applied sciences

MDPI

*Article*

# Adversarial Attack and Defense on Deep Neural Network-Based Voice Processing Systems: An Overview

**Xiaojiao Chen [1], Sheng Li [2] and Hao Huang [1,3,*]**

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; xiaojiaochen@stu.xju.edu.cn
[2] National Institute of Information and Communications Technology, Kyoto 619-0288, Japan; sheng.li@nict.go.jp
[3] Xinjiang Provincial Key Laboratory of Multi-Lingual Information Technology, Urumqi 830046, China
* Correspondence: hwanghao@gmail.com

**Abstract:** Voice Processing Systems (VPSes), now widely deployed, have become deeply involved in people's daily lives, helping drive the car, unlock the smartphone, make online purchases, etc. Unfortunately, recent research has shown that those systems based on deep neural networks are vulnerable to adversarial examples, which attract significant attention to VPS security. This review presents a detailed introduction to the background knowledge of adversarial attacks, including the generation of adversarial examples, psychoacoustic models, and evaluation indicators. Then we provide a concise introduction to defense methods against adversarial attacks. Finally, we propose a systematic classification of adversarial attacks and defense methods, with which we hope to provide a better understanding of the classification and structure for beginners in this field.

**Keywords:** adversarial attack; adversarial example; adversarial defense; speaker recognition; speech recognition



**Citation:** Chen, X.; Li, S.; Huang, H. Adversarial Attack and Defense on Deep Neural Network-Based Voice Processing Systems: An Overview. *Appl. Sci.* **2021**, *11*, 8450. https://doi.org/10.3390/app11188450

Academic Editor: Yoshinobu Kajikawa

Received: 15 August 2021
Accepted: 8 September 2021
Published: 12 September 2021



## 1. Introduction

With the successful application of deep neural networks in the field of speech processing, automatic speech recognition systems (ASR) and automatic speaker recognition systems (SRS) have become ubiquitous in our lives, including personal voice assistants (VAs) (e.g., Apple Siri (https://www.apple.com/in/siri (accessed on 9 September 2021)), Amazon Alexa (https://developer.amazon.com/en-US/alexa (accessed on 9 September 2021)), Google Assistant (https://assistant.google.com/ (accessed on 9 September 2021)), iFLYTEK (http://www.iflytek.com/en/index.html (accessed on 9 September 2021))), voiceprint recognition systems on mobile phones, bank self-service voice systems, and forensic testing [1]. The application of these systems has brought great convenience to people's personal and public lives, and, to a certain extent, enables people to access help more efficiently and conveniently.

Recent research, however, has shown that the neural network systems are vulnerable to adversarial attacks [2–5]. This will threaten personal identity information and property security and leaves an opportunity for criminals. From the perspective of security, the privacy of the public is in danger. Therefore, for the purpose of public and personal safety, mastering the methods of attack and defense will enable us to prevent problems before their probable occurrence.

In response to the problems mentioned above, the concept of adversarial examples [2] was born. The original adversarial examples were applied to image recognition systems [3,4,6,7] and then researchers expanded the adversarial examples to include speech recognition, speaker recognition, and other systems. Compared with the problem of adversarial example classification on pictures, the voice presents the following challenges: first, when disturbance is added to audio, it can be heard by humans, but the disruption of pictures is aimed at the pixels, and is harder to discover for humans. Secondly, in a practical

sense, image classification systems are primarily used in medical imaging, etc. Still, voice recognition systems are more valuable and are closely related to everyone who has a smartphone. Wrong instructions may cause the loss of a large amount of users' property.

With the further development of science and technology, new types of speech systems may emerge in an endless stream, but the problem that neural networks are vulnerable to attacks has not been solved. Therefore, before solving new issues, overview research on existing technologies is essential and vital. This article upholds this original intention, and the main contributions made in this paper are:

- In order to better illustrate the application of adversarial attacks and defenses in sound processing systems, we introduce in detail the contents of adversarial attacks, including methods for generating adversarial examples and metrics for adversarial attacks. At the same time, we summarize the main methods of adversarial aggression and defense in speaker recognition and speech recognition, respectively.
- Based on the above research methods, we systematically categorize the methods of adversarial attack and defense.

This overview is organized as follows. We first review the background information about attacks and VPSes by showing the basic concept of adversarial examples, automatic speech recognition systems, speaker recognition systems, and defense. Moreover, we introduce the threat model in detail. Accordingly, the methods of adversarial defense are categorized through their characteristics.

## 2. Background

In this section, we briefly introduce the basic concepts of attack and defense and the ASR system, and the speaker recognition system is explained to facilitate subsequent understanding.

### 2.1. Attack

Most of the attacks in the voice field are evading attacks. The basic concept is to convert the target value of the system into non-targets. The most vivid example is to add disturbance to the correct audio before passing the ASR system and result from the wrong text context. Taking the particularity of audio into account, usually, people can understand the task of attacking voice processing systems as having two points, (1) fooling the neural network to produce false results, (2) avoiding being discovered by humans. We review existing attack models, and we deem that the completion of the first task is based on the audio adversarial with the addition small perturbations to input audio. Then we use the principle of psychoacoustics [8] to achieve the goal that makes the attack unexpected and silent.

Figure 1 illustrates the general working flow of an adversarial attack in voice processing systems. The correct result of the original audio passing through the target system is A, while the attacker adding perturbation to the original audio will make the target system achieve the wrong result B, and B is different from A which means the attack has an outstanding performance.

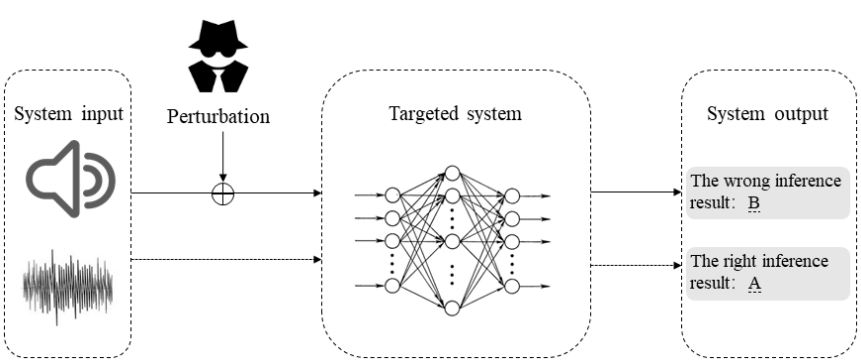


**Figure 1.** The general working flow of an adversarial attack in a Voice Processing system.

Therefore, next, we mainly introduce the main generation methods of adversarial examples and the application of acoustic masking in adversarial examples, and the evaluation metrics for attack models of voice processing systems.

2.1.1. Adversarial Examples

A breakthrough in attacks was made by Szegedy et al. [2]. For the first time it was proven that the neural network can be misclassified by adding a small amount of disturbance that is imperceptible to humans.

More formally, with a neural network $f$ and an input $x$, we want to find a small malicious perturbation $\delta$:

$$\tilde{x} = x + \delta \quad such\ that \quad ||\delta||_p < \epsilon \tag{1}$$

with the goal of forcing the network to produce an erroneous output for $\tilde{x}$, where $||\cdot||_p$ is the $p - norm$. In other words, if $x$ has a true output $y$, then the attacker forces the network to produce $\tilde{y} \neq y$ for the perturbed example $\tilde{x}$.

To systematically analyze approaches for generating adversarial examples, we analyze the approaches for generating adversarial examples.

L-BFGS: Szegedy et al. [2] first introduced adversarial examples against a deep neural network in 2014. They model the problem as a constrained minimization problem called L-BFGS:

$$min\ ||\delta||_2^2 \quad s.t. \quad f(x+\delta) = \tilde{y} \quad \delta \in [0,1]^m \tag{2}$$

In general, the exact computation of $min||\delta||_2$ is a hard problem, so they use the addition of the minimized loss function:

$$min\ c \cdot ||\delta||_2^2 + loss_f(x+\delta, \tilde{y}) \quad s.t. \quad \delta \in [0,1]^m \tag{3}$$

where, $\ell((x+\delta), \tilde{y})$ is a loss function of a deep neural network. One common loss function to use is cross-entropy. A line search is performed to find the constant $c > 0$ that yields an adversarial example of minimum distance: in other words, we repeatedly solve this optimization problem for multiple values of $c$, adaptively updating $c$ using bisection search or any other method for one-dimensional optimization. This generation algorithm has the characteristics of fast generation speed and low memory footprint, but there is still a lot of room for improvement in terms of confrontation.

The fast gradient sign method (FGSM): The fast gradient sign method [6] has two key difference from the L-BFGS method: first, it has been optimized for $L_\infty$ distance measurement, and second, its main purpose is to quickly generate the adversarial examples rather than generate very close examples. Given an input $x$ the fast gradient sign method sets:

$$\tilde{x} = x + \epsilon \cdot sign(\nabla_x J(\theta, x, y)) \tag{4}$$

where the perturbation is $\delta = \epsilon \cdot sign(\nabla_x J(\theta, x, y))$; $\epsilon$ is chosen to be sufficiently small so as to be undetectable; $\theta$ is the parameter of the classification model; $y$ satisfies $y = f(x)$ and is the correct output of $x$; $J(\theta, x, y)$ is the loss function used in this deep neural network. It is worth noting that this method is mainly focused on quickly generating adversarial examples rather than getting the smallest disturbance.

Basic iterative method (BIM): Although FGSM is simple and computationally efficient compared to other methods, it has a lower success rate with a nonlinear model. The reason that leads to this phenomenon is that, for the linear model, the direction in which the loss decreases is clear, and even if you iterate multiple times, the direction of the disturbance will not change. However, for a non-linear model, the direction may not be completely correct if you only perform one iteration, so multiple iterations are needed to determine

the optimal situation. The BIM [9] method has been improved on FGSM, and one step is divided into many small steps to iteratively obtain adversarial examples:

$$X_0^{adv} = X$$

$$X_{N+1}^{adv} = Clip_{X,\epsilon}\{X_N^{adv} + \alpha \cdot sign(\nabla_X J(\theta, X_N^{adv}, y))\} \quad (5)$$

where the $Clip_{X,\epsilon}(A)$ denotes element-wise clipping $A$. This method can generate the adversarial examples in nonlinear model, while at the cost of expensive computation.

Deepfool: Moosavi–Dezfooli et al. [10] proposed DeepFool to compute a minimal norm adversarial perturbation by the basic ideal of the distance from the input $x$ to the boundaries of the classifier. That is, they assume $\hat{k}(x) = sign(f(x))$, where $f$ is an binary classification function and satisfies $f(x) = w^T x + b$. It can be easily seen that its affine plane is $\mathcal{F} = x : w^T x + b = 0$. When a disturbance is added to a point $x_0$ and perpendicular to the plane $\mathcal{F}$, the disturbance added is the smallest and can meet the iteration requirements, as in the formula:

$$\begin{aligned} \delta_i(x_0) :&= argmin||\delta||_2 \\ &= -\frac{f(x_0)}{||w||_2^2} \end{aligned}$$

Furthermore, in the overall iterative process, the generation of adversarial examples can be expressed as:

$$\arg\min_{\delta_i} ||\delta||_2 \quad s.t.\ f(x_i) + \nabla f(x_i)^T \delta_i = 0 \quad (6)$$

DeepFool uses iteration to generate the minimum norm to counter the disturbance. At each step, the data values located within the classification boundary are modified step by step to outside the boundary until a classification error occurs. This method maintains almost the same resistance as FGSM, while the disturbances generated are smaller.

Jacobian-based Saliency Map Attack (JSMA): JSMA [11] was proposed by Nicolas et al. It is a method for generating adversarial examples for the type of deep neural network. It uses the forward guide number to implement it. The generation of the forward guide number uses the Jacobian matrix of the function function in the trained network. Given the function of network $F$, we can obtain the forward derivative by this formula:

$$\nabla F(X) = \frac{\partial F(X)}{\partial (X)} = [\frac{\partial F_j(X)}{\partial x_i}]_{i \in 1..M, j \in 1..N} \quad (7)$$

Universal adversarial examples: Methods such as FGSM, and DeepFool can only generate a single audio against perturbations, while universal adversarial examples [12] can generate perturbations almost imperceptible that attack any voice processing systems, and these perturbations are also harmful to humans. The method used in this paper [12] is similar to DeepFool, which uses anti-disturbance to push the image out of the classification boundary, but the same disturbance is for all. Although this article only targets a single network, ResNet, it has been proven that this malicious perturbation can be generalized to other networks.

Genetic algorithm: It can be seen that the methods mentioned above for generating adversarial examples are all based on the system loss function or the gradient of the system's network function. Unfortunately, the above method usually does not work when the attacker does not know the prior information of the system's loss function and network function.

Hence, researchers [13–15] use a genetic algorithm to generate a black-box adversarial example. A genetic algorithm is a gradient-free optimization method, which avoids having to grasp the prior information of the attacked systems. The algorithm accepts the original audio clip $x$ and target output label $\tilde{y}$ as entries. Adding random noise to a set of patterns in a given audio clip creates many candidate adversarial examples. To minimize the impact of noise on people, it is only necessary to place the sound in the least prominent position of the random system of audio examples. Calculate the fitness score of each population member

based on the predicted score of the target output label, and apply selection, crossover, and mutation [16] to create adversarial examples from the current generation to the next generation. Choice means that more adaptable members may become part of the next generation. Crossover population members and mix them to produce new 'child' added to the new population. In the end, mutation adds random noise to the child with a small probability, then passed on to the next generation. The algorithm iterates on this process for the preset number of epochs or before the attack is successful. Although this genetic algorithm resolves the problem of no ideal of prior information, it introduces a huge amount of calculation.

Signal processing methods: In addition to the methods mentioned above, there are also methods for generating counter-samples based on traditional signal processing. The researchers investigate the starting point for the characteristics of DNN input as frequency-domain features. First, the relationship between the frequency spectrum and the time domain waveform is used. For this advantage, Time domain inversion (TDI) [17] is proposed, which is a method of changing the time-domain audio waveform to obtain adversarial samples without changing the frequency spectrum. Secondly, it is considered that in the complex frequency domain, a random phase method (RPG) [17] is proposed to generate adversarial samples since the frequency spectrum can maintain the same amplitude under different phases. The two methods for generating counter-samples are based on the characteristics of the time domain signal from the FFT to the frequency domain.

The above is a more systematic description of the method of generating adversarial examples. It can be clearly seen that the methods of generating adversarial examples can be divided into these three categories, gradient-based methods, genetic algorithms, and traditional algorithms. These methods have their advantages in terms of the amount of calculation and the ease of implementation, and FGSM is a more widely used method.

#### 2.1.2. Psychoacoustics

Psychoacoustics [8] is the science of how human beings perceive the sound with the ear and what they perceive, and it explores the statistical relationships between acoustic stimuli and hearing sensations. Whether the human ear can hear a sound signal depends on its frequency, intensity, and interference from other sounds. The most recent attacks on speech recognition were carried out without being noticed. The psychoacoustic model is to find out the redundant information in the audio signal to not affect the auditory effect.

Research in psychoacoustics shows that hearing and understanding the human voice has its strengths and weaknesses. If there are multiple sources of sound, humans are more likely to focus on one source. The phenomenon is known as the Cocktail Party effect [18], in which humans can set an inappropriate sound. According to the psychoacoustics, we master the human voice frequency is limited to the band range 20 Hz form 20 kHz [8], which means that, beyond this range, human ears cannot perceive it.

The acoustic principles are the critical frequency band, absolute hearing threshold, frequency masking, and temporal masking. Therefore, a good understanding of the sensory response of the human auditory system (HAS) is essential for the development of a psychoacoustic model for generating the audio adversarial examples, where the perceptual quality of adversarial examples must be at the lowest extent. Furthermore, we can take advantage of auditory masking to better perturb the human voice:

- Frequency Masking. Frequency masking implies masking between two sounds of close frequency, where a low-level maskee is inaudible by a simultaneously occurring louder masker. In simple terms, the masker can be seen as creating a "masking threshold" in the frequency domain. Any signals which fall under this threshold are effectively imperceptible. Figure 2 gives a vivid example of simultaneous masking, where sound $S_0$ is the masker. Because of the presence of $S_0$, the threshold in quiet is elevated to produce a new hearing threshold named the masking threshold; in this example, the weaker signal $S_2$ and $S_3$ are entirely inaudible, as their sound pressure level is below the masking threshold.

- Temporal Masking. In addition to frequency masking, auditory masking can also occur when the maskee is present immediately preceding or following the masker. This is called temporal masking or non-simultaneous masking. There are two kinds of non-simultaneous masking: (1) pre-masking or backward masking, occurring just before the onset of the masker, and (2) post-masking or forward masking, occurring after the removal of the masker. In general, the physiological basis of non-simultaneous masking is that the auditory system requires a particular integration time to build the perception of sound, where louder sounds require longer integration intervals than softer ones.

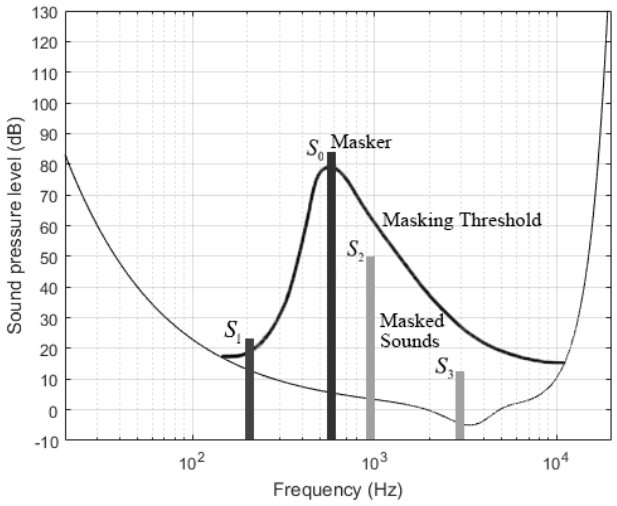


**Figure 2.** The masking effect of a pure tone with a 500 Hz loudness of 80 dB in the presence of other pure tones, and this picture is cited from [8].

More specially, it can be seen from the above that most people speaking systems use frequency domain features for processing, so we mostly use frequency-domain masking when attacking. In the actual use of frequency masking, the maker we choose can be the original voice so that only the adversarial example is below the threshold. However, in a real physical voice environment, Yuan et al. [19] found that the adversarial examples could not perceive in the presence of music and played music during the attack to mask the sound of the adversarial examples.

#### 2.1.3. Metrics

To evaluate the quality of adversarial examples, we use some standard metrics for efficiency and inconspicuousness.

Word Error Rate (WER): The WER calculation is based on a measurement called the "Levenshtein distance" [20]. The Levenshtein distance is a measurement of the differences between two "strings". We compute WER with the Levenshtein distance $\ell$:

$$WER = 100 \cdot \frac{\ell}{N} = 100 \cdot \frac{S + D + I}{N} \tag{8}$$

where the sum overall substituted words is $S$, inserted words $I$, and deleted words $D$. According to Equation (8), lower WER often indicates that the ASR software is more accurate in recognizing speech. A higher WER, then, often indicates lower ASR accuracy, while in the attack situation we except the high WER, which demonstrated a better attack rate. Therefore, this is an intuitive and effective indicator for evaluating attack performance. Segmental Signal-to-noise ratio (SNRseg): The WER can only measure the success of an adversarial example in fooling an ASR system. Considering the ASV system, we find the perturbation as a common factor. In addition, for adversarial examples, the smaller the disturbance, the better the attack performance. Specifically, we use the Segmental Signal-

to-Noise Ratio (SNRseg) [21] to measure the added perturbations. Given the original audio signal $x(t)$ and the adversarial perturbations $\delta(t)$ defined over the example index $t$, the SNRseg can be computed via

$$SNRseg(dB) = \frac{10}{K}\sum_{K=0}^{K-1}\log_{10}\frac{\sum_{t=Tk}^{Tk+T-1}x^2(t)}{\sum_{t=Tk}^{Tk+T-1}\delta^2(t)} \quad (9)$$

It can be obtained from Equation (9) that, the smaller the SNR value, the smaller the added perturbation.

Perceptual evaluation of speech quality (PESQ): PESQ [22] is an objective index of speech quality, and after the PESQ analysis, a score is given ranging from $-0.5$ to 4.5. A higher score means a better speech quality. The PESQ score is calculated from a stabilised ratio of the Bark spectral density degraded to the reference signal in each time–frequency cell.

PESQ, as a subjective speech quality evaluation method, can directly and genuinely reflect the real situation of speech quality, so it is a practical assessment index on the question of whether the adversarial example is inaudible.

### 2.2. Attack on ASRs

As a key technology of human–computer interface in information technology, an ASR system can convert raw human audio to text, which has is a great achievement and is widely used in various fields. Benefiting from the rapid development of deep neural networks, speech recognition has also made good progress. Existing ASR systems, like DeepSpeech2 [23] and Kaldi [24], have shown good recognition performance in the real physical world. Correspondingly, many state-of-the-art end-to-end speech recognition systems are represented by the transformers model [25] and CTC structure [26,27] also show good recognition performance. However, this also brings security risks to the speech recognition system, and there are recognition attack systems like [13,19,28–30], which force people to think more about the security of applications in the physical world.

As indicated in Figure 1, the ASR system can be attacked at the input stages. Furthermore, ASRs employ the Mel-frequency Cepstrum Coefficient (MFCC) [31] algorithm, for feature extraction, because of its ability to extrapolate important features, similar to the human ear. The feature vector is then sent to the model for either training or inferencing and obtains the recognized text.

There are many examples of successful attacks based on speech recognition systems. Carlini et al. [28] introduce the concept of adversarial examples to audio for the first time, inputting the waveform, which directly adds tiny distribution into the DeepSpeech model of the white-box, and using the gradient optimization FGSM method to generate targeted adversarial examples. Although this method develops adversarial examples for application in speech recognition systems for the first time, it is the only state-of-art technology. Therefore, In response to the situation encountered when the adversarial example is spread in the air, Yakura et al. [32] proposed the generation of countermeasure examples in a physical world on their basis. They simulated background noise with the white noise, the Bandpass filter, and impulse when the counter audio is attacked in the natural environment. The experimental results also achieved good results. However, the research is still in the experimental simulation stage, and a more extensive data set needs to be tested on this method.

Regarding attacks in a natural physical environment, CommanderSong [19] considered the distance of the attack in the real environment and whether the adversarial examples can be detected. For the first time, they used music to carry adversarial examples so that the adversarial examples can be hidden in the music without being noticed. Although this method is also under the white-box setting, it shows advantages in the attack results it produces.

This method of using music to carry adversarial examples has also been applied when others discuss the problem of black-box attacks in the real physical world. The authors

of [33] use more advanced but target-independent white-box models to approximate the target models, and their most essential idea is to take advantage of the transferability [34] of adversarial examples.

In the attack on the black-box speech recognition system, the authors of [13] first introduced the genetic algorithm to the black-box speech recognition system. The genetic algorithm is a method of solving optimization problems based on the principle of natural selection. The primary approach is to rely on biologically-inspired operators, such as mutation, crossover, and selection. At each step, the genetic algorithm will select some elite groups as the parents of the current population and, from them, the next generation will be generated. For each generation, it will retain the excellent genes of the previous generation so that after, the loop iteration, the optimal solution can be obtained. Since the black-box model does not know the system's parameter structure and other characteristics, the genetic algorithm is used as a gradient-independent algorithm that can iteratively apply noise to raw audio examples, pruning away poor performers at each generation, and ultimately end up with a troubled version of the input that successfully fooled a classification system, yet was still similar to the original audio. This attack was conducted on the Speech Commands classification model [13]. The genetic algorithm avoids attacking without prior knowledge, but the main problem of the genetic algorithm is that it requires a lot of calculation time to generate adversarial examples.

Extensive research was conducted by the research [14], which proposed a method combining genetic algorithms with gradient estimation. They also applied this method to a more complex DeepSpeech system. However, this method achieved a limited success rate with strict length restriction over the voices. The article shows that a high success rate is obtained at the cost of calculation.

For counterattacks in real-time [35], the real-time input system is a streaming problem. The previous attack methods only focused on the static input of the target model. However, the attacker cannot observe the entire original example for streaming input systems and can only receive past data. In [35], they use imitation learning and behavioral cloning algorithms for the whole of the problem and train real-time adversarial jamming generators through simulation demonstrations to obtain adversarial examples. The results prove the effectiveness of this method.

Although the above attacks achieve a high success rate through misleading classifiers, most of them exhibit a high degree of distortion, as they ignore the impact of hostile disturbances on human perception. Hence, the subsequent research [30] discussed and studied the inaudibility, robustness, and targetability of the adversarial examples, using the frequency in the acoustic model. The frequency masking effect is used for masking so that the user does not find that the system is being attacked. When calculating the masking threshold, the original method is optimized to make the masking threshold more accurate, strengthening the robustness and considering reverberation. Moreover the impact of the room impulse response, a robust and imperceptible directional adversarial example, was obtained. On Lingvo [36], a 100% success rate was obtained, but the method only remained at the laboratory stage and did not attempt an actual physical attack.

At the same time, as an attacker in a physical attack, sometimes one does not know when a user uses the attacking system and what the content of the system is. Based on this problem, AudiDoS [37] and other companies use a method to generate general interference in the external environment, continuously launching interference at any time to achieve the purpose of the attack. According to the experimental results, ASR uses the Librispeech data set, and an error rate of 78 appears. When an attack is needed, the solution is always to play a general interference to solve the problem.

In the latest black box attack, Ishida et al. [38] proposed to use multi-objective optimization to solve the problem of not knowing the prior information of the system. The first goal is the probability when the adversarial target example is the target mean we expect, the second goal is the variance. The third objective function is that the gap between the MFCC of the adversarial example and the MFCC of the original audio should be the

smallest. In this article, the use of genetic computing is compared with the first black box attack. Their proposed method uses an automatic adjustment method to obtain adversarial examples. Compared with the mutation of the genetic algorithm, it is easier to obtain adversarial examples using this method.

### 2.3. Attack on Speaker Recognition System

Different from speech recognition systems, speaker recognition mainly focuses on extracting individual dependent voice characteristics through embedding methods to identify speakers' identities regardless of their speech content. Specifically, Speaker recognition systems (SRS) [39–41] can be developed either for identification or verification [42] of individuals from their speech. In a closed set speaker identification scenario [42,43], we are provided with train and test utterances from a set of unique speakers. The task is to train a model that, given a test utterance, can classify it to one of the training speakers. Speaker verification [40,41], on the other hand, is an open set problem. A more straightforward introduction, the former is to determine one of the $N$ reference speakers according to the speaker's voice, which is a selection problem; the latter is to verify whether the speaker's identity is consistent with its declaration, which is a decision problem of choosing one or the other.

Speaker recognition system has became one of the indispensable technologies in biometric identification and other fields. Unfortunately, speaker recognition also brings security concerns because of the adoption of deep neural networks. Studies specifically focused on adversarial attacks on SRS have come up only very recently [44] and some of this work has revealed new, potential threats. Apart from studying robust spoofing countermeasures, it is important to study the weak links of attacks.

The adversarial attack on SRS aims to generate adversarial examples from the voice of a specific source speaker so that the attacked system will misclassify it as a registered speaker (non-target attack) or target speaker (targeted attack), but it is still correctly identified as a source speaker by ordinary users. Due to the significant attack results of audio adversarial examples in speech recognition, there have been many successful attacks in speaker recognition. The first is that [45] used the FGSM method to generate adversarial examples with the MFCC under the prior knowledge of the speaker recognition system and achieved a high attack success rate based on an end-to-end DNN-based speaker verification system. This study demonstrated the ability of the adversarial attack to deceive the automatic speaker verification (ASV) system.

Wang et al. [46] crafted the adversarial examples using FGSM and Local Distributional Smoothness (LDS) [47] to attack the well-trained speech verification model and proposed an adversarial regularization method by the adversarial examples. From the perspective of the attacker, although this approach improves the performance of the speaker verification system, the FGSM method is employed to attack, and similar to the above method [45].

In [48], inspired by imperceptible adversarial examples in a white-box environment, they make the best use of the psychoacoustic principle of frequency masking and constrict the perturbation under the masking threshold of original speech to produce targeted and imperceptible confrontational examples directly to the raw waveform. Furthermore they achieved a 98.5% success rate. In addition, they also tried to use music in non-human languages to attack, this also achieving good attack results, but these methods are still in the development stage.

In this study [49], the method of FGSM is used to produce a white box system based on GMM i-vector speaker certification, and the portability of the adversarial examples is verified in different networks. The adversarial examples of various features and vector models are tested in this paper, proving that the characteristics of an adversarial example can be used directly for an attack. However, the authors did not attack more recent DNN-based speaker recognition frameworks shown to have state-of-art performances.

In addition to the methods mentioned above, recently, the authors of [50] have borrowed from the technical methods of speech steganography, using a gated convolutional

autoencoder [51] to generate imperceptible audio adversarial examples, and using a multi-objective loss function for training. This method bypasses the gradient requirements as the general generation of adversarial examples. This network is used to generate adversarial examples. From the results, the PESQ score of 4.30 for targeted and non-targeted adversarial examples can achieve a high success rate.

In a real physical environment, and for the success of the attack, it is necessary to consider factors such as reverberation, noise, etc. Therefore, the authors of [52,53] base their work on the white box, which solves the impact of the room impulse response on the adversarial example in the actual attack. However, in comparison, the study [52] also considers that non-universal adversarial examples require a lot of time to train and the real-time nature of the attack—they both utilized a gradient optimization method to generate robust universal adversarial examples. However, both were still in the experimental stage and did not attack in the real world.

In the black-box setting, to take advantage of practical gradient information for gradient descent, to solve the optimization problem of adversarial example generation, Guangke et al. [54] used the gradient estimation algorithm (NES) based on the evolution strategy. They found an impressive attack success rate on Kaldi with their proposed method, but the problem of this article is that all the test data sets are small. As their study only included five speakers, an extensive study with a much higher number of test speakers is still needed.

The above content briefly describes the existing adversarial attack methods that some authors believe mainly occur speech recognition and speaker recognition. Researchers firstly start from the generation of audio adversarial examples to the generation of aggressive adversarial examples which to the human auditory system are imperceptible, and consequently many adversarial attack methods have been trialled successfully. Although many methods have achieved high attack success rates, each method still has comparatively more or less shortcomings.

### *2.4. Defence against Adversarial Attack*

Both the positive and negative aspects of contradiction are symbiotic. Since neural networks are vulnerable to attacks from adversarial examples, adversarial attacks pose a new threat to the security of DNN-based sound processing systems. Therefore, effective defense is very necessary.

To a certain extent, defense and attack are similar and they both need to find loopholes in the network. The difference is that defense performs repair and protection for loopholes, and attacks play a destructive role. The defense against adversarial attacks is mainly carried out in the field of pictures, and there are many successful cases of defense [55–57]. However, not all of these methods can be applied to audio. The main difficulties are as follows: first of all, in terms of the coverage of digital representation, the range after audio sampling in terms of amplitude is much wider than the range of (0, 256) of images. Secondly, in terms of content repeatability, audio is more complicated. The performance is much higher than that of pictures, which makes audio more sensitive than images, which makes it difficult to protect the original audio in defense.

The authors of Ref. [58] first discussed the robustness of the targeted adversarial audio examples generated in [28]. Their hypothetical confrontational voice is fragile. Given adversarial audio, the two parts will be transcribed separately if the audio is divided into two parts. The result is that spliced transcription is very different from the original intact transcription.

Based on the above ideas, it is natural to consider that the most significant feature of adversarial examples is that disturbances are added to the original audio, and these disturbances can also be quantified [10]. Therefore, a simple idea in the defense process is to remove the disturbances; in [59], the input transformer method of image adversarial attacks is used to deal with the disturbances in the audio adversarial examples, which dramatically reduces the attack rate of adversarial examples but, at the same time, they also have

weaknesses that cannot be ignored. This method is also very fragile when the adversarial example is generated in consideration of the problem of gradient disappearance [60]. Therefore, the author also proposed a time-domain-independent method to detect whether the input audio is an adversarial example. The author was famous at the time and detailed experiments have been performed in the confrontation environment for different white box or black box attack methods, and good results were achieved in the detection of confrontation audio.

The authors of [61] are inspired by MVP, and the phenomenon in which an adversarial audio inputting into different speech recognition systems has different recognition results. Combined with the characteristics of MVP, the authors proposed to input one audio into multiple different ASR systems and then performed the output results. The similarity is calculated and passed through a two-classifier to determine whether the input audio is adversarial in order to achieve anti-audio detection.

The authors of [62] used an audio modification method to detect adversarial samples. Firstly, the method has two steps: verify the initial audio samples against the recognition system and present the initial classification results. Subsequently, a modified audio signal is generated by audio modification of the initial audio sample. The generated modified audio signal is compared with the classification result of the original audio sample. If the classification results differ significantly, the initial audio sample is regarded an adversarial example. If the difference is slight, the initial audio sample is considered to be the original sample. The original audio and adversarial examples were passed through simultaneously, and the frequency spectrum and waveform of the audio were analyzed in this method. The experimental results show that the CW method at the laboratory detection level is effective in a DeepSpeech attack.

## 3. Attack Threat Model Taxonomy

In this section, depending on the adversary's background, prior knowledge, etc., we introduce the existing attack models in VPSes and classify them, and we hope to build an overall attack framework for comparison in future research. In addition, we also list some existing attack methods in Table 1 for intuitive understanding.

**Table 1.** The taxonomy of attack in speaker and speech recognition. 'Box' indicates the prior knowledge which the attacker master, which can be categorized by a white box, a gray box and a black box. 'Platform' is the certain attacked system. 'System' means the targeted system, especially the voice control system (VCs); 'Real/Simulated' shows whether the attack is in the real physical world.

| Work | Year | Box | Target | Platform | Corpus | System | Real/Simulated | Feature |
|---|---|---|---|---|---|---|---|---|
| [63] | 2017 | Black/White | Both | DeepSpeech2 | Librispeech | ASR | Simulated | STFT |
| [28] | 2018 | White | Targeted | DeepSpeech | MCVD [64] | ASR | Simulated | Waveform |
| [19] | 2018 | White | non-target | Kaldi | Random choice | ASR | Both | Waveform |
| [45] | 2018 | White | Targeted | ASV | YOHO [65] | ASV | Simulated | MFCC |
| [32] | 2018 | White | Targeted | Deepspeech | Music Clips | ASR | Real | waveform |
| [13] | 2018 | Black | non-target | CNNs [66] | Speech commands | ASR | Simulated | Waveform |
| [29] | 2018 | White | Targeted | Kaldi | WSJ | ASR | Simulated | STFT |
| [45] | 2018 | White | non-target | SVs [39] | YOHO [65] | ASV | Simulated | MFCC/Mel-Spectrum |
| [14] | 2019 | Black | targeted | DeepSpeech | MCVD | ASR | Simulated | Waveform |
| [15] | 2019 | Grey | non-target | Kaldi | TTSREADER | ASR | Simulated | Waveform |
| [67] | 2019 | Grey | Both | Alexa [68] | LJ [69] | VAs | Both | Waveform |
| [30] | 2019 | White | Targeted | Lingvo [36] | Librispeech [70] | ASR | Simulated | Waveform |
| [37] | 2019 | White | non-target | DeepSpeech | Librispeech | ASR | Simulated | Waveform |
| [54] | 2019 | Black | Targeted | Talentedsoft [71] | Voxceleb1 [70,72] | SRS | Real | Waveform |
| [46] | 2019 | White | non-target | SVs [73] | NTIMIT [74] | ASV | Simulated | feature |
| [75] | 2019 | White | Targeted | VGGVox [72] | Voxceleb | SRS | Simulated | Spectrogram |
| [76] | 2020 | White | non-target | CTC-based [66] | MCVD | ASR | Simulated | MFCC |
| [52] | 2020 | Grey | non-target | VCs [66] | Speech commands [77] | VCs | Simulated | Waveform |
| [33] | 2020 | Black | Targeted | Alex, Google, Cortana | CommanderSong [19] | ASR | Real | Spectrum |
| [48] | 2020 | White | Targeted | DNNs [40] | Aishell-1 [78] | SRS | Simulated | Waveform |
| [49] | 2020 | Both | non-target | SVs [79] | Voxceleb1 | ASV | Simulated | MFCC |
| [53] | 2020 | White | Both | Kaldi | CSTR VCTK [80] | SRS | Both | MFCC |
| [52] | 2020 | White | non-target | DNNs [40] | CSTR VCTK | SRS | Simulated | MFCC |
| [81] | 2020 | Gray | Targeted | SincNet [82] | NTIMIT/Librispeech | SRS | Simulated | Waveform |
| [50] | 2021 | White | Both | DNNs [40] | Voxceleb | SRS | Simulated | Waveform |
| [38] | 2021 | Black | Targeted | CNNs | Speech Command [66] | VCs | Simulated | MFCC |

### 3.1. Adversarial Knowledge

Whether to master the prior knowledge of the attacking system during the attack is vital information for the attacker. The grasp of prior knowledge has a significant impact on the efficiency and success rate of the attack. Therefore, according to the different levels of the attacker's prior knowledge of the speech system, the method of attack can be divided into the following categories:

White-box: When the attacker has full knowledge of the target system, including the model type, network architecture, and the values of all the parameters and training weights, etc., we call that such an attack one carried out in a white-box setting. An example of this access-type is an open-source model like DeepSpeech [83] and Kaldi [24]. Many white-box attack methods [13,19,28,29,45,46,52,76] have shown to a certain extent their great advantages in the context of prior knowledge. The white-box attack method of attack is an important foundation for our research, but it has limitations in real attacks.

Black-box: Conversely, when the attacker has almost no prior information about the system, we only know what the task of the system is. For example, we only know that Siri [84] is a system that can recognize speech, but its model, parameters, etc. are completely unknown. There are also many attack methods against black boxes like [13,14,33,54]. Considering the actual attack environment, the black box situation is reasonable.

Grey-box: Gray box is a situation in which we only know some of the parameters of the model, network architecture, etc., but not all of them. A typical example is the Azure Speaker Recognition model [85], in which we only understand its task category and its filter extraction but are limited with other information. The method similar to those addressed in [15,52,67,81] attacked the grey-box and achieved great performance.

In a black-box setting, when an attacker only has access to the logits or outputs, it is much harder to consistently create successful adversarial attacks. In specific unique black box settings, white box attack methods can be reused if an attacker creates a model that approximates the original targeted model. However, even though attacks can transfer across networks for some domains, this requires more knowledge of solving the task that the original model is solving than an attacker may have.

### 3.2. Adversarial Goal

Considering whether the adversarial example needs to become the excepted output after passing through the system, we group the attack into two different attacks.

Non-targeted: Non-targeted means that an arbitrary result that is different from the original output can be generated after the adversarial example passes through the attacked system. For example, in [13,15,19,76], the attacker only requires the system is to recognize as other text content. Non-target adversarial examples are usually generated in two ways: (1) perform several targeted attacks, obtain the least interference attack from the results; (2) maximize the possibility of misclassification.

Targeted: The adversarial example is required to output a desired result through the system. In other words, for the input $x$ and the network model $f(\cdot)$, the output $\tilde{y} = f(x + \delta)$ is the expected result. Using a simple example to illustrate, the correct text information of the audio we originally entered is "Hello", but the output converts it into "World" after the perturbation is added. Correspondingly, the non-target attack only requires output is not "Hello". During the past few years, many related works [4,32,33] have emerged to take advantage of the targeted adversarial example to finish the task of attack.

Targeted adversarial examples help the attacker obtain the desired output, especially in critical situations such as identifying a specific person whose information to tamper with. However, the targeted adversarial examples require more calculations to solve than the non-target adversarial examples. The degree and calculation time are longer, which also puts forward higher requirements for the attack. Non-target adversarial examples are easier to implement because they have more options and space to directly aim at the output.

### 3.3. Adversarial Perturbation Scope

The classification of attack also can be divided into two categories: individual and universal attack. This is mainly based on the scope of application of the perturbation added when the researcher generates the adversarial examples.

Individual: Individual attacks usually craft different perturbations for each clean input audio. The solution of the adversarial examples of this attack is often a conditional optimization problem. Therefore, the gradient optimization problem (FGSM) is often used in the white-box system, and the genetic algorithm is used to generate the problem in the black-box system.

Universal: This example is not limited to a specific example; it is suitable for all audio that needs to be attacked, so it has a huge advantage regarding time, not only in the process of adversarial example generation but also solving the problem of the attacker not knowing when to attack [37]. However, the ensuing problem is that the accuracy of the adversarial example is not high enough, and the disturbance may be relatively large.

Most current attacks generate adversarial examples individually. However, the pervasive perturbation makes it easier to fight against examples in the real world. When the input example changes, the adversary can achieve the purpose of the attack without changing the disturbance.

### 3.4. Real or Simulated World

Based on the classification of the attack environment, it can be divided into the internal laboratory and the real physics. The two complement each other and promote each other.

Real world: Real world means that the researchers put the attack in the real physical situation. When we consider the attack in the real physical world, whether from the propagation medium of the adversarial examples or the distance from the adversarial example attack, the physical Attacks in the world are relatively brutal to complete. Ref. [32] simulates the human ear effect, impulse response, and other issues that need to be considered when the adversarial example is spread in the air. The point of attack distance is also considered in [19] and the ideal environment is to perform close-range attacks without being noticed by the user.

Simulated world: Simulated world means an attack on the laboratory stage. If it is not attacked in a real physical environment, we can use a computer to simulate an attack. Many of the existing studies are on the state-of-art, and some are forward-looking studies to lay the foundation for future research. More are towards being able to attack this target in a real environment, in [32] The first is to consider the impact of room impulse response that may be encountered in an attack in a real environment. Simulations were carried out at the laboratory stage, but it laid a good foundation for subsequent real environment attacks.

## 4. Defense

This section will categorize defense methods in the audio domain from different perspectives, hoping to give a systematic classification method. Figure 3 shows that researchers usually use audio data processing and network model retraining to achieve the purpose of adversarial defense. Furthermore, we also listed some existing representative adversarial defense methods in Table 2.

### 4.1. Defensive Result

The results of defense are often more important to us. Whether the security of the targeted system is successful can be directly expressed from the effects of defense. Therefore, we can categorize the defense methods of the sound processing system from the perspective of whether it can ultimately withstand the attack:

Detecting: When we detect adversarial examples, we only completed the task of discovering adversarial examples. From the resulting perspective, adversarial examples can still produce wrong output results when passing through the target system. Detecting is also called an incomplete defense. In the testing process, the researchers transformed the

problem of detecting adversarial examples into a crisis of binary classification. In these kind of paper, the author obtains a reference example by using methods such as Multi-version programming (MVP) [61], CNN-based [86], dropout uncertainty [87] and uncertainty quantification [88] to compare the similarity with the adversarial example. If the similarity between the input audio and the reference audio is low, it means that the input audio is an adversarial example.

Complete defense: Unlike detection, the complete defense can still obtain correct recognition results in the adversarial sample input. In [46,59,89], the authors mainly use techniques such as quantization, local smoothing, down-sampling, and automatic encoding of the input audio signal to fight against the slight disturbance in the sample is eliminated. Audio that can be output correctly is obtained.

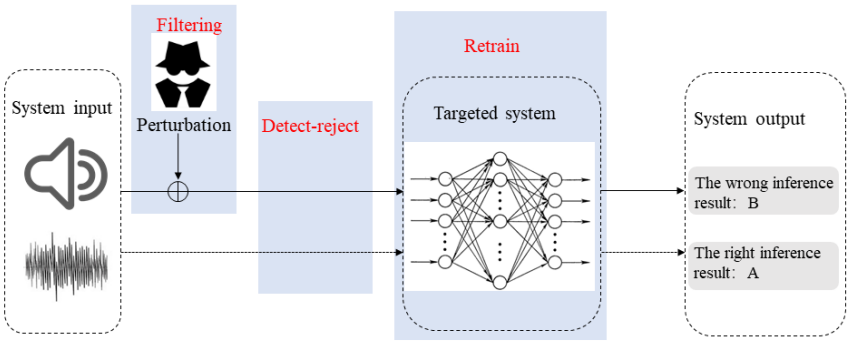


**Figure 3.** The general method of defense including filtering, detect–reject, retrain.

### *4.2. Classification from the Content of Defense*

In the process of defense, we can process the data in two stages before and after the input to achieve the effect of protection:

Data preprocessing: The data modification is that which modifies the training set in the training stage or the input data in the test stage through adversarial training, gradient hiding, transferability blocking, audio data compression [89], data randomization, etc. The adversarial example passes through the system after preprocessing and achieves the correct result to achieve an invalid attack of the target system.

Model modified: Modification of the network model is performed, which adjusts the target model directly to increase its robustness. Commonly used methods are through regularization [46], audio squeezing [59], or audio turbulence [19]. By modifying the network model, the protected system can be more robust, making the system less damaged by adversarial attacks. Studies such as [19,46] have also achieved more significant results in this regard.

**Table 2.** The taxonomy of defense in speaker and speech recognition. 'Task' indices whether the defense method is completely defense or detection. An adversarial example is the generating method of the attack method.

| Work | Year | Defense Method | Task | System | Adversarial Example |
|---|---|---|---|---|---|
| [59] | 2018 | Temporal dependency | Detecting | ASR | Genetic algorithm [13]/FGSM/ Commander Song [19] |
| [46] | 2019 | Adversarial regularization | Defense | ASV | FGSM/LDS |
| [61] | 2019 | MVP-$E_{ASR}$ | Detecting | ASR | FGSM |
| [62] | 2019 | Audio modification | Detecting | ASR | Carlini and Wagner Attacks [4] |
| [90] | 2020 | Adversarial training/Spatial smoothing | Defense | ASV | Projected Gradient Descent Method [91] |
| [92] | 2020 | Self-attention U-Net | Defense | ASR | FGSM/Evolutionary optimization [93] |
| [94] | 2021 | Hybrid adversarial training | Defense | SRS | FGSM |
| [95] | 2021 | Audio transformation | Detecting | ASR | Adaptive attack algorithm [95] |
| [96] | 2021 | Self-supervised learning model [97] | Defense | ASV | BIM |

### 4.3. From Different Areas of Defense Methods

Due to the threat of adversarial attacks to ASR and ASV systems based on deep neural networks, researchers have proposed different methods of defense against adversarial attacks from different sound research fields, mainly the including detect–rejection, filtering and retraining, as shown in Figure 3.

Detect–rejection: This method is from the perspective of the ASV field. Since the role of the speaker authentication system itself is to confirm whether the input audio is the target speaker, it is an alternative problem. Based on this idea, the input of the adversarial sample into the ASV system is in the original In the case of prior knowledge, and the adversarial examples can be output as a type of rejection so as to protect the security of the system. Recently, in [90,96,98], the use of ASV systems to detect adversarial examples has achieved remarkable results.

Filtering: Considering that the essential task of speech enhancement is to remove the mixed noise in the input audio, and the essence of the adversarial sample is to add disturbance to the pure audio, so the researchers target the added perturbation from the perspective of speech enhancement, using speech separation and the method of speech enhancement to eliminate the added disturbance; ref. [92] adopted the self-attention U-Net method to enhance the ASR system in the face of adversarial attacks.

Retraing: The retraining method is to fine-tune the network by augmenting the collected or simulated adversarial audio clips in a training set with explicit labels of noise. Based on the DNN adaptation fundamentals, it can make the network more robust to similar attacks. Wang et al. [99] generate adversarial examples using the fast gradient signal method(FGSM) and use these adversarial examples as augmented data to retrain the model to enhance the robustness of KWS model.

## 5. Future Working

Based on the existing literature, topics such as the large amount of calculation and space storage required in generating adversarial examples, the practical application of adversarial examples also lacking corresponding explanatory nature and the method of black-box attack are gradually adapting into long-standing problems of the attack direction. Therefore, we expect there will be a multitude of works directed to address these problems in the future to improve the real-time performance of an attacker. From the perspective of the defender, the adversarial defense is of great significance for increasing the robustness of the system, and we look forward to seeing the integration with other directions in future work, such as speech separation, speech conversion speech scene analysis, etc.

## 6. Conclusions

In this article, we researched adversarial attacks and defenses from the perspective of the security of the ASR and ASV systems in voice processing systems. We first demonstrated the generation methods of adversarial examples, psychoacoustics models, and existing forms of attack and defense against ASR and SRS. Secondly, we systematically classified adversarial attacks and, last but not least, analyzed the existing defense methods from the different defense perspectives.

In future work, this article systematically organizes the methods of adversarial attacks and defenses, laying a solid foundation for subsequent research, and will further follow-up research, which will focus on the adversarial attack and defense of the voice processing systems in the physical world.

**Author Contributions:** Conceptualization, X.C. and S.L.; methodology, X.C. and S.L.; formal analysis, X.C., S.L., H.H; investigation, X.C., S.L. and H.H.; resources, S.L. and H.H.; writing—original draft preparation, X.C.; writing—review and editing, X.C., S.L. and H.H.; visualization, X.C.; supervision, S.L. and H.H.; funding acquisition, S.L. and H.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Opening Project of Key Laboratory of Xinjiang Uyghur Autonomous Region, China (2020D04047); National Natural Science function of China (61663044, 61761041); NICT international funding, Japan and JSPS KAKENHI Grant No. 21K17837, Japan.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Singh, S. Forensic and automatic speaker recognition system. *Int. J. Electr. Comput. Eng.* **2018**, *8*, 2804. [CrossRef]
2. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
3. Sarkar, S.; Bansal, A.; Mahbub, U.; Chellappa, R. UPSET and ANGRI: Breaking high performance image classifiers. *arXiv* **2017**, arXiv:1707.01159.
4. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 39–57.
5. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
6. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
7. Xiao, Q.; Chen, Y.; Shen, C.; Chen, Y.; Li, K. Seeing is not believing: Camouflage attacks on image scaling algorithms. In Proceedings of the 28th {USENIX} Security Symposium ({USENIX} Security 19), Santa Clara, CA, USA, 14–16 August 2019; pp. 443–460.
8. Lin, Y.; Abdulla, W.H. *Principles of Psychoacoustics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 15–49.
9. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. *arXiv* **2016**, arXiv:1611.01236.
10. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
11. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrücken, Germany, 21–24 March 2016; pp. 372–387.
12. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1765–1773.
13. Alzantot, M.; Balaji, B.; Srivastava, M. Did you hear that? adversarial examples against automatic speech recognition. *arXiv* **2018**, arXiv:1801.00554.
14. Taori, R.; Kamsetty, A.; Chu, B.; Vemuri, N. Targeted adversarial examples for black box audio systems. In Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW), Francisco, CA, USA, 19–23 May 2019; pp. 15–20.
15. Wu, Y.; Liu, J.; Chen, Y.; Cheng, J. Semi-black-box attacks against speech recognition systems using adversarial samples. In Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 11–14 November 2019; pp. 1–5.
16. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
17. Abdullah, H.; Garcia, W.; Peeters, C.; Traynor, P.; Butler, K.R.; Wilson, J. Practical hidden voice attacks against speech and speaker recognition systems. *arXiv* **2019**, arXiv:1904.05734.
18. Cherry, E.C. Some experiments on the recognition of speech, with one and with two ears. *J. Acoust. Soc. Am.* **1953**, *25*, 975–979. [CrossRef]
19. Yuan, X.; Chen, Y.; Zhao, Y.; Long, Y.; Liu, X.; Chen, K.; Zhang, S.; Huang, H.; Wang, X.; Gunter, C.A. Commandersong: A systematic approach for practical adversarial voice recognition. In Proceedings of the 27th {USENIX} Security Symposium ({USENIX} Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 49–64.
20. Navarro, G. A guided tour to approximate string matching. *ACM Comput. Surv. (CSUR)* **2001**, *33*, 31–88. [CrossRef]
21. Yook, S.; Nam, K.W.; Kim, H.; Kwon, S.Y.; Kim, D.; Lee, S.; Hong, S.H.; Jang, D.P.; Kim, I.Y. Modified segmental signal-to-noise ratio reflecting spectral masking effect for evaluating the performance of hearing aid algorithms. *Speech Commun.* **2013**, *55*, 1003–1010. [CrossRef]
22. Rix, A.W.; Beerends, J.G.; Hollier, M.P.; Hekstra, A.P. Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs. In Proceedings of the IEEE-ICASSP, Salt Lake City, UT, USA, 7–11 May 2001; Volume 2, pp. 749–752.
23. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 173–182.

24. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Waikoloa, HI, USA, 11–15 December 2011.
25. Dong, L.; Xu, S.; Xu, B. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5884–5888.
26. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
27. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 4835–4839.
28. Carlini, N.; Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 1–7.
29. Schönherr, L.; Kohls, K.; Zeiler, S.; Holz, T.; Kolossa, D. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv* **2018**, arXiv:1808.05665.
30. Qin, Y.; Carlini, N.; Cottrell, G.; Goodfellow, I.; Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 5231–5240.
31. Muda, L.; Begam, M.; Elamvazuthi, I. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv* **2010**, arXiv:1003.4083.
32. Yakura, H.; Sakuma, J. Robust audio adversarial example for a physical attack. *arXiv* **2018**, arXiv:1810.11793.
33. Chen, Y.; Yuan, X.; Zhang, J.; Zhao, Y.; Zhang, S.; Chen, K.; Wang, X. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In Proceedings of the 29th {USENIX} Security Symposium ({USENIX} Security 20), Santa Clara, CA, USA, 12–14 August 2020; pp. 2667–2684.
34. Papernot, N.; McDaniel, P.; Goodfellow, I. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *arXiv* **2016**, arXiv:1605.07277.
35. Gong, Y.; Li, B.; Poellabauer, C.; Shi, Y. Real-time adversarial attacks. *arXiv* **2019**, arXiv:1905.13399.
36. Shen, J.; Nguyen, P.; Wu, Y.; Chen, Z.; Chen, M.X.; Jia, Y.; Kannan, A.; Sainath, T.; Cao, Y.; Chiu, C.C.; et al. Lingvo: A modular and scalable framework for sequence-to-sequence modeling. *arXiv* **2019**, arXiv:1902.08295.
37. Gong, T.; Ramos, A.G.C.; Bhattacharya, S.; Mathur, A.; Kawsar, F. AudiDoS: Real-Time Denial-of-Service Adversarial Attacks on Deep Audio Models. In Proceedings of the IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 978–985.
38. Ishida, S.; Ono, S. Adjust-free adversarial example generation in speech recognition using evolutionary multi-objective optimization under black-box condition. *Artif. Life Robot.* **2021**, *26*, 243–249. [CrossRef]
39. Heigold, G.; Moreno, I.; Bengio, S.; Shazeer, N. End-to-end text-dependent speaker verification. In Proceedings of the IEEE-ICASSP, Shanghai, China, 20–25 March 2016; pp. 5115–5119.
40. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In Proceedings of the IEEE-ICASSP, Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
41. Chung, J.S.; Nagrani, A.; Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv* **2018**, arXiv:1806.05622.
42. Hansen, J.H.; Hasan, T. Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Process. Mag.* **2015**, *32*, 74–99. [CrossRef]
43. Jati, A.; Georgiou, P. Neural predictive coding using convolutional neural networks toward unsupervised learning of speaker characteristics. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 1577–1589. [CrossRef]
44. Jati, A.; Hsu, C.C.; Pal, M.; Peri, R.; AbdAlmageed, W.; Narayanan, S. Adversarial attack and defense strategies for deep speaker recognition systems. *Comput. Speech Lang.* **2021**, *68*, 101199. [CrossRef]
45. Kreuk, F.; Adi, Y.; Cisse, M.; Keshet, J. Fooling end-to-end speaker verification with adversarial examples. In Proceedings of the IEEE-ICASSP, Calgary, AB, Canada, 15–20 April 2018; pp. 1962–1966.
46. Wang, Q.; Guo, P.; Sun, S.; Xie, L.; Hansen, J. Adversarial Regularization for End-to-End Robust Speaker Verification. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 4010–4014.
47. Miyato, T.; Maeda, S.i.; Koyama, M.; Nakae, K.; Ishii, S. Distributional smoothing with virtual adversarial training. *arXiv* **2015**, arXiv:1507.00677.
48. Wang, Q.; Guo, P.; Xie, L. Inaudible adversarial perturbations for targeted attack in speaker recognition. *arXiv* **2020**, arXiv:2005.10637.
49. Li, X.; Zhong, J.; Wu, X.; Yu, J.; Liu, X.; Meng, H. Adversarial attacks on GMM i-vector based speaker verification systems. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020; pp. 6579–6583.
50. Shamsabadi, A.S.; Teixeira, F.S.; Abad, A.; Raj, B.; Cavallaro, A.; Trancoso, I. FoolHD: Fooling Speaker Identification by Highly Imperceptible Adversarial Disturbances. In Proceedings of the IEEE-ICASSP, Toronto, ON, Canada, 6–11 June 2021; pp. 6159–6163.

51. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 933–941.
52. Xie, Y.; Shi, C.; Li, Z.; Liu, J.; Chen, Y.; Yuan, B. Real-Time, Universal, and Robust Adversarial Attacks Against Speaker Recognition Systems. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020; pp. 1738–1742.
53. Li, Z.; Shi, C.; Xie, Y.; Liu, J.; Yuan, B.; Chen, Y. Practical adversarial attacks against speaker recognition systems. In Proceedings of the International Workshop on Mobile Computing Systems and Applications, Austin, TX, USA, 3–4 March 2020; pp. 9–14.
54. Chen, G.; Chen, S.; Fan, L.; Du, X.; Zhao, Z.; Song, F.; Liu, Y. Who is real bob? adversarial attacks on speaker recognition systems. *arXiv* **2019**, arXiv:1911.01840.
55. Zuo, F.; Luo, L.; Zeng, Q. Countermeasures Against L0 Adversarial Examples Using Image in Processing and Siamese Networks. *arXiv* **2018**, arXiv:1812.09638.
56. Raghunathan, A.; Steinhardt, J.; Liang, P. Certified defenses against adversarial examples. *arXiv* **2018**, arXiv:1801.09344.
57. Xu, W.; Evans, D.; Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv* **2017**, arXiv:1704.01155.
58. Yang, Z.; Li, B.; Chen, P.Y.; Song, D. Towards mitigating audio adversarial perturbations. In Proceedings of the ICLR 2018 Workshop Submission, Vancouver, BC, Canada, 30 April–3 May 2018.
59. Yang, Z.; Li, B.; Chen, P.Y.; Song, D. Characterizing audio adversarial examples using temporal dependency. *arXiv* **2018**, arXiv:1809.10875.
60. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 274–283.
61. Zeng, Q.; Su, J.; Fu, C.; Kayas, G.; Luo, L.; Du, X.; Tan, C.C.; Wu, J. A Multiversion Programming Inspired Approach to Detecting Audio Adversarial Examples. In Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Portland, OR, USA, 24–27 June 2019.
62. Kwon, H.; Yoon, H.; Park, K.W. POSTER: Detecting audio adversarial example through audio modification. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2521–2523.
63. Cisse, M.; Adi, Y.; Neverova, N.; Keshet, J. Houdini: Fooling deep structured prediction models. *arXiv* **2017**, arXiv:1707.05373.
64. Mozilla Common Voice (MVCD) Dataset2. Available online: https://commonvoice.mozilla.org/en/datasets (accessed on 16 May 2021).
65. Campbell, J.P. Testing with the YOHO CD-ROM voice verification corpus. *Proc. Int. Conf. Acoust. Speech Signal Process.* **1995**, *1*, 341–344.
66. Sainath, T.N.; Parada, C. Convolutional neural networks for small-footprint keyword spotting. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
67. Li, J.B.; Qu, S.; Li, X.; Szurley, J.; Kolter, J.Z.; Metze, F. Adversarial music: Real world audio adversary against wake-word detection system. *arXiv* **2019**, arXiv:1911.00126.
68. Amazon Alex. Available online: https://developer.amazon.com/en-US/alexa (accessed on 8 May 2021).
69. The LJ Speech Dataset. Available online: https://keithito.com/LJ-Speech-Dataset/ (accessed on 16 May 2021).
70. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In Proceedings of the IEEE-ICASSP, South Brisbane, QL, Australia, 19–24 April 2015; pp. 5206–5210.
71. The Talentedsoft. Available online: http://www.talentedsoft.com (accessed on 16 May 2021).
72. Nagrani, A.; Chung, J.S.; Zisserman, A. Voxceleb: A large-scale speaker identification dataset. *arXiv* **2017**, arXiv:1706.08612.
73. Wan, L.; Wang, Q.; Papir, A.; Moreno, I.L. Generalized end-to-end loss for speaker verification. In Proceedings of the IEEE-ICASSP, Calgary, AB, Canada, 15–20 April 2018; pp. 4879–4883.
74. Jankowski, C.; Kalyanswamy, A.; Basson, S.; Spitz, J. NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database. In Proceedings of the International Conference on Acoustics, Speech, and Signal in Processing, Albuquerque, NM, USA, 3–6 April 1990; pp. 109–112.
75. Marras, M.; Korus, P.; Memon, N.D.; Fenu, G. Adversarial Optimization for Dictionary Attacks on Speaker Verification. In Proceedings of the INTERSPEECH, Graz, Austria, 15–19 September 2019; pp. 2913–2917.
76. Liu, X.; Wan, K.; Ding, Y.; Zhang, X.; Zhu, Q. Weighted-sampling audio adversarial example attack. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 4908–4915.
77. Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv* **2018**, arXiv:1804.03209.
78. Bu, H.; Du, J.; Na, X.; Wu, B.; Zheng, H. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In Proceedings of the Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA), Seoul, Korea, 1–3 November 2017; pp. 1–5.
79. Dehak, N.; Kenny, P.J.; Dehak, R.; Dumouchel, P.; Ouellet, P. Front-end factor analysis for speaker verification. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *19*, 788–798. [CrossRef]
80. Yamagishi, J.; Veaux, C.; MacDonald, K. *CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit (Version 0.92)*; University of Edinburgh: Edinburgh, UK, 2019.
81. Li, J.; Zhang, X.; Jia, C.; Xu, J.; Zhang, L.; Wang, Y.; Ma, S.; Gao, W. Universal adversarial perturbations generative network for speaker recognition. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020; pp. 1–6.
82. Ravanelli, M.; Bengio, Y. Speaker recognition from raw waveform with sincnet. In Proceedings of the IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 1021–1028.

83. Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv* **2014**, arXiv:1412.5567.
84. Apple's Siri. Available online: https://www.apple.com/in/siri/ (accessed on 8 May 2021).
85. Azure Speaker Identification API. Available online: https://azure.microsoft.com/en-us/services/cognitive-services/speaker-recognition/ (accessed on 8 May 2021).
86. Samizade, S.; Tan, Z.H.; Shen, C.; Guan, X. Adversarial Example Detection by Classification for Deep Speech Recognition. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020.
87. Jayashankar, T.; Roux, J.L.; Moulin, P. Detecting Audio Attacks on ASR Systems with Dropout Uncertainty. *arXiv* **2020**, arXiv:2006.01906.
88. Däubener, S.; Schönherr, L.; Fischer, A.; Kolossa, D. Detecting adversarial examples for speech recognition via uncertainty quantification. *arXiv* **2020**, arXiv:2005.14611.
89. Das, N.; Shanbhogue, M.; Chen, S.T.; Chen, L.; Kounavis, M.E.; Chau, D.H. Adagio: Interactive experimentation with adversarial attack and defense for audio. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, 10–14 September 2018; pp. 677–681.
90. Wu, H.; Liu, S.; Meng, H.; Lee, H.y. Defense against adversarial attacks on spoofing countermeasures of ASV. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020; pp. 6564–6568.
91. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
92. Yang, C.H.; Qi, J.; Chen, P.Y.; Ma, X.; Lee, C.H. Characterizing speech adversarial examples using self-attention u-net enhancement. In Proceedings of the IEEE-ICASSP, Barcelona, Spain, 4–8 May 2020; pp. 3107–3111.
93. Khare, S.; Aralikatte, R.; Mani, S. Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization. *arXiv* **2018**, arXiv:1811.01312.
94. Pal, M.; Jati, A.; Peri, R.; Hsu, C.C.; AbdAlmageed, W.; Narayanan, S. Adversarial defense for deep speaker recognition using hybrid adversarial training. In Proceedings of the IEEE-ICASSP, Toronto, ON, Canada, 6–11 June 2021; pp. 6164–6168.
95. Hussain, S.; Neekhara, P.; Dubnov, S.; McAuley, J.; Koushanfar, F. WaveGuard: Understanding and Mitigating Audio Adversarial Examples. In Proceedings of the 30th {USENIX} Security Symposium ({USENIX} Security 21), Virtual, 11–13 August 2021.
96. Wu, H.; Li, X.; Liu, A.T.; Wu, Z.; Meng, H.; Lee, H.Y. Adversarial defense for automatic speaker verification by cascaded self-supervised learning models. In Proceedings of the IEEE-ICASSP, Toronto, ON, Canada, 6–11 June 2021; pp. 6718–6722.
97. Liu, A.T.; Li, S.W.; Lee, H.y. Tera: Self-supervised learning of transformer encoder representation for speech. *arXiv* **2020**, arXiv:2007.06028.
98. Li, X.; Li, N.; Zhong, J.; Wu, X.; Liu, X.; Su, D.; Yu, D.; Meng, H. Investigating robustness of adversarial samples detection for automatic speaker verification. *arXiv* **2020**, arXiv:2006.06186.
99. Wang, X.; Sun, S.; Shan, C.; Hou, J.; Xie, L.; Li, S.; Lei, X. Adversarial Examples for Improving End-to-end Attention-based Small-footprint Keyword Spotting. In Proceedings of the IEEE-ICASSP, Brighton, UK, 12–17 May 2019; pp. 6366–6370.