IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2022, accepted 29 October 2022, date of publication 7 November 2022, date of current version 16 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3220639

RESEARCH ARTICLE

A Highly Stealthy Adaptive Decay Attack Against Speaker Recognition

XINYU ZHANG, YANG XU, SICONG ZHANG, AND XIAOJIAN LI

Key Laboratory of Information and Computing Science Guizhou Province, School of Cyber Science and Technology, Guizhou Normal University, Guiyang 550001, China

Corresponding author: Yang Xu (xy@gznu.edu.cn)

This work was supported in part by the Central Government Guides Local Science and Technology Development Special Funds under Grant [2018]4008; in part by the Science and Technology Planned Project of Guizhou Province, China, under Grant [2020]2Y013; and in part by the Guizhou Postgraduate Research Fund Project under Grant YJSKYJJ [2021]102.

ABSTRACT Speaker recognition based on deep learning is currently the most advanced and mainstream technology in the industry. Adversarial attacks, an emerging and powerful attack against neural network models, also posing serious security problems for speaker recognition. Common gradient-based attack methods such as FGSM (Fast Gradient Sign Method), PGD (Projected Gradient Descent), and MI-FGSM (Momentum Iteration-FGSM) generate adversarial examples that are poorly stealthy and easily perceived by the human ear. To improve the stealthiness of the adversarial examples, this paper proposes a new attack method called the Adaptive Decay Attack (ADA), whose stealth is very close to the CW_2(Carlini&Wagner) method based on optimization attacks, with much less computation time than CW_2. The method takes the set number of iterations as the termination condition, automatically adjusts the size of the maximum perturbation according to whether the attack is successful or not, and then uses the decay methods in learning rates such as exponential decay and cosine annealing to continuously reduce the step size. The experimental results show that under the two speaker recognition models x-vector, and i-vector, the proposed attack method improves the stealthiness metrics such as SNR and PESQ by at least 30% and 39%, respectively, compared with the best PGD attack under speaker identification of untargeted attacks. For the speaker identification task with targeted attacks, the average improvement is at least 20% and 25% compared to PGD. For the speaker verification task, the improvement is at least 29.5% and 33.4% compared to PGD. In addition, we also use this attack method for adversarial training to enhance the robustness of the model. Experimental results show that ADA-based adversarial training takes 28.31% less time than PGD-based adversarial training, and its improved robustness is generally superior to PGD-based adversarial training. Specifically, the attack success rate of PGD and ADA methods decreased from 50.88% to 36.47% and 64.74% to 45.82%, respectively.

INDEX TERMS Deep learning, adversarial attacks, speaker recognition, adaptive decay attack, adversarial training.

I. INTRODUCTION

A speech contains the identity of the speaker, text content, language information, etc. [1] Compared with other biometric recognition technologies, a speech is easy to collect, low cost, and the recognition process is contactless [2]. Speaker recognition, as a technique to recognize or identify a person from speech, is widely used in daily life and work, such as in controlling smart homes [3], financial transactions [4], personalized services for electronic products [5], forensic tests [6], etc.

Studies have shown that speaker recognition has been subject to malicious spoofing attacks, such as voice conversion [7] and speech synthesis [8], which have existed in the past, to a recent emerging type of attack called adversarial attacks. Speech conversion aims to change the source speaker's voice to that of the targeted speaker's tone while keeping the content of the voice unchanged. Speech synthesis aims at converting

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

any text into the corresponding speech. The main idea of adversarial attacks is to add a small artificial perturbation to a piece of original speaker utterances to form a new piece of audio that still sounds like the original speaker, at least to humans, and the model forces the identity of the new audio to be someone else.

The powerful capabilities of neural networks have led to their widespread use in various fields related to people's daily lives, yet recent studies have shown that neural networks are vulnerable to adversarial attacks. Adversarial attacks, also known as adversarial examples, have posed a significant security threat to the currently widely used neural network techniques since adversarial attacks were proposed. Regarding the reason for adversarial examples, Goodfellow et al [9] argued that the linear nature of deep neural networks in high-dimensional space leads to the creation of adversarial examples, which is believed that neural networks have high-dimensional and linear characteristics so that the initial perturbation values will be superimposed continuously when passed backward in the neural network, which is eventually sufficient to change the classification results of the model. With the continuous development and improvement of adversarial attacks, they have successfully deceived the current neural network-based designs for autonomous driving [10], face recognition [11], speech recognition [12], malicious code detection [13], and other related tasks. In recent years, adversarial attacks in speaker recognition scenarios have not been extensively studied, and it has become significant to understand the vulnerability of speaker recognition to adversarial attacks and how to increase its robustness.

Our contributions are as follows:

•In the task of attacking speaker recognition, we provide common gradient-based and optimization-based attack methods such as FGSM, PGD, MI-FGSM, and CW_2. And we propose a new attack method called the ADA, which can be applied to different speaker recognition models and all recognition tasks. The aim of the new method is to improve the stealthy of the generated adversarial examples from being easily perceived by humans. Experimental results show that this method improves the stealthy significantly compared to other gradient-based methods, and the stealthiness is very close to that of the optimization-based CW_2 method, and the computation time is much faster than that of CW_2.

•In addition, we consider the problem of how to improve the robustness of the model. We compare the proposed ADA-based method for adversarial training with the FGSM-based and PGD-based adversarial training methods for analysis. Experiments demonstrate that ADA-based adversarial training improves the robustness of the model overall better than the other two methods and requires less time for training.

The remainder of this paper is organized as follows: Section II covers a basic introduction to speaker recognition and adversarial attacks. Section III describes the research related to adversarial attacks in speaker recognition. In Section IV, we introduce some attack methods, reveal their shortcomings, propose a new attack method called the ADA, and then introduce the defense method of adversarial training. Section V contains the experimental setup and experimental environment, the models used, and the metrics measured. Section VI presents the results of the attack and defense. Finally, Section VII summarizes the overall contents of this paper and proposes future research directions.

II. BACKGROUND

A. BASICS OF SPEAKER RECOGNITION

Speaker recognition [14], also known as voice recognition, is a technology that distinguishes the voices of different speakers according to the identity of the speaker. Speaker recognition is fundamentally different from speech recognition technology. Speech recognition is a technology that converts speech signals into text content, and in most cases does not care whom the speaker is, hoping to filter out information related to the speaker's identity from the signal and retain only the textual content information. Speaker recognition technology, on the contrary, wants to filter out information related to the text content from the signal and retain only the speaker's identity information, robustly identifying the speaker's identity among the different speech segments.

A complete speaker recognition system is shown in Figure 1 below. Speaker recognition technologies are divided into two main categories according to the task and application scenario they are designed to recognize: speaker verification (SV) and speaker identification (SI). The question to be solved by speaker verification technology is: "Is this speech spoken by this particular person?" The recognition result is either accepted or rejected, so the voice verification technology can be seen as a 1-to-1, two-category problem. At the registration stage, the speaker verification technique first performs feature extraction based on all audio examples provided by a particular speaker, and further aggregates the audio features to generate a model with the ability to represent the identity of that speaker. In the recognition phase, unidentified audio data is provided, which is then compared with the model generated in the previous step, resulting in a matching score. We compare this match score with a predefined threshold to get the recognition result. If the match score is greater than the threshold, it is recognized as accepted by the model; conversely, it is recognized as rejected. The higher the score, the more likely it is that the new audio is spoken by the registrant.

The speaker identification technology needs to deal with the question: "Who spoke the passage?" This is limited to a particular speaker in a set containing N particular speakers, which can be seen as a many-to-one, multi-classification problem. Speaker identification can be subdivided into closed-set speaker identification (CSI) and open-set speaker identification (OSI). In closed-set speaker identification, The recognition result is that the person with the highest matching score in a set of N speakers; while in open-set speaker identification, due to the role of impostor (i.e., not in the set of speakers), our set size becomes N+1. And the recognition

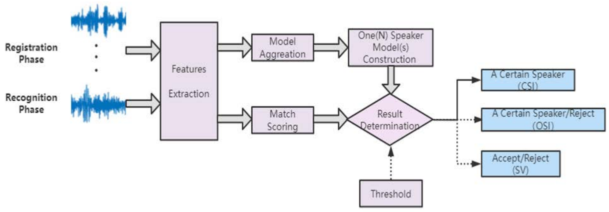


FIGURE 1. The framework of the speaker recognition system.

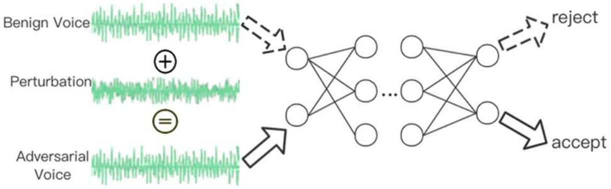


FIGURE 2. Adversarial attack on the speaker verification system.

result must satisfy two conditions, a): the highest score in the set; b): the score must be greater than the threshold. If the highest score is below the threshold, the audio is recognized as an impostor.

Speaker recognition system can be classified according to the recognition task, but also according to the content of the recognition, into three categories: text-dependent recognition, text-independent recognition, and text-prompted recognition. Text-dependent recognition, usually called ''fixed text'' speaker recognition, requires restricting the text content and duration of the speaker's speech. Text-independent recognition, on the contrary, can be recognized regardless of the content and duration of the speaker's speech. Text-prompted recognition randomly selects one text from a set containing multiple texts and then asks the speaker to say this text for speaker recognition. The experiments conducted in this paper all use text-independent recognition.

B. BASICS OF ADVERSARIAL ATTACKS

Adversarial attacks are an example generated by adding a small perturbation to the original benign data that is imperceptible to the human ear, which can effectively fool the target model into giving a wrong prediction output with high confidence. A satisfactory adversarial example often needs to satisfy two conditions: firstly, it must be able to force the model to classify errors and be imperceptible to humans after adding small perturbations, i.e., it has a high success rate of attack; secondly, the smaller the perturbations added, the better, i.e., it has high stealthiness.

Figure 2 illustrates an example of adversarial attack on the speaker verification task: adversarial audio formed by artificially adding subtle perturbation to the original audio that the human ear cannot imperceptible. The adversarial example causes the speaker verification model to give a different result from the original example and then switch from rejection to acceptance, but the human ear doesn't sound different from the two audios. If the attacker chooses to attack speaker identification task, the impostor can be recognized as one of the speakers in the registered set, or one of the speakers in the registered set can be recognized as another person in the set. From the above, it is seen that the presence of adversarial attacks may expose the speaker recognition system to serious security problems.

In the case of untargeted attacks, the attacker does not need to specify a specific attack category when generating the adversarial example, but only needs to make the target model misclassify the adversarial example; whereas targeted attacks not only require the target model to misclassify but also require the adversarial example generated by the attack algorithm to further fool the target model to identify as the specified target category, which is more complicated than untargeted attacks. The theoretical difference between the two is that the untargeted attack maximizes the loss function that is different from the original label of the example, and the targeted attack minimizes the loss function of the original label and the target label. The optimization equation for both is as follows:

$$\begin{cases} \min l(f(x+\delta), t) \\ \text{s.t. } \|\delta\| \le \varepsilon \end{cases} \tag{1}$$

$$\begin{cases} \min l(f(x+\delta), y) \\ \text{s.t. } \|\delta\| \le \varepsilon \end{cases} \tag{2}$$

where $f(\cdot)$ denotes the given model, x denotes the input example, δ denotes the added perturbation, y is the true label corresponding to the input example x, t is the label set by the attacker, and ε is the set maximum perturbation.

Also, adversarial attacks can be classified into two types of white-box [15] and black-box attacks [16], [17] according to whether they know the specific details of the model. In a white-box attack, the attacker knows all the information about the target model, such as the network structure and model parameters, and even the parameters and structure of the defense, to effectively design attack algorithms; while in a more sophisticated black-box attack, the attacker cannot get any information about the model and can only iteratively query the model and estimate the target model based on the results returned by the model. The commonly used black-box approach is to build an alternative model, aiming to train a model with similar decision bounds to the target model, perform a white-box attack on this model, and then migrate the generated adversarial examples to the target model. Compared with the black-box attack scheme, the white-box attack scheme has the advantage of being easy to implement. In this paper, we mainly consider the untargeted and targeted attacks under the white-box attack and study the black-box attack in the subsequent work.

The current adversarial examples generation algorithms for white-box attacks mainly include two types: 1) gradient-based attack methods; and 2) optimization-based methods. Gradient-based attack methods are mainly designed for maximizing the target loss, solving the gradient according to the loss value, and further adding adversarial perturbations in the gradient direction, thus effectively fooling the target model to generate false prediction outputs. This type of attack algorithms can often generate the adversarial examples quickly, but the perturbation of the adversarial examples is more obvious. Most of the attack algorithms belong to this type of attack method, mainly including FGSM [9], PGD [18], MI-FGSM [19], etc. The attack method proposed in this paper is also based on the gradient attack, and the added perturbation is guaranteed to be small and the stealthiness of adversarial examples to be high. The optimization-based attack method, on the other hand, views the adversarial example generation process as an optimization problem, and finally generates the adversarial examples by continuously optimizing the target loss, and the representative of this type of attack algorithm is the CW_2 [20] algorithm. The adversarial examples generated by this method tend to have smaller adversarial perturbations but at the cost of the very low attack efficiency of this algorithm.

With the continuous development of adversarial attacks, the defense methods of adversarial attacks have also received extensive attention and research. In the field of speech recognition, it is mainly from two aspects of eliminating adversarial perturbations and improving the robustness of models. In eliminating adversarial perturbations, the main reference is large to the methods in the image domain such as feature compression, JPEG compression, quantization, random smoothing, and other input transformation-based defense methods [21], [22], [23]. By combining the characteristics of audio (e.g., temporality, etc.) and input transformation methods to eliminate adversarial perturbations, it is not yet known whether this can be applied in the field of speaker recognition. In terms of improving the robustness of the speaker recognition model, the speaker recognition model based on deep learning is trained using a dataset with mixed adversarial examples and original examples by adversarial training [9] to improve the sensitivity of the speaker recognition model to the adversarial examples. In this paper, the adversarial training approach is mainly adopted for an active defense to improve the robustness of the model.

III. RELATED WORK

Jati et al. [24] used classical attack methods such as FGSM, PGD, etc. for attack models, it is demonstrated that the speaker recognition system is highly vulnerable to adversarial attacks, then a series of ablation experiments are conducted to find the best parameters for the attack methods, and finally, adversarial training is performed by combining different attack methods, and it is found that the adversarial training based on PGD is the best defense method, which effectively improves the robustness of the model. However, it lacks to consider the security issues under targeted attacks and open set identification scenarios.

Kreuk et al. [25] claimed that the vulnerability of the end-to-end DNN-based speaker verification system against FGSM attacks is first demonstrated. The authors also experiment with the speaker verification system against attacks in the cross-feature (MFCC and Mel-spectrum), and cross-dataset cases. In this paper, no defense method is proposed, and the attack method and recognition task scenario are single.

Li et al. [26] shown that the traditional speaker verification system based on the i-vector is vulnerable to adversarial attacks, and the adversarial examples generated with the FGSM attack method are migratory and can pose a threat to different recognitional models such as x-vector systems under cross-model and cross-feature conditions. However, the attack method and recognition task are single, and no defense method is proposed.

Chen et al. [27] performed a black-box targeted adversarial attack on speaker recognition systems for the first time and proposed a method based on the attack algorithm BIM and the gradient estimation algorithm NES to generate adversarial examples to attack these traditional speaker recognition models such as GMM-UBM and i-vector models, and achieve close to 100% attack success rate on both open source and commercial voice recognition systems (Tiancong Intelligence), and can effectively migrate to the Microsoft Azure voice recognition system, including API attacks and over-the-air physical attacks in real-world scenarios. However, attacks under DNN-based speaker recognition models are not considered.

Shamsabadi et al. [28] proposed a white-box steganography-based adversarial attack method that changes the previous approach from optimizing adversarial loss to using a Gated Convolutional Autoencoder (GCA) operating in the DCT domain by the inter-frame cosine similarity between the MFCC feature vectors extracted from the original audio file and the adversarial audio file degree to take human perception into account and is trained using a multi-objective loss function (perceptual loss + adversarial loss) to generate and hide the adversarial perturbations in the original audio file. This approach reduces the perceptibility of noise to some extent and has a high PESQ metric.

Wang et al. [29] Based on the psychoacoustic principle of frequency masking, use a masking threshold instead of a parametric number to limit the size of perturbations to generate perturbations inaudible to the human ear and perform a targeted white-box attack on the speaker recognition system x-vector, specifying any speaker target, with a success rate of 98.5%. In addition, this attack method is also applied to non-speech data such as music to perform the attack.

Wang et al. [30] used two types of attacks, FGSM and LDS (local distributional smoothness), to generate adversarial examples to attack the end-to-end speaker verification model, respectively, and experimentally demonstrate the vulnerability of the speaker verification model to adversarial attacks, and then combine these two types of adversarial examples for model regularization to improve model robustness.

IV. PROPOSED METHOD

A. ATTACK METHOD

In general, gradient-based untargeted attacks generate adversarial examples mainly by solving the optimization problem for the following equation:

$$\begin{cases} argmax_{x'} L\left(x', y\right) \\ \text{s.}t \|x' - x\|_p \leq \varepsilon \end{cases} \tag{3}$$

Maximize the loss function L of the label corresponding to the adversarial example with the true label y in the limit of the maximum perturbation ϵ and p-parametrization.

FGSM: FGSM is a fast gradient-based untargeted attack method, only one iteration to complete the attack, belongs to the single-step attack, in the generation time is the shortest, yet the success rate of the attack is very limited. The method maximizes the loss concerning the original target label by adding perturbations to the original example in the lp parameter limit and performing updates along the gradient direction of the loss function. In this paper, the experiments are mainly conducted under the $l\infty$ paradigm. Its formula for generating adversarial examples is as follows:

$$x' = x + \epsilon \cdot sign(\nabla_x L(x, y)) \tag{4}$$

where ϵ is the maximum perturbation allowed to be added (hyperparameter), also the step size of the optimization, $\nabla_{\mathbf{x}} L(\mathbf{x}, y)$ is the partial derivative of the loss function, in the CSI task the cross-entropy loss function is used, while in the OSI, SV task the margin loss is used due to the problem of judging the threshold.

Under targeted attacks, it is required to minimize the loss with a designated target, t is a designated target. Its formula for generating the adversarial example is as follows:

$$x' = x - \epsilon \cdot sign(\nabla_x L(x, t)) \tag{5}$$

PGD: To solve the linearity assumption problem in FGSM, PGD is proposed to solve the internal maximum problem. PGD is an improved version of FGSM by dividing the perturbation size of one iteration of FGSM into a small fraction of each iteration and then projecting the updated adversarial example perturbation to a prescribed range, replacing the overflow with a boundary value. Compared with FGSM, PGD can find noise points more precisely and effectively and belongs to a multi-step attack, which consumes much more computational resources and time than a single-step attack, and its worst effect of generating adversarial examples is also comparable to FGSM. The adversarial example generation algorithm for the projected gradient descent method is shown in the following equation:

$$\begin{cases} x'_k = Clip\{x'_{k-1} + \alpha \cdot sign(\nabla_x L(x'_{k-1}, y))\} \\ \text{s.}t.x'_0 = x \end{cases} \tag{6}$$

where $Clip\{*\}$ is used to crop the overflow value to ensure that the adversarial example is within the domain of the original example, α is the perturbation value that increases with each iteration.

MI-FGSM: Also known as **MIM**. A method based on momentum iterative gradient, which memorizes the gradient of the loss function for each iteration based on BIM, i.e., when performing iterations, the perturbation in each round is not only related to the current gradient, but also to the previously calculated gradient, which can stabilize the update direction and avoid local maximum.

$$g_k = \mu \cdot g_{k-1} + \frac{\nabla_x L(x'_{k-1}, y)}{\| \nabla_x L(x'_{k-1}, y) \|_1} \tag{7}$$

$$x'_k = x'_{k-1} + \alpha \cdot \text{sign}(g_k) \tag{8}$$

g_k indicates that the gradient of the previous k iterations is stored, and μ is defined as the decay factor (hyperparameter).

CW$_2$: Unlike the other above methods, CW$_2$ uses L2 norm to the optimization of Equation 3 to measure the difference between the adversarial example x' and the original example x. Furthermore, the problem of optimizing δ is transformed by introducing a new variable ω to optimize ω:

$$\delta_i = 1/2(\tanh(\omega_i) + 1) - x_i \tag{9}$$

$$minimize \| 1/2(tanh(\omega) + 1) - x \|_2^2 + c{\cdot}f(1/2(tanh(\omega){+}1)) \tag{10}$$

This turns the optimization problem into an unconstrained minimization problem. By mapping to tanh space, the adversarial examples can transform on $(-\infty, +\infty)$, which is beneficial for optimization. The following equation is generally used for the loss function:

$$L(x', t) = max\{max[Z(x') : i \neq t]_i - [Z(x')]_t, -k\} \tag{11}$$

$Z(*)$ is the output of the logit layer. k is the preset confidence, the larger k is, the higher the confidence of the generated adversarial examples.

ADA: For CW$_2$, although it is an attack method based on finding the minimum perturbation, the generation efficiency is extremely low, and the practicality is not high. There is no doubt that PGD and MIM perform very powerfully in terms of attack performance, yet the adversarial examples generated using their attack ideas are not guaranteed to generate small enough perturbations to create adversarial examples that are easily perceived by humans to some extent. For PGD, the attack success rate is closely related to the hyperparameter maximum perturbation value ε and the step size α. If the maximum perturbation value ε is set too large, the attack success rate will always satisfy the attack demand with the number of iterations, but the added perturbation is not the most satisfying; Conversely, if the maximum perturbation value ε is set too small, the increased perturbation, on the one hand, will also be very small, which may make the generated examples not adversarial. On the other hand, an unreasonable step size α may cause the gradient optimization process to fail to converge and oscillate back and forth between the local optimum or the global optimum. For MIM, to ensure the success rate of the attack, the information of previous gradients is additionally added to each gradient update for calculation, which makes the addition of a larger perturbation, and the larger the hyperparameter μ is set, the larger the perturbation is.

Given the shortcomings of these two attack methods, the ADA is proposed to find the minimum perturbation that satisfies the success of the attack, and the complete attack steps are shown in Algorithm 1. First, input the original benign example x, initialize the step size α, and indeed the attack type as an untargeted attack or targeted attack. The optimal perturbation is performed instead of setting a fixed size maximum perturbation value like FGSM, PGD, or MIM. Specifically, the norm is constrained by projecting the adversarial perturbation δ within the maximum perturbation range around the original audio x. The perturbation size is then modified based on the results of the two-category of judgments. And if the example after adding the perturbation in the $(k-1)_{th}$ iteration if it is not adversarial, expand the range of the maximum perturbation value in the next iteration to $(1+\lambda)\varepsilon_{k-1}$; Conversely, after the adversarial example is adversarial, the range of the maximum perturbation value is narrowed down to $(1-\lambda)\varepsilon_{k-1}$ in the next round of iterations. After each two-category of judgment, the value of step size α is reduced, and the means of reduction are exponential decay and cosine annealing in the learning rate decay method. As many iterations pass, the maximum perturbation value and step size become smaller and smaller, and finally, an adversarial example that is both adversarial and satisfies the added perturbation is small enough is returned. We refer to the exponential decay function to reduce the size of α as ADA-E and the cosine annealing function to reduce the size of α as ADA-C. The following algorithm is an example of the exponential decay function.

Algorithm 1 ADA-E

Input: controlling with/without targeted attack m, number of iterations K, gradient information $grad$, benign example x, label (untargeted) or preset label (targeted) y, loss function L_{cross}, model$f(*)$, step size α, sign function $sign(*)$, clipping function $clip(*)$, perturbation size ε, adjusting the range of perturbations λ, Exponential decay function $ExponentialLR(*)$

Output: adversarial example x'

```
Initialize
if targeted attack $m \leftarrow 1$ else $m \leftarrow -1$ end if
for$k \leftarrow 1$ in $K$ do
    $grad \leftarrow m\nabla_{x_{k-1}}(L_{cross}(f(x_{k-1}), y))$
    $x_k \leftarrow x_{k-1} - \alpha_{k-1} \cdot sign(grad)$
    $x_k \leftarrow clip\,(x_k, x_k - \varepsilon_k, x_k + \varepsilon_k)$
    $x_k \leftarrow clip\,(x_k, -1, 1)$
    if $x_{k-1}$ is $y$ then
    $\varepsilon_k \leftarrow (1-\lambda)\,\varepsilon_{k-1}$
    else
    $\varepsilon_k \leftarrow (1+\lambda)\varepsilon_{k-1}$
    end if
    $\alpha_k \leftarrow ExponentialLR(\alpha_{k-1})$
end for
return $x'$
```

B. ADVERSARIAL TRAINING

As a typical active defense method, the idea of adversarial training is very straightforward. The generated adversarial examples are added to the training process, so that the model learns the adversarial example data in advance, which can be understood as a min-max optimization problem:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y)] \tag{12}$$

where θ is the weight parameter of the model, δ is the size of the perturbation, $\mathcal{S}$ is the range of the perturbation, and $\mathcal{D}$ is the data distribution.

The inner layer is a maximization that aims to find the perturbation that maximizes the loss function, which simply means that the added perturbation should try to cheat the neural network. The outer layer is a minimization formula that optimizes the neural network, i.e., when the perturbation is fixed, we train the neural network model to minimize the loss of the training data, i.e., to make the model robust to the perturbation. Adversarial training is more time-consuming than normal training, and the resulting model will be less accurate for benign examples, yet it is still a powerful tool to defend against adversarial attacks.

Taking ADA-based adversarial training as an example, the adversarial training objective function can be expressed as:

$$\begin{aligned} L(\theta, x, y) = {} & cL(\theta, x, y) \\ & + (1-c)L(\theta, x+\alpha\cdot\text{sign}(\nabla_x L(\theta, x, y)), y) \end{aligned} \tag{13}$$

where $x + \alpha \cdot \text{sign}(\nabla_x L(\theta, x, y))$ is the adversarial example generated by the benign example x iteratively according to the ADA method; c is used to balance the accuracy of the benign and adversarial examples, i.e., the ratio taken by the adversarial and benign examples.

V. EXPERIMENTAL SETUP

A. DATASETS AND EXPERIMENTAL ENVIRONMENT

Like [24] and [31], the datasets are taken from Librispeech [32], the speech database Librispeech, which contains 1000 hours of 16 kHz recordings, cut and organized into text-annotated audio files of about 10 seconds each. We provide a total of 5 datasets, the first 3 datasets for the 3 types of identification tasks, which are taken from "dev-other" and "train-other-500" in Librispeech named as enroll$_{10}$, test$_{10}$, and imposter$_{10}$. enroll$_{10}$ has 10 people (5 men and 5 women), and each person takes 10 random speech data for speaker registration; test$_{10}$ also has 10 people, but the 10 people taken must be the same as enroll$_{10}$, and each person takes 100 random speech data (no conflict with enroll$_{10}$) for testing; imposter$_{10}$ denotes the impostor dataset mainly used for OSI, SV tasks, where all 10 speakers in the dataset are different from enroll$_{10}$, and each speaker is randomly taken 100 voices. The latter two datasets are used for adversarial training. The datasets are taken from "train-clean-100" named train$_{251}$, test$_{251}$, both of which contain 251 individuals (126 men and 125 women). The train$_{251}$ is used for training and contains 25652 speech data, and test$_{251}$ is used for testing and contains 2887 speech data.

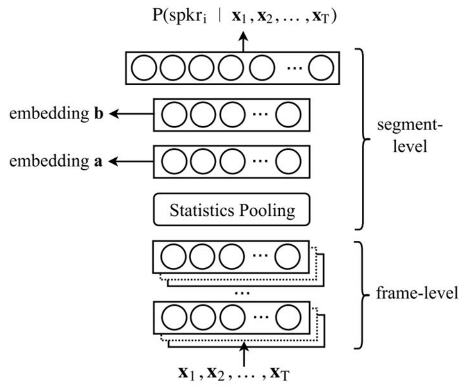

P(spkr$_i$ | $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$)
embedding **b**
embedding **a**
Statistics Pooling
segment-level
frame-level
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$

FIGURE 3. x-vector network architecture.

This experiment was implemented on an Ubuntu 20.04 system with an Intel i7-11700KF at 3.6GHz CPU, an NVIDIA GeForce RTX 3070Ti with 8GB of video memory, and 32GB of RAM.

B. MODEL INTRODUCTION

We will use the two models i-vector [33], and x-vector [34] to implement the attack on the three types of recognition tasks, for the AudioNet [35] model is more biased toward doing adversarial training.

The x-vector system is a speaker recognition system based on DNN, which is the mainstream baseline model framework in the current speaker recognition field. The DNN is trained to extract the vocal features of the speaker, and the extracted speaker embedding is called the x-vector. The whole system can be divided into two modules, and the complete architecture is shown in Figure 3 [36] below: the x-vector system contains five frame-level TDNN layers, one statistical pooling layer, two sentence-level fully connected layers, and one SoftMax layer.

After the speaker model is trained, the back-end will use the extracted speaker features x-vector to train a PLDA [37] model for channel compensation to reduce the impact of channel noise on the system and use the model for similarity scoring.

Before the rise of deep learning-based speaker recognition, i-vector, which belongs to the traditional speaker recognition models, have been the most popular. I-vector is a simplified version of joint factor analysis based on JFA [38], that is, a Total factor matrix (T) is used to describe both speaker information and channel information, and then the speech is mapped to a fixed and low-dimensional vector. The existence of channel information in matrix T will interfere with the recognition system and even seriously affect the recognition accuracy of the system. Therefore, channel compensation for i-vector is required, so WCCN [39], Linear Discriminant Analysis LDA [40] and Probabilistic Linear Discriminant Analysis (PLDA) are usually used. The framework of i-vector system is shown in Figure 4 below:

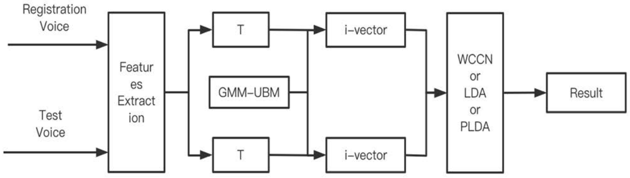


FIGURE 4. i-vector system framework.

TABLE 1. Structure of AudioNet network with DSP added.

Input: audio waveform $\boldsymbol{x} \in [-1,1]^T$
DSP: $[-1,1]^T \rightarrow \mathrm{R}^{32\times T'}$ Conv1D(32,64, k = 3) BatchNorm, ReLU; MaxPool1D(2) Conv1D(64,128, k = 3); BatchNorm, ReLU Conv1D(128,128, k = 3); BatchNorm, ReLU Conv1D(128,128, k = 3); BatchNorm, ReLU; MaxPool1D(2) Conv1D(128,128, k = 3); BatchNorm, ReLU Conv1D(128,64, k = 3); BatchNorm, ReLU, MaxPool1D(2) Conv1D(64,32, k = 3), BatchNorm, ReLU MaxPool1Dovertime(the output is speaker embedding) FullyConnected(32,251)

AudioNet is a one-dimensional convolutional neural network model with a digital signal processing (DSP) front-end [24] added to the original model for extracting the log-Mel spectrogram from the time-domain waveform of the audio as an input to the convolutional layer. The neural network consists of 8 convolutional layers and is mainly used to transform the spectrogram into a single 32-dimensional vector of speaker embedding. BatchNorm and ReLU operations are performed for all CNN layers, and only MaxPooling1D is added at the end of the CNN layers in layers 1, 4, and 6. The final fully connected layer maps the speaker embedding into the class logits. The complete network architecture is shown in Table 1.

C. METRICS

In this paper, we will evaluate the attack effect of each generation algorithm on speaker recognition models using attack success rate (ASR), signal-to-noise ratio (SNR), perceptual evaluation of speech quality (PESQ), and time for generating adversarial examples.

The attack success rate is used to indicate the percentage of generated adversarial examples that are misclassified by the model, and the untargeted attack is defined as:

$$ASR = \frac{Num(f(x') \neq y)}{Num(f(x) = y)} \tag{14}$$

$Num(*)$ represents the number, and if it is a targeted attack, the numerator is changed to $Num(f(x') = t)$.

For measuring the perceptibility of speech adversarial examples, we use speech quality evaluation methods such as signal-to-noise ratio (SNR) and speech quality perception assessment (PESQ). The signal-to-noise ratio is the ratio of the power of the signal to the power of noise, and the unit of measurement is dB. The main measure of distortion in the experiments is the size of the added perturbation relative to the original audio, and then the difference between the adversarial audio generated by the various generation algorithms is compared, which is calculated as follows:

$$SNR = 10lg(\frac{Ps}{Pn}) \tag{15}$$

Ps represents the power of benign examples and Pn represents the power of perturbations. The larger the value of the signal-to-noise ratio, the better.

The calculation of PESQ is more complicated, mainly by extracting the difference between the input two signals in the time-frequency domain or transform domain feature parameters and then mapping the feature parameter differences by a neural network model to obtain an objective sound quality score. The PESQ score ranges from 0 to 5. Higher scores indicate better voice quality.

The time to generate the adversarial examples is mainly used to accurately compare the generation speed of various attack algorithms in seconds.

VI. EXPERIMENTAL RESULTS

A. ALGORITHM PARAMETER SETTING

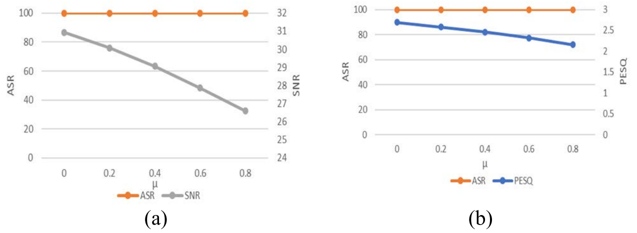
The step size of FGSM is $\varepsilon = 0.002$[24,31]. We also set the maximum perturbation $\varepsilon = 0.002$, number of iterations $K = \{10,20,30\}$ for PGD, MIM, ADA-E, and ADA-C. The step size $\alpha = 0.0004$ for PGD and MIM, similarly the initial step size $\alpha = 0.0004$ in ADA. For CW_2 we use 9 binary search steps to minimize adversarial perturbations, run 60-600 iterations to converge, and vary the confidence k from 0, 5, 10. In the experimental results, PGD-T, MIM-T, ADA-E-T, and ADA-C-T are used to represent the number of iterations of PGD, MIM, ADA-E, and ADA-C, e.g., PGD-10 means 10 iterations of PGD. CW_2-k denotes CW_2 when the confidence is set to k.

The first thing we do is to perform a series of ablation experiments in the x-vector model untargeted closed-set speaker recognition with MIM and ADA-E as examples to find the best parameters for the attack.

In the MIM experiment, its hyperparameter $\mu = \{0,0.2,0.4,0.6,0.8\}$, the $\lambda = 0.2$ that modifies the range of perturbation size in the ADA-E experiment, and the decay factor in the exponential decay that is the bottom $\gamma = \{0.75,0.8,0.85,0.9,0.95\}$. After experiments, it is proved that the success rate of the attack is always kept at 100% when the hyperparameter μ takes any value, so we will choose the value of μ when the perturbation is the smallest, i.e., the maximum value of SNR and PESQ, so we have the most suitable u = 0 in this scenario. Similarly, it can be obtained that the decay factor γ of ADA-E is most suitable to take 0.85 under the premise of ensuring a high success rate. Figures 5 and 6 below show the graphs of the tuning results for the two attack methods.

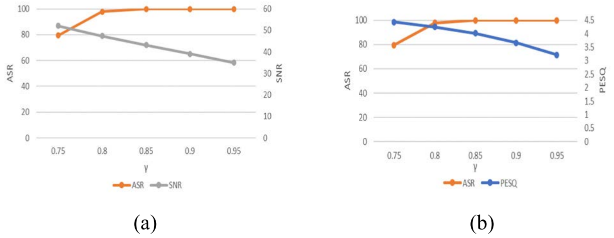
B. SPEAKER IDENTIFICATION FOR UNTARGETED ATTACK

Table 2 and Table 4 show the untargeted attacks under closed-set identification and open-set identification, respectively. In terms of attack success rate, no matter which identification task or which identification model, or which test dataset, the FGSM attack is the weakest attack among all attack methods, for example, the success rate is only 32.37% in x-vector for closed set recognition, while all other methods can achieve 100% attack success rate to deceive the model because FGSM is a single-step attack and does not need to perform iterations, but its speed of generating adversarial examples is far from that of other methods.



(a) (b)

FIGURE 5. MIM tuning results.

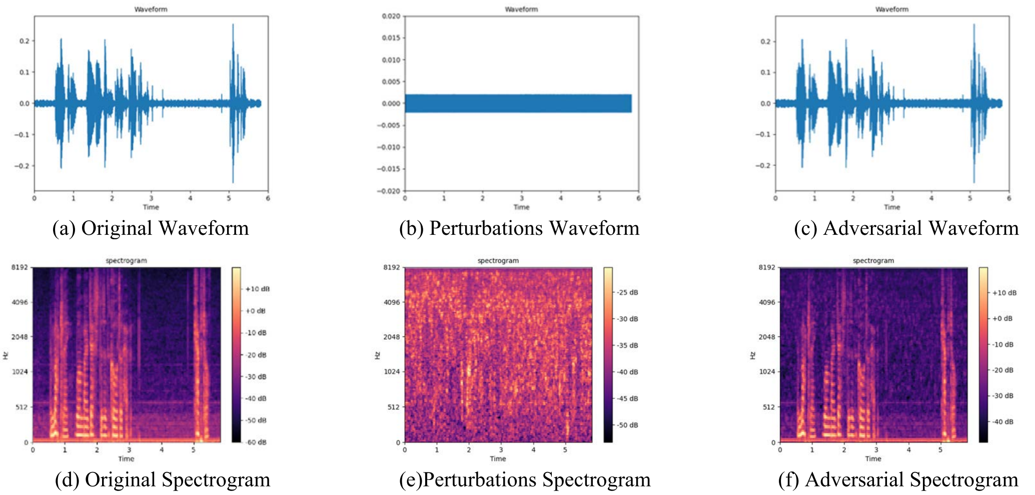


(a) (b)

FIGURE 6. ADA-E tuning results.

Other gradient-based attack methods stop finding adversarial examples based on the number of iterations and are close in generation time. In terms of the audio quality of the generated adversarial examples, PGD is the stealthiest in generating adversarial examples among the three compared methods FGSM, PGD, and MIM. Specifically, the SNR and PESQ values take the maximum value in PGD-10, yet the maximum SNR does not exceed 35 dB and PESQ score does not exceed 3 in both recognition models, and the perturbation increases with the increase of the number of iterations, and the SNR and PESQ values gradually decrease in PGD and MIM. For CW_2, the generated adversarial example has the highest SNR metric among all experiments, thanks to its optimization-based attack method, with the attendant problem that it consumes the most computational time of all methods. As the confidence k is set larger, the higher the success rate of CW_2 and the lower the stealthiness.

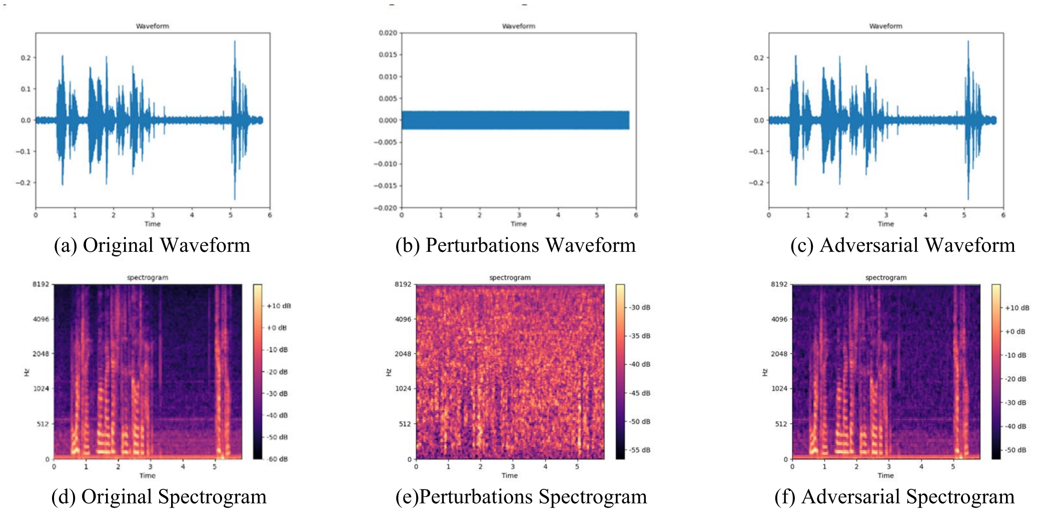
The ADA method proposed in this paper guarantees a high attack success rate and generation time very close to other methods, the lowest SNR and PESQ indexes in the adversarial examples generated by ADA-E and ADA-C methods are 42db and 4, which are 30% and 39% higher than those of PGD-10 with the best comparative experimental results, and the improved effect will continue to be enhanced with the increase in the number of iterations, the smaller the generated adversarial example perturbation will be, the more it can escape the detection of the human ear, but the computation time will also increase. Compared to CW_2, ADA-C-30 has



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 7. **The speaker with id 127: Waveforms and spectrogram of FGSM generated adversarial audio.**



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

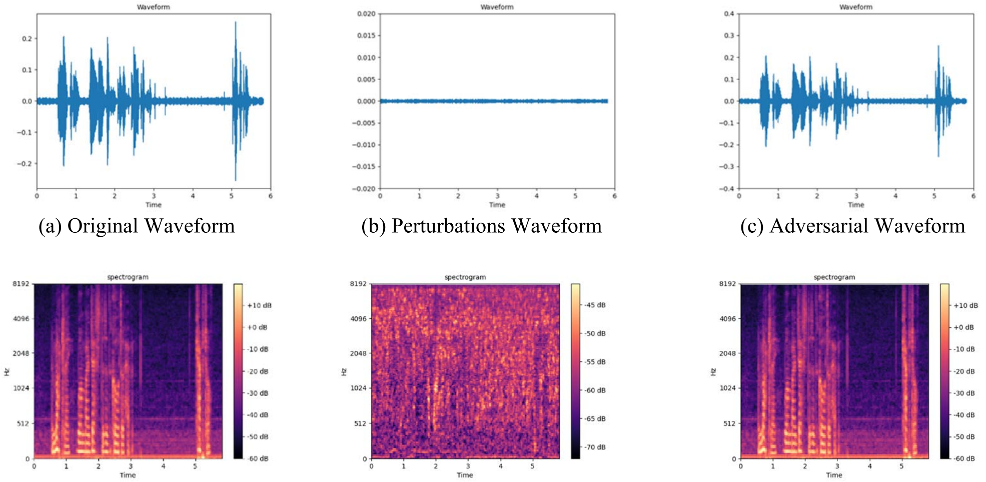
(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 8. **The speaker with id 127: Waveforms and spectrogram of PGD generated adversarial audio.**

higher PESQ than the former on datasets test$_{10}$, with a greater advantage in time consumption, reducing the time by at least 94%.

C. SPEAKER IDENTIFICATION FOR TARGETED ATTACK

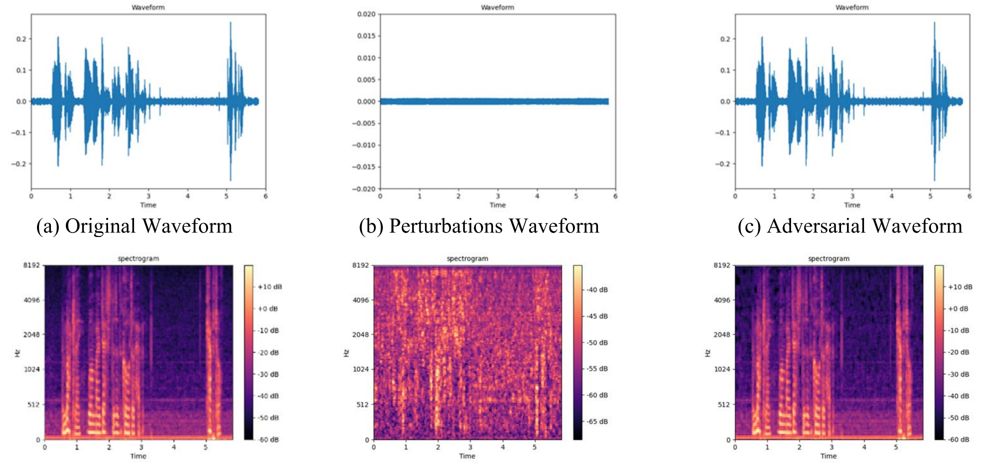
Table 3, Table 5, and Table 6 show the targeted attacks under closed-set identification and open-set identification, respectively. The targeted attacks are selected according to the set difficulty level. Simple indicates that the label of the most likely class other than the actual label of the normal example is used as the targeted class label; Hard indicates that the label of the least likely class other than the actual label of the normal example is used as the targeted class label. Under Simple difficulty, the success rate of FGSM attack in closed-set identification is still the lowest among all attack methods, but it is higher than that of untargeted attack under the same condition, and the success rates of targeted attack under x-vector of PGD-10 and MIM-10 are 99.88 and 99.89% respectively lower than that of untargeted attack under the same condition, and the success rate of attack can still reach 100% as the number of iterations increases. The success rate of the attack can still reach 100% with an increasing number of iterations, while the ADA can maintain a 100% success rate. The adversarial examples generated by all attack methods started equal or slightly improved in SNR, and PESQ metrics compared to the untargeted attacks under the same conditions. For open-set identification, the adversarial examples generated by the imposter$_{10}$ dataset are more confusing to deceive both models than test$_{10}$ for both untargeted and targeted attacks, and the PESQ metric of the adversarial examples generated by the imposter$_{10}$ dataset is greater than that of the adversarial examples generated by test$_{10}$ in terms of the stealthiness metric, while the opposite occurs for the SNR metric.



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 9. The speaker with id 127: Waveforms and spectrogram of CW_2 generated adversarial audio.



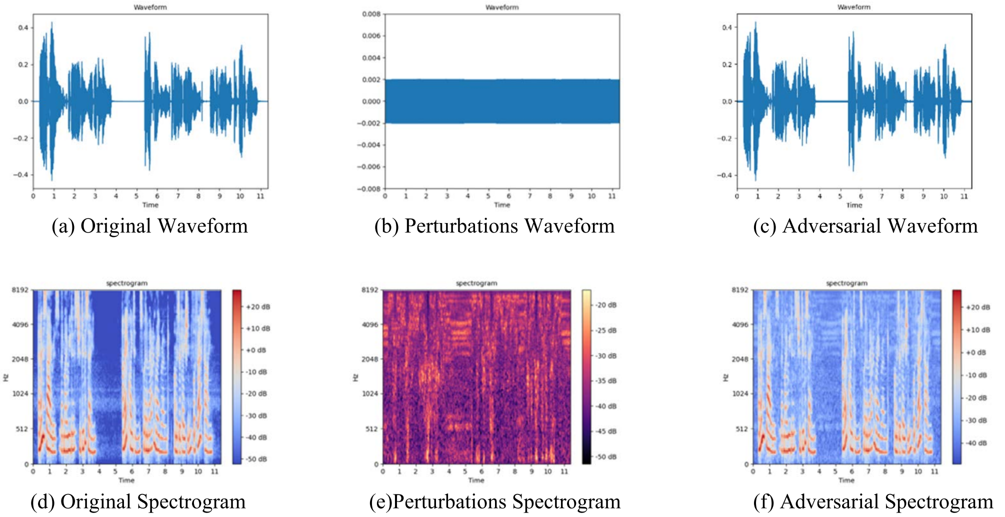
(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 10. The speaker with id 127: Waveforms and spectrogram of ADA-E generated adversarial audio.

The attack success rate of the FGSM method appears extremely low under Hard difficulty, e.g., only 0.99% in the closed set identification of the $test_{10}$ test set under the x-vector model, and other comparison methods including the ADA cannot achieve 100% attack success rate at 10 iterations, yet the success rate of the ADA is higher than all comparison methods. This is because the ADA sacrifices the stealthiness of audio adversarial examples in exchange for an increase in success rate. And the improvement in SNR and PESQ is not as great as in the untargeted attack or Simple difficulty, but it is still the attack method with the highest stealthiness, which can be easily observed in the table.

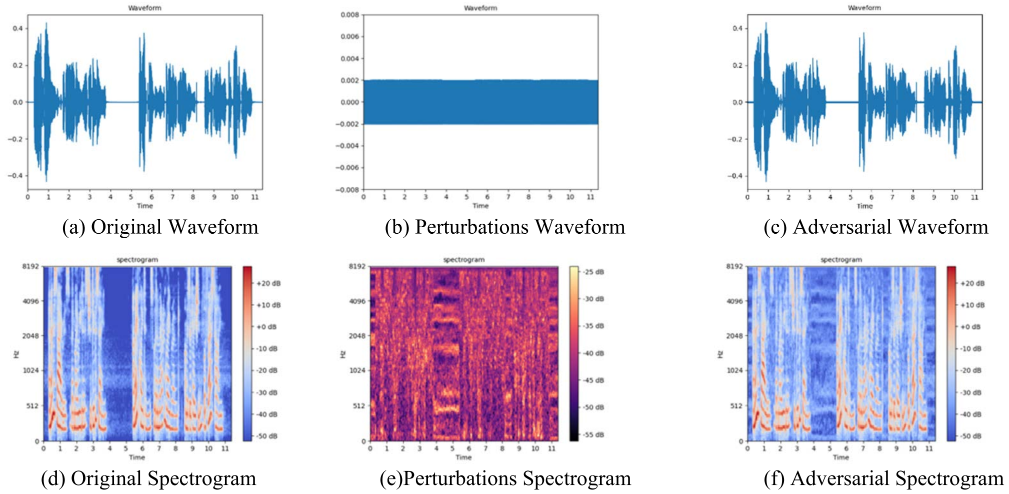
Combining the experimental results of Simple and Hard, without considering the CW_2 success rate, the stealthiness is slightly higher than the ADA method on the dataset $imposter_{10}$ and very close to the ADA method on dataset $test_{10}$. The computation time of CW_2 is also the most and will also increase with the difficulty of the attack. The lowest SNR and PESQ values of ADA attack are 43db and 4.01 respectively under the Simple difficulty of targeted attack; the lowest SNR and PESQ indexes of ADA attack are 35db and 3.14 respectively under the Hard difficulty of targeted attack. The stealthiness of the ADA method under targeted attack is the highest among all methods, and the change in the number of iterations has a significant positive correlation with the change in SNR and PESQ metrics. The SNR and PESQ can be improved by 20% and 25.4%, respectively, on average for the ADA under the targeted attack compared to PGD-10. In general, the targeted attack is more difficult compared to the untargeted attack.



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 11. **The speaker with id 6227: Waveforms and spectrogram of FGSM generated adversarial audio.**



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

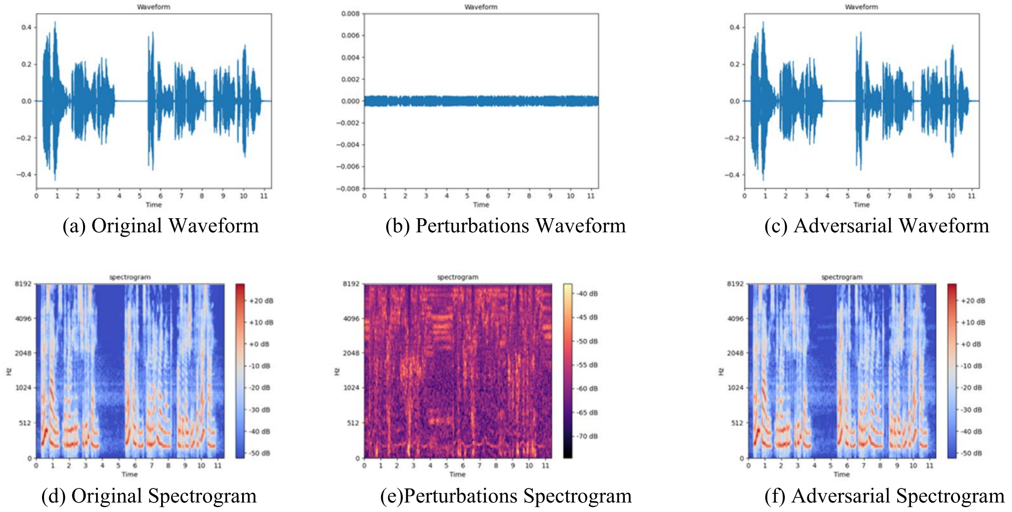
FIGURE 12. **The speaker with id 6227: Waveforms and spectrogram of PGD generated adversarial audio.**

D. ATTACK FOR SPEAKER VERIFICATION

In the speaker verification experiments, we specifically and mainly attack those 10 speaker verification models in the registered enroll$_{10}$ dataset separately as imposters (i.e., the imposter$_{10}$ dataset), which is more realistic, and then calculate the average of each attack result to obtain Table 7. Observation of Table 7 reveals that the ADA method improves SNR and PESQ by at least 29.5% and 33.4% on average compared to PGD-10 specifically, which is like the improvement in untargeted speaker identification, proving that this attack method is general and can be applied in all speaker recognition tasks. To understand the adversarial examples under the speaker recognition domain more intuitively, we take the speaker verification task as an example, from the following Figure 7-14 show the waveform and spectrogram after taking the original benign examples of two different speakers at random using various attack methods to generate the adversarial examples. After the comparison of Figures 7-10 and 11-14, the human eye can intuitively find that, through the comparison of waveform and spectrum, the size of perturbation increased by the attack method proposed in this paper is much smaller than other gradient attack methods but slightly larger than CW$_2$. This phenomenon is reasonable, CW$_2$ to find the minimum perturbation of the sample at the cost of huge computation time, but according to the experimental results ADA-C-30 and CW$_2$ generated examples of PESQ, SNR values are very close. Among these the perturbations added by FGSM and PGD are not easily distinguishable in the waveform, yet the perturbations added by PGD are also superior to the FGSM method from the comparison of the spectrogram.

E. ANALYSIS OF DIFFERENT MODELS

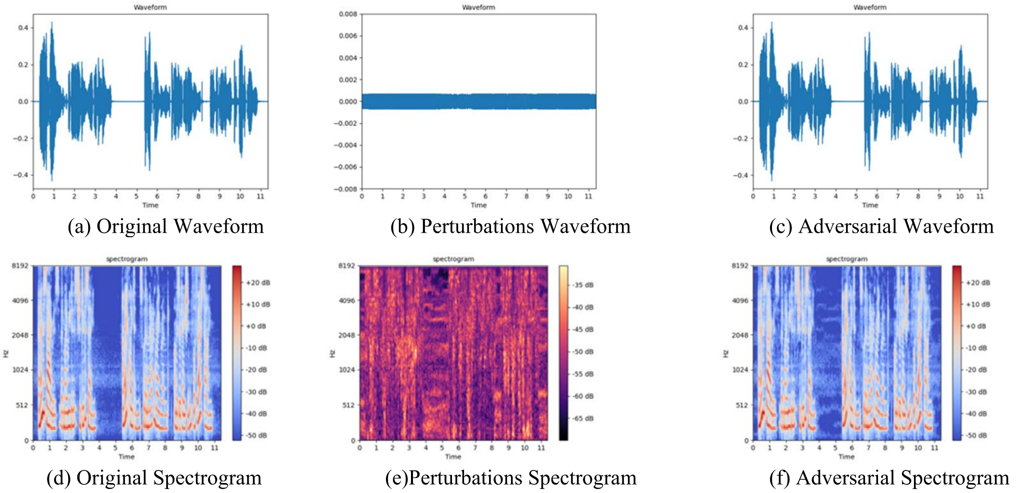
It is observed from Tables 2-7 that either the neural network-based x-vector system or the GMM-UBM-based i-vector system is vulnerable to adversarial attack spoofing and cannot



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 13. The speaker with id 6227: Waveforms and spectrogram of CW_2 generated adversarial audio.



(a) Original Waveform (b) Perturbations Waveform (c) Adversarial Waveform

(d) Original Spectrogram (e)Perturbations Spectrogram (f) Adversarial Spectrogram

FIGURE 14. The speaker with id 6227: Waveforms and spectrogram of ADA-E generated adversarial audio.

resist the adversarial attack. Among them, i-vector systems are more threatened by adversarial attacks than x-vector systems, e.g., the success rate of FGSM attacks under i-vector systems is higher than x-vector systems in all speaker recognition tasks, etc. In terms of stealthiness, the SNR metrics and PESQ metrics of the adversarial examples generated by the two systems are not significantly different. The SNR metrics of the adversarial examples generated on the i-vector system are equal to or higher than those generated on the x-vector system, and the size of the PESQ metrics has advantages and disadvantages for each of the two systems under different recognition tasks. In terms of generation time, it is more difficult and takes more time to generate adversarial examples on the i-vector system.

F. ANALYSIS OF ADVERSARIAL TRAINING

Table 8 presents the robustness of the trained model by attacking it with different attack methods after we trained the model separately in a specific way of adversarial training to test the robustness of the model. Among them, we selected three adversarial training methods, using FGSM-based adversarial training, PGD-10-based adversarial training, and ADA-C-10-based adversarial training, denoted as FGSM AT, PGD-10 AT, and ADA-C-10 AT, and the number of training epochs set was 150, with 50% of the adversarial examples and 50% of the benign examples in the adversarial training, and the maximum perturbation of all methods in the adversarial training $\varepsilon = 0.002$.

The experimental results show that the neural network AudioNet model without adversarial training is extremely vulnerable to adversarial attacks, even the worst attack FGSM has an 82.61% success rate, and the other three attack methods can achieve a 100% attack success rate. By comparing the adversarial training based on the three different methods, first, we can find that the models after adversarial training not only have a slight decrease in accuracy for

TABLE 2. Experimental results of various attack algorithms under untargeted closed-set speaker identification with test 10 dataset.

Attack	x-vector				i-vector			
Method	ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
FGSM	32.37	28.49	2.22	**100.3**	54.1	28.49	2.21	**420.1**
PGD-10	100	32.66	2.86	469.8	100	32.71	2.83	1929.4
PGD-20	100	31.73	2.75	886.7	100	31.82	2.74	3587.6
PGD-30	100	31.42	2.72	1291.1	100	31.53	2.70	5135.2
MIM-10	100	30.94	2.69	471.2	100	31.05	2.65	1940.7
MIM-20	100	28.01	2.37	898.1	100	28.10	2.33	3601.4
MIM-30	100	26.40	2.20	1295.2	100	26.46	2.16	5155.3
ADA-E-10	100	43.20	4.02	470.1	100	43.44	4.02	1930.3
ADA-E-20	100	44.50	4.12	887.3	100	44.77	4.12	3588.2
ADA-E-30	100	45.43	4.19	1293.2	100	45.67	4.19	5135.9
ADA-C-10	100	42.82	4.00	470.2	100	42.97	4.00	1929.3
ADA-C-20	100	44.85	4.14	888.7	100	45.10	4.14	3599.1
ADA-C-30	100	45.40	**4.21**	1292.2	100	45.67	**4.21**	5133.2
CW$_2$-0	97.89	**45.66**	3.94	23275.3	98.44	**47.30**	4.03	97000.2
CW$_2$-5	100	43.98	3.83	23277.1	100	45.42	3.85	97010.3
CW$_2$-10	100	42.69	3.72	23278.2	100	43.20	3.74	97014.5

TABLE 3. Experimental results of various attack algorithms under test 10 dataset with targeted closed-set speaker identification.

Attack	Attack	x-vector				i-vector			
Level	Method	ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
Simple	FGSM	47.01	28.49	2.22	**99.9**	61.31	28.49	2.21	**421.5**
	PGD-10	99.88	32.97	2.86	470.3	100	33.08	2.84	1930.4
	PGD-20	100	31.92	2.76	887.6	100	32.07	2.74	3589.9
	PGD-30	100	31.60	2.73	1292.2	100	31.73	2.71	5136.3
	MIM-10	99.89	31.57	2.72	472.5	100	31.76	2.69	1940.9
	MIM-20	100	28.47	2.38	899.8	100	28.66	2.36	3602.1
	MIM-30	100	26.79	2.21	1292.6	100	26.92	2.19	5157.1
	ADA-E-10	100	44.03	4.04	470.5	100	44.46	4.04	1929.5
	ADA-E-20	100	45.38	4.15	886.2	100	45.99	4.15	3586.3
	ADA-E-30	100	**46.21**	4.21	1292.6	100	46.85	4.21	5134.8
	ADA-C-10	100	43.35	4.01	469.3	100	43.64	4.01	1929.6
	ADA-C-20	100	45.50	4.16	887.2	100	46.27	4.17	3589.3
	ADA-C-30	100	46.12	**4.23**	1291.9	100	47.06	**4.24**	5136.1
	CW$_2$-0	98.44	45.42	3.92	23276.6	98.45	**47.18**	4.03	97101.1
	CW$_2$-5	100	43.77	3.81	23274.2	100	45.44	3.83	97105.7
	CW$_2$-10	100	42.59	3.70	23279.8	100	43.16	3.72	97095.4
Hard	FGSM	0.99	28.49	2.22	**101.5**	13.63	28.49	2.21	**419.7**
	PGD-10	97.22	32.93	2.87	468.9	99	33.06	2.84	1929.9
	PGD-20	99.33	31.90	2.76	886.8	99.33	32.05	2.74	3588.2
	PGD-30	99.79	31.57	2.73	1292.3	99.88	31.75	2.71	5136.8
	MIM-10	98.67	31.46	2.72	471.5	99.55	31.70	2.69	1944.3
	MIM-20	100	28.37	2.38	898.9	100	28.62	2.37	3605.1
	MIM-30	100	26.68	2.21	1296.6	100	26.89	2.20	5158.2
	ADA-E-10	99.44	36.85	3.37	368.2	100	37.22	3.35	1928.2
	ADA-E-20	100	37.00	3.39	887.5	100	37.49	3.39	3587.9
	ADA-E-30	100	37.90	3.51	1292.5	100	38.48	3.51	5134.5
	ADA-C-10	99.23	37.23	3.40	469.7	100	37.61	3.39	1929.8
	ADA-C-20	100	37.36	3.49	887.1	100	37.97	3.49	3589.3
	ADA-C-30	100	**38.56**	**3.67**	1292.1	100	39.41	**3.68**	5135.6
	CW$_2$-0	97.45	38.23	3.30	27656.2	97.89	**39.88**	3.44	110624.4
	CW$_2$-5	100	37.28	3.19	27660.1	100	38.78	3.35	111023.2
	CW$_2$-10	100	36.20	3.08	27658.7	100	37.51	3.21	110822.3

all benign examples but also add a lot of training time. FGSM AT showed the least decrease in accuracy for benign examples and the most decrease in adversarial training for PGD-10 AT.

TABLE 4. Experimental results of various attack algorithms under untargeted open-set speaker recognition.

Test Set	Attack Method	x-vector				i-vector			
		ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
$test_{10}$	FGSM	75.38	28.49	2.22	**100.1**	91.24	28.49	2.21	**420.7**
	PGD-10	100	33.03	2.87	469.2	100	33.13	2.84	1929.1
	PGD-20	100	31.94	2.76	887.3	100	32.08	2.74	3588.5
	PGD-30	100	31.6	2.73	1291.8	100	31.75	2.71	5136.7
	MIM-10	100	31.6	2.73	472.4	100	31.85	2.70	1943.1
	MIM-20	100	28.53	2.39	898.8	100	28.69	2.37	3602.8
	MIM-30	100	26.81	2.21	1295.3	100	26.92	2.19	5156.2
	ADA-E-10	100	43.84	4.03	470.5	100	44.94	4.05	1930.3
	ADA-E-20	100	45.16	4.14	886.7	100	46.71	4.17	3588.2
	ADA-E-30	100	46.07	4.20	1292.2	100	47.69	4.23	5135.9
	ADA-C-10	100	44.71	4.08	469.1	100	45.27	4.07	1930.2
	ADA-C-20	100	46.21	4.19	886.9	100	46.95	4.19	3588.1
	ADA-C-30	100	47.02	**4.27**	1292.1	100	47.94	**4.27**	5134.8
	CW_2-0	98.66	**48.55**	4.10	23301.3	98.88	**50.33**	4.20	96992.4
	CW_2-5	100	46.55	3.99	23297.2	100	48.32	4.08	97087.2
	CW_2-10	100	44.73	3.87	23302.9	100	46.49	3.98	97112.3
$imposter_{10}$	FGSM	80.60	27.98	2.39	**202.9**	81.70	27.98	2.39	**770.4**
	PGD-10	100	32.66	3.10	1084.3	100	32.78	3.10	3804.1
	PGD-20	100	31.64	2.97	2051.6	100	31.72	2.97	7251.5
	PGD-30	100	31.29	2.93	3016.6	100	31.36	2.93	10590.2
	MIM-10	100	31.37	2.95	1076.4	100	31.51	2.95	3815.3
	MIM-20	100	28.68	2.61	2052.1	100	28.76	2.61	7255.3
	MIM-30	100	27.12	2.42	3058.5	100	27.20	2.42	10592.3
	ADA-E-10	100	42.68	4.19	1067.9	100	43.15	4.21	3797.4
	ADA-E-20	100	43.78	4.27	2046.7	100	44.01	4.29	7249.3
	ADA-E-30	100	44.58	4.31	3019.2	100	44.71	4.33	10588.5
	ADA-C-10	100	42.21	4.12	1072	100	42.58	4.14	3798.2
	ADA-C-20	100	44.12	4.28	2024.3	100	44.42	4.29	7250.7
	ADA-C-30	100	44.64	4.33	3023.2	100	44.88	4.35	10589.6
	CW_2-0	99.20	**50.58**	**4.43**	51811.4	99.20	**51.14**	**4.45**	190489.2
	CW_2-5	100	48.48	4.35	51821.5	100	49.02	4.37	190512.9
	CW_2-10	100	46.09	4.21	51809.8	100	46.71	4.22	190387.5

TABLE 5. Experimental results of various attack algorithms under test 10 dataset with targeted open-set speaker identification.

Attack Level	Attack Method	x-vector				i-vector			
		ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
Simple	FGSM	47.01	28.49	2.22	**98.8**	60.97	28.49	2.21	**421.5**
	PGD-10	100	33.04	2.86	470.8	100	33.13	2.85	1930.4
	PGD-20	100	32.01	2.75	888.6	100	32.10	2.74	3589.9
	PGD-30	100	31.66	2.72	1292.5	100	31.75	2.71	5136.3
	MIM-10	100	31.61	2.71	473.1	100	31.77	2.71	1940.9
	MIM-20	100	28.76	2.40	899.8	100	28.88	2.40	3602.1
	MIM-30	100	27.17	2.23	1292.2	100	27.26	2.24	5157.1
	ADA-E-10	100	44.03	4.04	470.1	100	44.20	4.03	1929.5
	ADA-E-20	100	45.38	4.15	888.2	100	45.55	4.15	3586.3
	ADA-E-30	100	**46.21**	4.21	1292.4	100	46.44	4.21	5134.8
	ADA-C-10	100	44.19	4.05	469.6	100	44.55	4.06	1929.6
	ADA-C-20	100	45.50	4.16	886.5	100	45.91	4.17	3589.3
	ADA-C-30	100	46.12	**4.23**	1292.5	100	46.55	**4.24**	5136.1
	CW_2-0	98.89	44.18	3.84	23276.6	98.89	**47.18**	4.03	97101.1
	CW_2-5	100	42.86	3.74	23274.2	100	44.11	3.92	97105.7
	CW_2-10	100	41.52	3.62	23279.8	100	42.66	3.71	97095.4
Hard	FGSM	1.11	28.49	2.22	**99.2**	14.41	28.49	2.21	**419.7**
	PGD-10	99.55	32.77	2.86	469.2	100	32.89	2.87	1929.9
	PGD-20	100	31.81	2.75	887.3	100	31.92	2.74	3588.2
	PGD-30	100	31.49	2.71	1291.5	100	31.61	2.71	5136.8
	MIM-10	99.88	31.15	2.70	470.2	100	31.37	2.68	1944.3
	MIM-20	100	28.24	2.38	899.3	100	28.48	2.38	3605.1
	MIM-30	100	26.66	2.21	1296.1	100	26.86	2.21	5158.2
	ADA-E-10	99.55	35.14	3.19	469.1	100	35.75	3.21	1928.2
	ADA-E-20	100	35.41	3.25	887.2	100	35.75	3.25	3587.9
	ADA-E-30	100	36.01	3.35	1292.8	100	36.48	3.34	5134.5
	ADA-C-10	99.55	35.00	3.15	469.5	100	35.13	3.14	1929.8
	ADA-C-20	100	35.74	3.34	887.9	100	36.26	3.34	3589.3
	ADA-C-30	100	**36.77**	**3.52**	1293	100	37.46	**3.53**	5135.6
	CW_2-0	98.88	35.45	3.00	28026.2	98.88	**39.88**	3.44	110624.4
	CW_2-5	100	34.21	2.87	28033.1	100	36.31	3.19	111023.2
	CW_2-10	100	33.01	2.75	27998.7	100	34.36	2.99	110822.3

Second, the training time consumed by FGSM AT is the most time-efficient among these three approaches, yet the improved model robustness is the weakest among these three approaches, which cannot resist PGD-30, MIM-30,

TABLE 6. Experimental results of various attack algorithms under imposter 10 dataset with targeted open-set speaker identification.

Attack Level	Attack Method	x-vector				i-vector			
		ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
Simple	FGSM	99.3	27.98	2.39	**202.8**	98.9	27.98	2.39	**770.8**
	PGD-10	100	32.65	3.10	1084.7	100	32.76	3.09	3804.4
	PGD-20	100	31.62	2.97	2051.8	100	31.71	2.97	7251.7
	PGD-30	100	31.28	2.93	3016.9	100	31.35	2.93	10590.2
	MIM-10	100	31.36	2.95	1076.8	100	31.49	2.95	3815.7
	MIM-20	100	28.68	2.61	2052.2	100	28.68	2.61	7255.8
	MIM-30	100	27.11	2.42	3058.6	100	27.07	2.42	10592.6
	ADA-E-10	100	42.68	4.19	1068.2	100	43.12	4.21	3797.9
	ADA-E-20	100	43.79	4.27	2046.9	100	44.25	4.29	7249.6
	ADA-E-30	100	44.59	4.31	3019.2	100	46.85	4.32	10588.9
	ADA-C-10	100	42.21	4.12	1072.5	100	44.99	4.14	3798.5
	ADA-C-20	100	44.13	4.28	2024.7	100	44.57	4.30	7250.8
	ADA-C-30	100	44.64	4.33	3023.5	100	45.10	4.34	10589.5
	CW$_2$-0	100	**49.96**	**4.42**	51821.4	100	**51.67**	**4.45**	191262.2
	CW$_2$-5	100	48.16	4.34	51671.5	100	49.99	4.36	190851.4
	CW$_2$-10	100	45.55	4.17	51709.8	100	46.02	4.19	191167.8
Hard	FGSM	2.3	27.98	2.39	**203.5**	35.2	27.98	2.39	**771.4**
	PGD-10	100	31.54	3.02	1085.2	100	32.39	3.05	3805.1
	PGD-20	100	31.34	2.96	2052.2	100	31.46	2.94	7252.5
	PGD-30	100	31.02	2.92	3017.4	100	31.15	2.91	10593.2
	MIM-10	100	30.65	2.90	1078.2	100	30.91	2.89	3816.3
	MIM-20	100	27.94	2.56	2053.8	100	28.14	2.54	7257.3
	MIM-30	100	26.41	2.37	3059.4	100	26.56	2.36	10595.3
	ADA-E-10	100	36.94	3.70	1068.2	100	37.11	3.69	3798.4
	ADA-E-20	100	37.13	3.74	2047.6	100	37.33	3.75	7250.3
	ADA-E-30	100	**40.14**	4.04	3020.5	100	40.30	4.05	10588.5
	ADA-C-10	100	36.02	3.60	1073.7	100	36.35	3.59	3798.9
	ADA-C-20	100	38.42	3.92	2025.7	100	39.02	3.93	7250.7
	ADA-C-30	100	40.04	**4.09**	3023.7	100	**40.44**	**4.11**	10589.6
	CW$_2$-0	100	37.84	3.49	63221.7	100	38.99	3.61	234356.7
	CW$_2$-5	100	36.62	3.32	64419.8	100	37.75	3.45	234429.8
	CW$_2$-10	100	35.43	3.17	63356.5	100	36.56	3.32	234489.2

TABLE 7. Experimental results of various attack algorithms for speaker verification under imposter 10 dataset.

Attack Method	x-vector				i-vector			
	ASR	SNR	PESQ	TIME	ASR	SNR	PESQ	TIME
FGSM	55.1	27.98	2.39	**195.2**	64.4	27.98	2.39	**762.5**
PGD-10	100	32.46	3.08	1045.3	100	32.56	3.08	3793.3
PGD-20	100	31.49	2.96	1989.5	100	31.58	2.96	7301.2
PGD-30	100	31.18	2.92	2991.1	100	31.24	2.92	10585.9
MIM-10	100	31.03	2.93	1065.8	100	31.18	2.93	3801.1
MIM-20	100	28.36	2.59	1999.7	100	28.40	2.59	7310.5
MIM-30	100	26.81	2.40	2998.3	100	26.81	2.40	10590.9
ADA-E-10	100	42.41	4.18	1036.6	100	42.71	4.19	3790.6
ADA-E-20	100	43.58	4.26	1987.4	100	43.88	4.27	7295.4
ADA-E-30	100	44.41	4.30	2988.5	100	44.67	4.31	10580.5
ADA-C-10	100	42.04	4.11	1039.8	100	42.25	4.12	3792.7
ADA-C-20	100	43.9	4.27	1988.2	100	44.20	4.28	7297.8
ADA-C-30	100	44.41	**4.32**	2990.5	100	44.72	**4.33**	10582.4
CW$_2$-0	98.7	**45.97**	4.22	51813.2	99.20	**46.88**	4.31	191333.5
CW$_2$-5	100	43.62	4.07	51811.5	100	45.21	4.16	190943.7
CW$_2$-10	100	42.02	3.96	51822.8	100	43.17	4.05	191274.9

TABLE 8. Model robustness after adversarial training for various specific methods under untargeted speaker identification.

Defense Method	Training Time	Benign Example Accuracy	Attack Success Rate of Various Attack Methods			
			FGSM	PGD-30	MIM-30	ADA-C-30
NO defense	2750s	99.22	82.61	100	100	100
FGSM AT	10535s	96.74	**14.75**	100	100	99.96
PGD-10 AT	29275s	96.57	15.2	50.88	100	64.74
ADA-C-10 AT	20985s	96.67	22.58	**36.47**	100	**45.82**

and ADA-C-30 attacks, and only improves the resistance to FGSM attacks, which reduces the success rate of FGSM attacks by about 68%.

PGD-10 AT takes the longest time, about three times longer than FGSM AT and 1.5 times longer than ADA-C-10 AT, and its improved defensive effect is generally stronger than the counter training of FGSM AT, yet weaker than ADA-C-10 AT, specifically reducing the FGSM, PGD-30, and ADA-C-30 by about 67%, 50%, and 36%, respectively method's attack success rates.

And the proposed method for adversarial training in this paper not only takes less time and improves the model robustness overall the best among these three adversarial pieces of training, only slightly lower than the other two adversarial pieces of training in resisting FGSM attacks, yet the defense effect is still efficient, specifically reducing the attack success rate of FGSM, PGD-30, and ADA-C-30 methods by about 60%, 64%, and 55%, respectively, with significant improvement in resisting PGD and ADA-C attacks.

Finally, despite the adversarial training of different methods, it was not possible to improve the defense against MIM attack methods, and MIM was able to achieve a 100% attack success rate. The possible reason is that the adversarial examples generated by FGSM, PGD, and ADA-C are completely different from those generated by MIM methods, so although the adversarial training was performed to increase the diversity of model recognition examples, it was still not able to defend against MIM attack.

VII. CONCLUSION AND FUTURE DIRECTIONS

To explore the adversarial examples in the field of speaker recognition, this paper attacks two different speaker recognition models and reveals that there are serious security problems in speaker models. The proposed attack method compensates the shortcomings of traditional attack methods FGSM, PGD, MIM, and CW_2, greatly improves the stealthiness or reduces the generation time of adversarial examples and is applicable to all recognition tasks and different models. Finally, the proposed method is used for adversarial training, and its improved model robustness is generally better than the FGSM-based and PGD-based adversarial training.

The deficiency of this paper is that the research and experiment are carried out under the assumption of white-box attack, which has certain limitations. The next step will be to study speaker recognition under black box attacks and explore other defense methods besides adversarial training.

REFERENCES

[1] C. Zhang, K. Koishida, and J. H. L. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 1, pp. 11–23, Sep. 2018, doi: 10.1109/TASLP.2018.2831456.

[2] N. Maghsoodi, H. Sameti, H. Zeinali, and T. Stafylakis, "Speaker recognition with random digit strings using uncertainty normalized HMM-based I-vectors," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 11, pp. 1815–1825, Nov. 2019, doi: 10.1109/TASLP.2019.2928143.

[3] H. Ren, Y. Song, S. Yang, and F. Situ, "Secure smart home: A voiceprint and internet based authentication system for remote accessing," in *Proc. 11th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2016, pp. 247–251.

[4] L. Fan, Q.-Y. Jiang, Y.-Q. Yu, and W.-J. Li, "Deep hashing for speaker identification and retrieval," in *Proc. Interspeech*, Sep. 2019, pp. 2908–2912.

[5] A. Q. Ohi, M. F. Mridha, M. A. Hamid, and M. M. Monowar, "Deep speaker recognition: Process, progress, and challenges," *IEEE Access*, vol. 9, pp. 89619–89643, 2021, doi: 10.1109/ACCESS.2021.3090109.

[6] P. Univaso, "Forensic speaker identification: A tutorial," *IEEE Latin Amer. Trans.*, vol. 15, no. 9, pp. 1754–1770, Sep. 2017, doi: 10.1109/TLA.2017.8015083.

[7] C.-Y. Huang, Y. Y. Lin, H.-Y. Lee, and L.-S. Lee, "Defending your voice: Adversarial attack on voice conversion," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 552–559, doi: 10.1109/SLT48900.2021.9383529.

[8] R. J. Weiss, R. Skerry-Ryan, E. Battenberg, S. Mariooryad, and D. P. Kingma, "Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5679–5683, doi: 10.1109/ICASSP39728.2021.9413851.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[10] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, and D. Tao, "Perceptual-sensitive GAN for generating adversarial patches," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 1028–1035.

[11] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dec. 2016, pp. 1528–1540.

[12] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 1–7, doi: 10.1109/SPW.2018.00009.

[13] D. Park, H. Khan, and B. Yener, "Generation & evaluation of adversarial examples for malware obfuscation," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019, pp. 1283–1290, doi: 10.1109/ICMLA.2019.00210.

[14] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, vol. 52, no. 1, pp. 12–40, 2010.

[15] X. Yuan, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *Proc. USENIX Secur. Symp.*, Jun. 2018, pp. 49–64.

[16] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 15–20, doi: 10.1109/SPW.2019.00016.

[17] S. Khare, R. Aralikatte, and S. Mani, "Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization," 2018, *arXiv:1811.01312*.

[18] A. Madry, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193, doi: 10.1109/CVPR.2018.00957.

[20] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[21] C. Guo, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[22] J. Lin, C. Gan, and S. Han, "Defensive quantization: When efficiency me ETS robustness," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[23] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 72–85, Jan. 2021, doi: 10.1109/TDSC.2018.2874243.

[24] A. Jati, C.-C. Hsu, M. Pal, R. Peri, W. Abdalmageed, and S. Narayanan, "Adversarial attack and defense strategies for deep speaker recognition systems," *Comput. Speech Lang.*, vol. 68, Jul. 2021, Art. no. 101199, doi: 10.1016/J.CSL.2021.101199.

[25] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1962–1966, doi: 10.1109/ICASSP.2018.8462693.

[26] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM I-vector based speaker verification systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6579–6583, doi: 10.1109/ICASSP40776.2020.9053076.

[27] G. Chen, S. Chenb, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? Adversarial attacks on speaker recognition systems," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 694–711, doi: 10.1109/SP40001.2021.00004.

[28] A. S. Shamsabadi, F. S. Teixeira, A. Abad, B. Raj, A. Cavallaro, and I. Trancoso, "FoolHD: Fooling speaker identification by highly imperceptible adversarial disturbances," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6159–6163, doi: 10.1109/ICASSP39728.2021.9413760.

[29] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," in *Proc. Interspeech*, Oct. 2020, pp. 4228–4232.

[30] Q. Wang, P. Guo, S. Sun, L. Xie, and J. H. L. Hansen, "Adversarial regularization for End-to-End robust speaker verification," in *Proc. Interspeech*, Sep. 2019, pp. 4010–4014.

[31] G. Chen, Z. Zhao, F. Song, S. Chen, L. Fan, and Y. Liu, "SEC4SR: A security analysis platform for speaker recognition," 2021, *arXiv:2109.01766*.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210, doi: 10.1109/ICASSP.2015.7178964.

[33] N. Dehak, P. J. Kenny, R. Dehak, D. Pierre, and O. Pierre, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011, doi: 10.1109/TASL.2010.2064307.

[34] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5329–5333, doi: 10.1109/ICASSP.2018.8461375.

[35] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek, "Interpreting and explaining deep neural networks for classification of audio signals," 2018, *arXiv:1807.03418*.

[36] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, Aug. 2017, pp. 999–1003.

[37] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Jun. 2007, pp. 1–8, doi: 10.1109/ICCV.2007.4409052.

[38] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 4, pp. 1435–1447, May 2007, doi: 10.1109/TASL.2006.881693.

[39] Y. Long, W. Guo, and L. Dai, "An SIPCA-WCCN method for SVM-based speaker verification system," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Jul. 2008, pp. 1295–1299, doi: 10.1109/ICALIP.2008.4589961.

[40] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 155–176, 1996.

XINYU ZHANG was born in Zhejiang, China. He received the B.E. degree in network engineering from Ningbo University of Technology. He is currently pursuing the M.E. degree in cyberspace security with Guizhou Normal University, China. His research interest includes speaker recognition based on deep learning.

YANG XU was born in Shandong, China. He received the Ph.D. degree in computer software and theory from Guizhou University. He is currently a Professor and a Postgraduate Supervisor with the Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang, China. He is a Senior Member of China Computer Federation (CCF). His research interests include cybersecurity and deep learning.

SICONG ZHANG was born in Chongqing, China. He received the B.E. degree in electrical engineering and automation from the Civil Aviation University of China, the M.E. degree in computer science and technology from Guizhou Normal University, and the Ph.D. degree in software engineering from Guizhou University. He is currently a Lecturer and a Postgraduate Supervisor with the Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang, China. His research interests include cybersecurity and deep learning.

XIAOJIAN LI was born in Guangxi, China. He received the M.E. degree in computer science and technology from Guizhou Normal University, where he is currently pursing the Ph.D. degree. His research interests include cybersecurity and algorithm robustness.