fftshift(freq);
f = (-n/2:n/2-1)*(96000/n);
plot(f,freq)
title('Self Conv Signal')
%% Lower-Freq Filter
[b, a] = butter(9,20000/96000,'low');
x_new = filter(b,a,x_2);

freq = abs(fft(x_new));
```

```
freq(1) = 0;
n = length(x_new);
Y = fftshift(freq);
f = (-n/2:n/2-1)*(96000/n);
plot(f,Y);
title('Recover signal');
xlabel('Freq/Hz')
% sound(x_new,fc);
```

9.2 Ultrasound Defense Code

```
clc;close all;

[attack_upsample_sig,upsample_fs] =
audioread('40khz.wav');
disp(upsample_fs);
disp(length(attack_upsample_sig));
attack_upsample_sig=resample(attack_upsample_sig,9600
0,48000);
length_attack = size(attack_upsample_sig,1);
N = length_attack;
t=(0:N-1)/96000;
f = 10000;
defense_sig = sin(2*pi*f*t)'/1;

attack_upsample_sig = attack_upsample_sig +
defense_sig;
attack_upsample_sig =
attack_upsample_sig/max(attack_upsample_sig);
figure;subplot(211),plot_fft(attack_upsample_sig,9600
0)

xlabel("f/kHz");

t = (1:1:length(attack_upsample_sig))/upsample_fs;
subplot(212),plot(t,attack_upsample_sig);
xlabel("t/s");

nonlinear_sig =
attack_upsample_sig.*attack_upsample_sig +
attack_upsample_sig;
nonlinear_sig = nonlinear_sig/max(nonlinear_sig);
figure;subplot(211),plot_fft(nonlinear_sig, 96000)
```

```
xlabel("f/kHz");
%
t = (1:1:size(nonlinear_sig,1))/upsample_fs;
subplot(212),plot(t,nonlinear_sig);
xlabel("t/s");
% saveas(gcf,'nonlinear.jpg');

d = fdesign.lowpass('Fp,Fst,Ap,Ast',5/48,8/48,1,60);
d_low =
fdesign.lowpass('Fp,Fst,Ap,Ast',12/48,14/48,1,60);
d_high =
fdesign.highpass('Fst,Fp,Ast,Ap',12/48,14/48,60,1);
% d =
fdesign.lowpass('Fp,Fst,Ap,Ast',4/48,7.5/48,1,60);
Hd = design(d,'butter');
Hd_low=design(d_low,'butter');
Hd_high = design(d_high,'butter');
%[mix_base_sig, attack_upsample_sig] =
butter(6,15000*2/96000);
mix_base_sig=filter(Hd,nonlinear_sig);
mix_base_sig=mix_base_sig/max(mix_base_sig);

figure;subplot(211),plot_fft(mix_base_sig, 96000)
xlabel("f/kHz");

subplot(212),plot(t,mix_base_sig);
xlabel("t/s");

high_freq_sig=filter(Hd_low,nonlinear_sig);
high_freq_sig=filter(Hd_high,high_freq_sig);
high_freq_sig = high_freq_sig/max(high_freq_sig);
high_freq_sig = high_freq_sig(100:end);
figure;subplot(211),plot_fft(high_freq_sig,48000)
xlabel("f/kHz");
%
t = (1:1:size(high_freq_sig,1))/upsample_fs;
subplot(212),plot(t,high_freq_sig);
xlabel("t/s");

d_con_low =
fdesign.lowpass('Fp,Fst,Ap,Ast',10/48,13/48,1,60);
Hd_con_low = design(d_con_low, 'butter');
high_freq_sig_col = high_freq_sig .* high_freq_sig;
```

```
%high_freq_sig_col = conv(high_freq_sig,
high_freq_sig);
high_freq_sig_col =
high_freq_sig_col/max(high_freq_sig_col);
high_freq_sig_col = high_freq_sig_col(100:end);
high_freq_sig_col =
filter(Hd_con_low,high_freq_sig_col);

figure;subplot(211),plot_fft(high_freq_sig_col,48000)
xlabel("f/kHz");
t = (1:1:size(high_freq_sig_col,1));
subplot(212),plot(t,high_freq_sig_col-0.4);
xlabel("t/s");
saveas(gcf,'attack_conv.jpg');
t = (1:1:size(record_sig,1));
plot(t,record_sig)
sound( record_sig, 48000)

t = (1:1:size(high_freq_fft_col,1))/upsample_fs;
subplot(212),plot(t,high_freq_sig_col);
xlabel("t/s");
audiowrite('attack.wav',high_freq_sig_col,96000);

order = 50;
error_anc = zeros( size(high_freq_sig ,1),1);
y_anc = zeros( size(mix_base_sig, 1),1);

after_anc =
ifft(mix_base_fft(1:size(attack_base_sig,1)) -
attack_base_fft(1:size(attack_base_sig,1)));

FrameSize = 256;
Length = size(high_freq_sig,1);
NIter = Length/FrameSize;
lmsfilt2 =
dsp.LMSFilter('Length',100,'Method','Normalized LMS',
'StepSize',0.01);
mix = zeros( size(high_freq_sig,1),1);
wout = zeros(100,ceil(NIter));

for k = 1:NIter-1
   x = high_freq_sig_col((k-
1)*FrameSize+1:k*FrameSize);
```

```
    d = mix_base_sig((k-1)*FrameSize+1:k*FrameSize);

    [y,e,w] = lmsfilt2(x,d);

    error_anc((k-1)*FrameSize+1:k*FrameSize) = e;
    mix((k-1)*FrameSize+1:k*FrameSize) = d;
    y_anc((k-1)*FrameSize+1:k*FrameSize) = y;
    wout(:,k)  = w;
end

N = size(error_anc,1);
error_anc_fft = abs(fft(error_anc))/N*2;
f = 96000/N:96000/N:96000;
figure;subplot(211),plot(f/1000,error_anc_fft);ylim([
0 0.01]),xlim([0 15])
xlabel("f/kHz");
title("error anc fft");
y_anc_fft = abs(fft(y_anc))/size(y_anc,1)*2;
f = 96000/size(y_anc,1):96000/size(y_anc,1):96000;
subplot(222),plot(f/1000,y_anc_fft);ylim([0
0.01]),xlim([0 15])
xlabel("f/kHz");
f =
96000/size(mix_base_fft,1):96000/size(mix_base_fft,1)
:96000;
subplot(212),plot(f/1000,mix_base_fft);ylim([0
0.01]),xlim([0 15])
xlabel("f/kHz");
t = (1:1:N)/96000;
subplot(212),plot(t,error_anc);
xlabel("t/s");
saveas(gcf,'after_anc.jpg');

final_anc = mix_base_sig - error_anc;
final_anc_fft = abs(fft(final_anc));
figure;subplot(211),plot(f/1000,final_anc_fft);ylim([
0 0.001]),xlim([0 25])
xlabel("f/kHz");
title("error anc fft");

t = (1:1:N)/96000;
subplot(212),plot(t,final_anc);
xlabel("t/s");
```

```
audiowrite('error_out.wav',error_anc,96000);
audiowrite('y_out.wav',y_anc,96000);

function m=plot_fft(X,fs)
Fs=fs;
L=length(X);
n = 2^nextpow2(L);
Y = fft(X,n);
P2 = abs(Y/L);
P1 = P2(1:n/2+1);
P1(2:end-1) = 2*P1(2:end-1);
plot(0:(Fs/n):(Fs/2-Fs/n),P1(1:n/2))
ylim([0,0.0005]);
m=1;
end
```

9.3 Hopping Attack Code

```
clc; clear; close all
 [signal,Fs] = audioread('attack record.m4a');
fs = 96000;
sound_in = resample(signal,fs,Fs);
sound_in = sound_in/max(sound_in);
 delta = 1200;
[B,A] = butter(10,delta/(fs/2));
s = filter(B,A,sound_in);
s = s + 1;
fcList = [25000,32000,40000];
deltaList = [800,800,800];

interval = 7200; % hopping interval
% Divided into 100ms tracks
group_num = floor(length(sound_in)/interval);
freq_ord = 1;
hopping = zeros(length(sound_in),1);
time = length(sound_in)/fs;
hopping_interval = interval/fs;
for i=1:group_num
    fc = fcList(freq_ord + 1);
    delta = deltaList(freq_ord + 1);
    freq_ord = mod(freq_ord + 1,3);

    x = modulate(s((i-
1)*interval+1:i*interval),fc,fs,'am');

    hopping((i-1)*interval+1:i*interval) =
x/max(abs(x));
    [B,A] = butter(10,(fc-delta)/(fs/2),'high');
    out = filter(B,A,hopping((i-
1)*interval+1:i*interval));
    [D,C] = butter(10,(fc+delta)/(fs/2));
    out = filter(D,C,out);

    hopping((i-1)*interval+1:i*interval) =
out/max(abs(out));
end
sound(hopping,fs);
```

9.4 CNN Trainnig Code

Due to it written on notebook in Jupyter, you can see more detail in README.txt and code file folder.

10. Reference

[1] Guoming Zhang et al.DolphinAttack：Inaudible Voice Commands [C]. , 2017.

[2] Yoon K. Convolutional Neural Networks for Sentence Classification [C] .2014.

[3] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible Voice Commands: The Long-Range Attack and Defense[C]. NSDI. 2018.

[4] Dongqiu Huang,Zhihong Tian, Shen Su,et al. A defense scheme of voice control system against DolphinAttack.CIAT.2020

[5] Shen, Sheng & Roy, Nirupam & Guan, Junfeng & Hassanieh, Haitham & Roy Choudhury, Romit. (2018). MUTE: bringing IoT to noise cancellation. 282-296. 10.1145/3230543.3230550.