Self-attention is What You Need to Fool a Speaker Recognition System

Fangwei Wang
College of Computer and Cyberspace Security
Hebei Normal University
Shijiazhuang, China
fw_wang@hebtu.edu.cn

Ruixin Song
College of Computer and Cyberspace Security
Hebei Normal University
Shijiazhuang, China
songrx@stu.hebtu.edu.cn

Zhiyuan Tan
School of Computing, Engineering and Built Environment
Edinburgh Napier University
Edinburgh, UK
Z.Tan@napier.ac.uk

Qingru Li
College of Computer and Cyberspace Security
Hebei Normal University
Shijiazhuang, China
qingruli@hebtu.edu.cn

Changguang Wang
College of Computer and Cyberspace Security
Hebei Normal University
Shijiazhuang, China
wangcg@hebtu.edu.cn

Yong Yang
Center of Information Technology
Yunnan University
Kunming, China
yy@ynu.edu.cn

***Abstract*—Speaker Recognition Systems (SRSs) are becoming increasingly popular in various aspects of life due to advances in technology. However, these systems are vulnerable to cyber threats, particularly adversarial attacks. Traditional adversarial attack methods, such as the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), are designed for a white-box setting where attackers have complete knowledge of the inner workings of the target systems. This limits the practicality of these attacks. To overcome this limitation, we propose a new attack model that uses a neural network to generate adversarial examples directly, without the need for full knowledge of the recognition model in a target SRS. In addition, we have designed a novel loss function to balance the effectiveness and confidentiality of adversarial examples. Our new approach was evaluated against SincNet, a state-of-the-art SRS. Experimental results show that our approach achieves outstanding performance, with the best attack success rate of 99.83% and the best Signal-to-Noise Ratio (SNR) value of 41.30.**

Keywords—speaker recognition systems, adversarial attack, adversarial example, information security

I. INTRODUCTION

Voice is the most direct and fast means for humans to communicate and exchange information. It is also a behavioral biometric that contains information about human identity, such as gender, age, and emotion [1]. Research has found that it is possible to distinguish different people by their voices because the information in each person's voice is completely distinct from each other [2]. Just like fingerprints, voiceprints can also be used as a marker of a person's identity and have become an important means of identity authentication. Therefore, Speaker Recognition Systems (SRSs) have been developed to identify people based on the unique characteristics of their voice, and have been embedded into various smart devices, nowadays.

However, with the widespread use of SRSs, security threats against this technique have gradually emerged. To our knowledge, voice conversion attacks [3], replay attacks [4] and synthesis attacks [5] have been the focus of previous offensive security research works before Deep Neural Networks (DNNs) were introduced to build SRSs in recent years. Since Szegedy et al. [6] discovered the vulnerability of DNNs, the research interest in attacks on deep neural networks has been growing. Nowadays, adversarial attacks against DNN-based SRSs [7, 8] have gained increasing attention from researchers. This type of attack can be broadly described as a perturbation that changes the classification of a recognition model, making the smart device believe that a certain voice of *Alice* belongs to *Tom*. This is because the original voice of *Alice* has been processed by attackers, who fine-tune some information about the voice and control these changes from being noticed by people. Chen et al. [9] summarize a variety of attack purposes. For example, someone illegally accesses a device without authorization or hides identity information when illegally accessing the device, or limits someone's right by forging voice information. This type of attack can cause irreversible damage to personal safety and property.

In the image domain, there are gradient-based methods for generating adversarial examples, such as L-BFGS [6], the Fast Gradient Sign Method (FGSM) [2], iterative FGSM (i-FGSM) [10], or Project Gradient Descent (PGD) [11], etc. Inspired by recent studies on adversarial attacks in the image domain, some scholars have begun to focus on the generation of adversarial audio examples. The method of generating adversarial examples in the image domain can be migrated to the audio domain. Just like in the image domain, in the audio domain the adversarial attacks are also categorized as white-box, black-box, and gray-box attacks. For white-box attacks [12, 13, 14] to achieve good performance, they require the adversary to acquire crucial knowledge such as the gradient and training parameters of the attacked target model. Therefore, these attacks are not applicable in actual scenes because the knowledge of the target model is difficult to obtain, especially from commercial systems that are often not released to the public.

Gray-box adversarial attacks have become a popular research topic due to the challenge of obtaining model information. Compared to white-box attacks, gray-box attacks are more practical and do not require complete information from the target model. Li et al. [12] presented a novel adversarial examples generation method that referenced the generating approach [15] and successfully introduced the generation of adversarial examples in the image domain to the audio domain. Inspired by their work, we propose our adversarial attack method, where we borrow the structure of the network [16] from the image domain and apply our newly designed perturbation

generator to directly transform the input signal waveform into adversarial audio examples for real-world attack scenarios. Furthermore, a new type of loss function is designed to guarantee a good balance between the quality of adversarial examples and the attack success rate. Our method ensures that the adversarial examples are similar to the original examples in terms of hearing, so the attack is not easy to detect by humans.

The main contributions of this research are:

- A new perturbation generator in which an attacker directly converts the input raw signal waveform into adversarial examples and uses them to attack the advanced SRSs with a higher attack success rate. This generator has also been shown to have the potential in real-time attack.
- A new loss function that can optimize the perturbation generator to balance the effectiveness and imperceptibility of the adversarial examples. This helps ensure that adversarial attacks are successful.

In this paper, we conducted experiments to test our proposed attack method against the well-trained state-of-the-art SRS called SincNet, using the TIMIT dataset. The results demonstrate that our method achieves an impressive attack success rate of 99.83%. Additionally, the best signal-to-noise ratio value reaches 41.30.

The rest of the paper is structured as follows. The related work on adversarial attacks is summarized in Section II. Section III provides our adversarial attack framework on a speaker recognition system. Section IV lists the experimental evaluation. Section V concludes this paper.

II. Related Work

Smart devices can be controlled through voice commands with SRSs installed in them. These systems identify users through valid voice information [17], making it convenient for those who cannot type and speeding up the identity verification process. SRSs are now commonly used in everyday life, and many smart devices featuring them as an identity verification method. The identity vector (i-vector) [18] based on the Gaussian Mixture Model (GMM) [19] is typically used in existing speaker recognition models. However, recent studies show that SRSs are now incorporating DNNs, where acoustic features are fed into neural networks to generate deep embeddings.

A new end-to-end CNN-based SRS, SincNet [20], implements band-pass filters with parametrized sinc functions. Unlike previous SRSs, SincNet uses neurons in the hidden layer to directly extract feature information from the original signal waveform instead of relying on hand-crafted features. We conducted a performance assessment of our attack method using SincNet as a target model.

Adversarial attacks are classified as white-box, black-box, and gray-box attacks based on the amount of information the attacker has about the target model [21, 22].

A. White-box Attack

To attack a target model, attackers can gain complete information about its internal structure, parameters, and defense mechanisms. One common method is to use gradients to design attack strategies. For example, some attackers use an approach under a white-box scenario to attack SRS [7, 8]. Other methods, such as the Adaptive Decay Attack (ADA) [23], focus on improving the confidentiality of adversarial examples. Furthermore, universal perturbation can be generated using internal information of the target model [13, 14, 24]. Then, a well-trained neural network called the Adversarial Transformation Network (ATN) that can directly transform input data into adversarial examples. However, it is important to note that launching such attacks heavily relies on having a lot of knowledge about the victim model, which is relatively impractical in real-world scenarios.

B. Black-box Attack

In a black-box attack, the attacker has no access to information about the victim and can only make assumptions about the internal workings based on its input and outputs. Researchers have proposed various methods to generate adversarial examples, such as SirenAttack by Du et al. [25] that uses the Particle Swarm Optimization (PSO) algorithm, and the optimization-based approach with constraints proposed by Chen et al. [26, 27, 28]. They also developed a threshold estimation method and a gradient estimation algorithm based on Natural Evolution Strategy (NES) to generate adversarial examples. Another technique is CC-CMA-ES by Zheng et al. [29], which uses a Cooperative Co-evolution (CC) framework in conjunction with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and has been successful against commercial systems in an absolute black-box environment. Deng et al. [30] recently proposed a decision-based method to attack SincNet that is effective in real-world scenarios. However, in a black-box setting, obtaining internal information about the victim model requires numerous queries, making black-box attacks a significant challenge.

C. Gray-box Attack

In this type of attack, the attackers only access partial information and cannot see the interior of the target. Confidence scores and feature representations may be obtained in such attacks [22, 31]. Li et al. [32] developed a generative network that can create universal adversarial perturbations in a gray-box environment. Furthermore, Zhang et al. [33] introduced a new voiceprint mimicry attack called VMask, which employs a gradient-based technique to produce adversarial perturbations. They used psychoacoustic masking to manage the disturbance, making it imperceptible to humans, thereby demonstrating the effectiveness in practical circumstances.

In Table I, we have compiled a summary of related work on the topic. From the table, it is evident that only a handful of studies have utilized neural networks to create adversarial examples, with SincNet being the target victim model.

TABLE I. Related work on adversarial attacks against SRSs

Methods	Type	Untargeted /	Target	Attack

		Targeted	Model	Method
Sirenattack [25]	Black	Targeted	ResNet18 VGG19	PSO
FAKEBOB [27]	Black	Both	i-vector Commercial Services	NES
Xie [14]	White	Targeted	x-vector	Gradient-based
Li [32]	Gray	Both	SincNet	Generative Network
Li [12]	White	Both	SincNet	Generative Network
Occam [29]	Black	Targeted	Commercial Services	CC-CMA-ES
Deng [30]	Black	Targeted	SincNet	Decision-based
AdvPulse [13]	White	Targeted	x-vector	Optimization-based
Two-step [24]	White	Targeted	VGG	Optimization-based
ADA [23]	White	Both	i-vector x-vector	Gradient-based
VMask [33]	Gray	Targeted	VGGVox	Gradient-based

III. Adversarial Attacks on Speaker Recognition Systems

A. Problem Formulation

Our aim is to create adversarial audio examples that can influence the target classifier to produce a classification that differs from the ground truth label. To achieve this, we introduce perturbations into the original clean inputs to obtain polluted speech. Although obvious changes in the original input can increase the success rate of adversarial attacks, they may also be noticeable due to the distinctiveness of the voice. From practical purposes, a sufficiently small perturbation makes it harder for the listener to detect any alteration in the sound, thereby ensuring the confidentiality of the attack. In this study, we propose an attacker network specifically designed for generating adversarial audio examples, with a loss function that regulates the distance between the adversarial example and the original example. This ensures the effectiveness of the attack and the imperceptibility of the adversarial examples.

Our study focuses on nontargeted attacks against SincNet [20]. Suppose that we have an input audio waveform S, which is depicted as the original example with its ground truth label t. Ordinarily, in case of no attack, SincNet accurately identifies the speaker when we input S. To attack the CNN-based speaker recognition system, SincNet, perturbations to S are introduced using the perturbation generator to create an adversarial example S'. If SincNet's prediction result for S' is anything other than t when S' passes through the target model, the attack is successful. We can formally define this attack problem as

$$F(S) = t,\ F(S') = y,\ \text{s.t. } D(S, S') \le \varepsilon, \tag{1}$$

where $F(\cdot)$ is a well-trained speaker recognition system, y is the predicted label of S'. $D(\cdot)$ is a distance metric used to calculate the similarity between S and S', and ε denotes a very tiny value for controlling the perturbation range.

B. The Attack Framework

The framework is comprised of three main components as shown in Fig. 1.

1) The generation of adversarial examples, represented by the blue rectangle, is heavily on the perturbation generator that creates an adversarial example S' of a clean example S to prepare for subsequent attacks. The details of the perturbation generator are discussed in Section III-D.

2) The target classifier, marked by the green rectangle, is referred to as F, whose details can be found in Section III-C. The classification result is indicated by $F(\cdot)$. In the event of an adversarial attack, the loss function is calculated using the result provided by the target classifier.

3) The loss function, shown in the red rectangle, consists of three elements which are detailed in Section III-E. When optimizing the loss function, the quality of the generated adversarial examples is progressively enhanced.

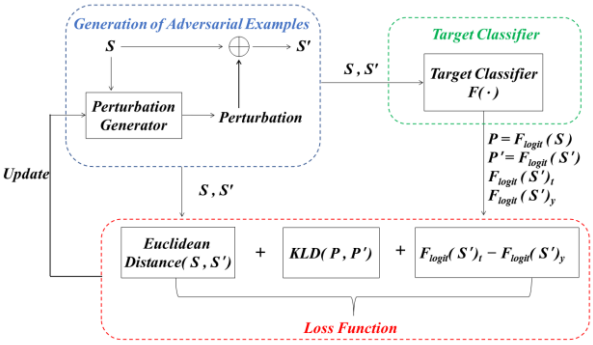

Fig. 1. The workflow of the whole attack framework.

C. Target Classifier

For this study, we utilized the publicly available pre-trained SincNet model from [20], which was trained on the TIMIT dataset [34], as the target classifier. Our experimental setup was identical to that used in [20]. SincNet replaces the first layer of a standard CNN with a set of learnable bandpass filters. The first layer of a network is critical for extracting low-dimensional features, which are necessary for higher-level networks to learn more useful feature information.

D. Perturbation Generator

We propose a method for creating adversarial examples through a perturbation generator, whose process is illustrated by Fig. 2. The original audio waveform is inputted into a neural network and processed through a multi-layer structure to produce an adversarial example directly, achieving a successful attack with minimal perturbations. Due to the high computational speed of neural networks, this method outperforms other approaches that rely on perturbation generation algorithms.

The perturbation generator is based on the Adversarial Imitation Network (AIN) structure [16], which incorporates the residual technique [35] and the self-attention mechanism. The model has two components: a convolutional Encoder and Decoder.

- The convolutional Encoder has five Enc blocks, each with 3-dimensional convolution, BatchNorm, and LeakyReLU, followed by a residual block. The Enc block encodes the input and outputs a representative state that has the same shape as the

input. Additionally, the self-attention module can comprehensively analyze all the input data, allowing the machine to focus on the relationships among different components of the input and assign high weights to essential information. Harnessing these relationships to their fullest during the training process leads to improved model training results. A self-attention block is placed between two Enc blocks to capture correlation between sampling points and allow for efficient encoding, which enable the encoder to encode the input into the representative vectors.

- The convolutional Decoder has a structure similar to the Encoder.

The encoder-decoder is not a universal way to generate adversarial audio examples. This structure has only appeared in the image domain and is not commonly used in the audio domain. In this work, the Encoder analyzes the input and produces a vector, which the Decoder uses to generate the output. Random noise is injected into the hidden layers of the neural network during training and testing to diversify the adversarial examples. In addition, the approach of introducing random noise directly into the original example lacks precision in regulating the magnitude of the added noise. Each data point in the dataset may require a different level of perturbation to achieve a successful attack. The arbitrary random noise can cause the detection of adversarial examples or, conversely, lead to unsuccessful attacks due to exceedingly subtle perturbations. However, the perturbation generator that we develop can automatically generate the optimal perturbations required to perform a successful attack.

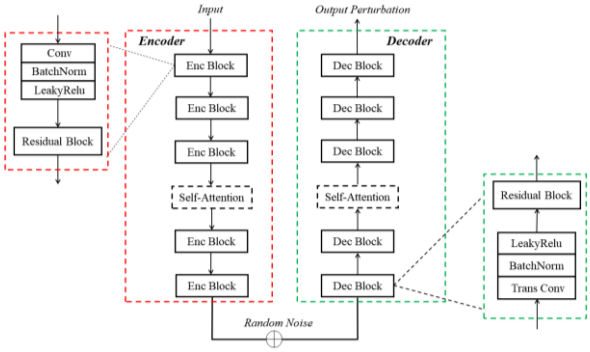

Fig. 2. The architecture of perturbation generator.

We aim for the perturbations created by the perturbation generator to be minimal, making them undetectable to humans. By directly introducing these perturbations to the original audio sampling values, we generate time-domain perturbations. To observe the size and frequency bands of the perturbations, we present the visualization results of the two examples. We can determine the smallest possible magnitude of the adversarial perturbation by comparing these results with those of the original and adversarial examples. If the range of the perturbation is insignificant, it proves that the inserted perturbation is small enough.

The waveforms before and after adding perturbations in the time domain are provided, which will make the stealthiness of the adversarial examples more intuitive. Additionally, researchers have made a discovery about human frequency perception—it is non-linear. The human ear is more sensitive to variations in low-frequency signals and less responsive to changes in high-frequency signals. To align with this observation, the Mel spectrogram employs a Mel scale crafted to match the characteristics of the human ear, preserving the essential information necessary for comprehending speech. If the audios are converted to Mel spectrograms, it becomes easier to identify these perturbations [31]. Thus, we convert both the original audio and adversarial audio into Mel spectrograms to observe the size and frequency bands of the perturbations. For a more detailed example, see Section IV.

E. Loss Function

We have also designed a loss function to improve the training of the perturbation generator. Our aim is to generate strong adversarial examples while maintaining a high level of audio similarity to the original examples. The loss function is a multitask function with three components, divided into two categories: Distance loss (with Euclidean Distance and Kullback-Leibler divergence) and Misclassification loss. The total loss is expressed by (2).

$$Loss = \lambda_1 Loss_1 + \lambda_2 Loss_2 + \lambda_3 Loss_3, \quad (2)$$

where λ_1, λ_2, and λ_3 denote the weight of each component, respectively.

1) Distance loss: The Distance loss is used to measure the distance between the original example S and the adversarial example S'. It ensures that the difference between them is small enough to maintain similarity. This is important to maintain the stealthiness of the adversarial examples. $Loss_1$ is calculated using speech sampling points, directly measuring the disparity between the two examples themselves. On the contrary, $Loss_2$ serves to restrict noise according to the distribution similarity of the recognition results. Both the original and adversarial examples are fed into the target classifier, producing sets of probability values for all the classes. This part of the loss function prevents the adversarial perturbations that we create from inducing excessive changes in the recognition results. The two components working together are shown to be more effective, whose detailed experimental results can be found in Section IV-B.

a) $Loss_1$: In the experiments, the examples are read as N-dimensional arrays, allowing for direct calculation of the Euclidean distance, as shown in (3).

$$Loss_1 = Euclidean\ Distance\ (S, S') \quad (3)$$

b) $Loss_2$: The $Loss_2$ is calculated using the Kullback-Leibler Divergence (KLD), which is a distribution similarity metric that helps to prevent excessive distortion of the results. Calculating the result of KLD can be challenging due to the vast number of classes in the dataset, leading to an extremely small probability value after the softmax layer. To address this, we utilize the pre-softmax layer (i.e., logit layer) of the classifier $F(\cdot)$ to determine the distance between two examples. For an input S, the output P of the logit layer means a distribution that contains all probability scores, and is indicated by $P = F_{\text{logit}}(\cdot)$. Likewise, P', denoting the respective output of the input S', can be calculated using the same equation.

$$Loss_2 = KLD\,(P, P') \quad (4)$$

2) Misclassification loss ($Loss_3$): This is to ensure the effectiveness of adversarial examples. For example, $F_{\text{logit}}(S')_t$ is the logit value at position t with input S'. Our aim is to minimize infinitely the value of $F_{\text{logit}}(S')_t$. To achieve this goal, we can maximize the output values in all categories except for t.

$$Loss_3 = F_{\text{logit}}(S')_t - F_{\text{logit}}(S')_y\,, \quad (5)$$

where $y = argmax_{\,i\in\{X\}-\{t\}} F_{\text{logit}}(S')_i$. X indicates all the indexes in the dataset.

IV. Experimental Results and Analysis

A. Experimental Settings

1) Dataset: The TIMIT [34] dataset is one of the most widely used datasets in the audio domain, featuring speakers from various regions in the United States with different dialects, genders, races, and education backgrounds. Each person in the dataset has 10 sentences, in which two are in the dialect class (SA), five are in the phonetically-compact class (SX), and three are in the phonetically-diverse class (SI). We followed the approach used in [20], with 462 speakers in the dataset, training on five SX sentences and testing on three SI sentences.

The target model, SincNet, works with the raw audio without any feature extraction. The audio files are read as N-dimensional arrays with different lengths. Our perturbation generator is adapted from the image domain, where two-dimensional convolution is used in the network. Therefore, we cannot directly use the data in the network. To address this issue, we first cut the audio files into a fixed length to resize the data. This ensures that the number of sampling points in the audio is suitable for the neural network.

2) Evaluation Metrics: a) Attack Success Rate (ASR): The contaminated adversarial examples are not recognized as the ground-truth labels, indicating a successful adversarial attack. This metric measures the ratio of successful adversarial attacks to the total number of attacks. *b) Signal-to-Noise Ratio (SNR):* The formula is expressed as $SNR = 10\,log_{10}(\frac{Ps}{Pn})$, where Ps and Pn represent the effective power of signal and noise, respectively. This metric is used to adjust the magnitude of the noise. A higher SNR indicates a better quality of adversarial examples. *c) Generation time:* The generation time of the adversarial example, which is a metric of the effectiveness of an attack. A shorter time to generate adversarial examples means that we launched a successful real-time attack.

B. Experimental Results of Adversarial Attack

This section discusses the effectiveness of our proposed attack method. Experiments were carried out using a system that had PyTorch 1.2, a 2080 Ti-11G GPU, and an Intel Xeon E5-2696 v2 CPU. Table II reveals the experimental results of our method.

1) Effectiveness of the adversarial attack: To validate the effectiveness of our attack, the baseline data of SincNet (target model) is presented. Under no adversarial attack, the recognition accuracy of the target model is 98.48%. However, as shown in Table II, the recognition accuracy of the target model drops to about 0.3% in the event of an adversarial attack, suggesting that our proposed method has significant impacts on the target SRS.

TABLE II. Experiential results by fine tuning $\lambda_1 : \lambda_2 : \lambda_3$

$\lambda_1 : \lambda_2 : \lambda_3$	SNR (dB)	ASR (%)	Generation Time (s)
0 : 0 : 1	37.21	96.10	
1 : 0 : 1	37.22	96.32	
0 : 1 : 1	39.43	94.59	
1 : 1 : 1	36.99	99.78	
1 : 1: 100	38.03	98.65	0.7
1 : 100 : 1	41.30	95.90	
100 : 1: 1	37.20	95.96	
100 : 1 : 100	37.21	95.89	
1 : 100 : 100	36.71	99.83	

2) Impacts of the components of the loss function: In this experiment, the three losses differ in orders of magnitude. Adding them together to obtain a total loss function for network training may cause different convergence speeds due to their different weights. This will lead to a situation when we observe that the total function tends to be stable, while a component with an order of magnitude smaller has yet to reach a stable state. To address this issue, we standardize the magnitudes of the three components in the total loss function during the training process. Furthermore, the loss function that we use consists of three components, each having a specific impact on the attack performance. To understand these impacts, we assigned different weights to conduct ablation studies. As shown in Table II, the weights are set to 0, 1 and 100 respectively, enabling us to evaluate the impact of each component on the outcomes.

The two components, $Loss_1$ and $Loss_2$, work together to minimize the distance between clean and adversarial examples. Adjusting the weights of these components during training by increasing λ_1 or λ_2 can greatly enhance the quality of the adversarial examples. The experimental results show that $Loss_2$ is more effective in improving the auditory quality of the adversarial examples. Removing either λ_1 or λ_2 results in a significant drop in the SNR and ASR compared to the best results. On the other hand, $Loss_3$ is used to deceive the target model, making it misclassify the examples. It, therefore, controls the effectiveness of the attack. Increasing the weight of λ_3 improves the ASR but sometimes comes at the expense of a lower SNR. However, when both misclassification and distances are scaled up equally, the changes in results are minimal. For instance, when both λ_2 and λ_3 are increased by a factor of 100, as seen in the fourth and ninth rows of Table II, the results barely change. This suggests that the two components may counterbalance each other, leading to relatively complex results.

3) Stealthiness of the adversarial attack: We have evaluated stealthiness by comparing the visualization results of the original and adversarial examples from different speakers. In each subfigure of Fig. 3, the images on the left come from the original audios, while those on the right are from the adversarial examples. The target model misclassifies the generated adversarial example as an arbitrary speaker, whose ID is called the predicted label. The ground-truth label of the original audio and the predicted label of the adversarial example are shown in each image of Fig. 3.

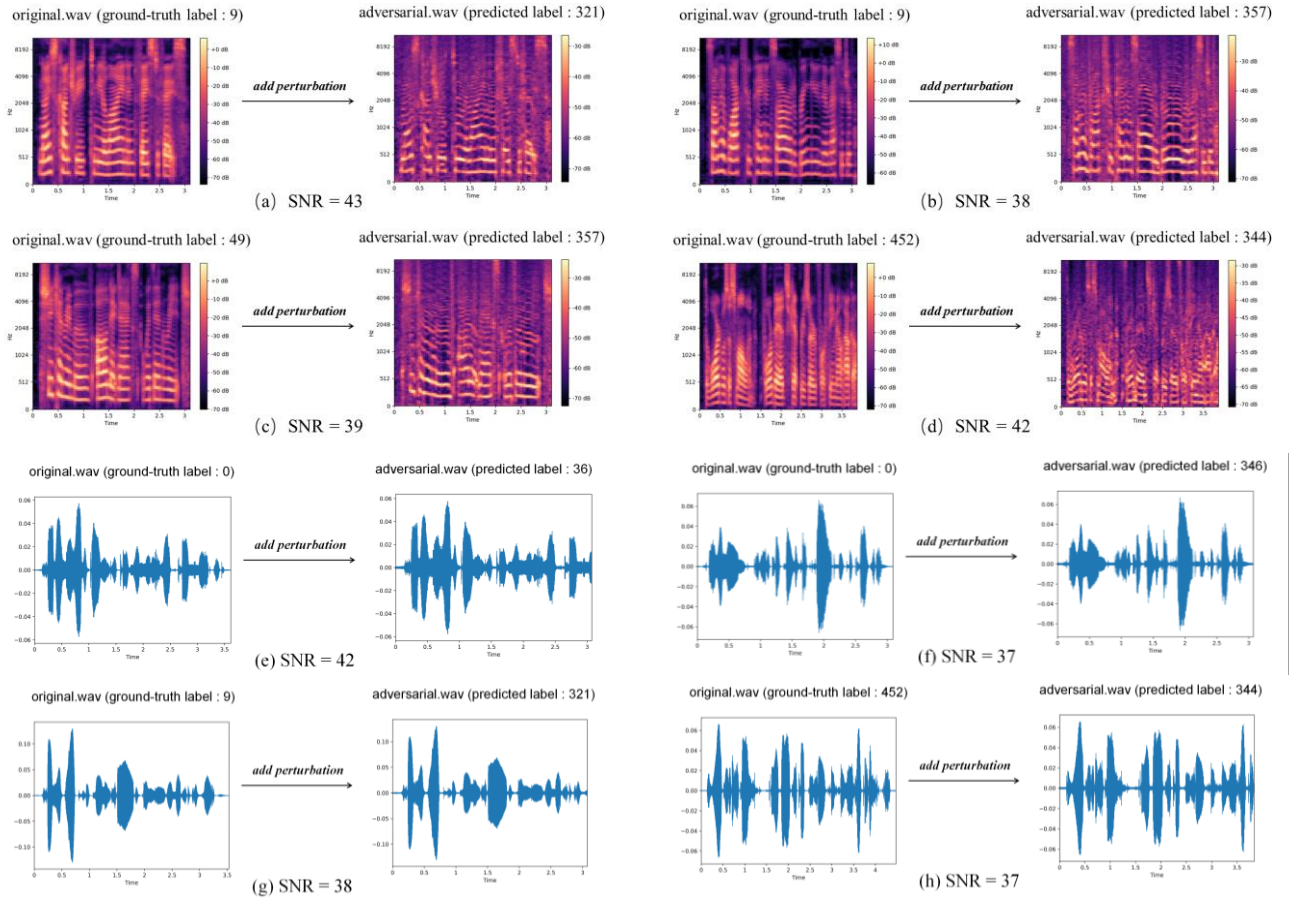

Fig. 3. The visualization results of original examples and adversarial examples.

Additionally, the SNR, which indicates the magnitude of the perturbation, is calculated for each pair of original and adversarial audios. These images demonstrate that our method can generate an adversarial example with high similarity to the original example, regardless of the speaker of the audio.

4) Efficiency: Our perturbation generator offers fast training speeds, with the entire process taking only 3 epochs and around 3 hours to achieve a stable state in the neural network. Additionally, the perturbation generator does not require gradients during the testing phase, resulting in a short average generation time of about 0.7 seconds for an adversarial example. This indicates the potential for a real-time attack.

C. Performance Comparison

In this section, we compare the results with other studies, among which Li et al. [32] developed a generative network to synthesize different Universal Adversarial Perturbations (UAPs) against SincNet. Similarly, to our experimental approach, their study also adjusted the parameters of the optimization function. Besides, our method acquires the feedback by querying the target model, and continuously adjusts training parameters based on the feedback information. So our proposed method also belongs to gray-box attack. As this article shares an experimental scenario with our work, we believe that comparing our results with this study is more appropriate to demonstrate the performance of our work. There are many results in [32], and we have selected the results that correspond to the adversarial examples with the strongest attack capability for comparison. Table III shows the ASR and its corresponding SNR. The success of an attack is heavily on the intrusiveness of the adversarial example. The experimental results demonstrate that our method can achieve a superior ASR of 99.83%, which is 2% higher than the compared work. This implies that our method has a higher success rate of attack, albeit with a slightly lower SNR.

Furthermore, our method offers a diversity of adversarial examples, as illustrated in Fig. 3. Even when using audio from the same speaker, we can produce a variety of adversarial audios that the target model misclassifies as different speakers, making the attack more diverse difficult to control. This feature ensures that the attack remains effective in real-world scenarios, and not limited to a single type of attack.

TABLE III. THE PERFORMANCE COMPARISON OF ADVERSARIAL EXAMPLES GENERATED BY DIFFERENT METHODS

Methods	Dataset	Target Model	ASR (%)	SNR (dB)
UAPs [32]	TIMIT	SincNet	97.50	44.13
Our method	TIMIT	SincNet	99.83	36.71

V. POTENTIAL DEFENSE METHODS

This part explores the feasible defense methods that could effectively defend against our proposed attack. Our approach is query-based. It becomes difficult for the attacker to succeed when the target classifier has the defense mechanism which restricts the number of queries.

Besides, adversarial training is a currently widely used defense technique, which integrates adversarial examples into the original dataset to retrain the neural network to improve the robustness of the model. However, the adversarial examples used to train the model cannot be updated in time owing to the attacker frequently changes the attack parameters, which will also diminish the defensive efficacy of adversarial training.

VI. Conclusion and Future Work

There has been an increase in the use of deep learning-based systems in smart devices, which has led researchers to investigate adversarial attacks in the field of speaker recognition systems. This paper proposes a perturbation generator for creating adversarial audio examples that are characterized by high intensity, good confidentiality, diversity, and fast generation speed. Our method overcomes the limitations of traditional methods that rely heavily on internal model information. It is applicable in scenarios where attackers cannot access the model parameters and gradients. The experimental results show that our method outperforms other closely related research.

However, our proposed method also has certain limitations. Our query-based approach may become detectable if there are too many queries. Moreover, in practical scenarios, some devices will have defense mechanisms that limit the number of queries. Moving forward, we plan to explore the generation of adversarial examples in restricted black-box scenarios, as to well as focus on physical attacks in real environments. In these scenarios, adversarial examples can only be generated by surrogate models, rather than using feedback information from the target model. Further, adversarial audio examples are played through loudspeakers, and we will study issues such as distortion caused by air propagation.

Acknowledgement

This research was funded by NSFC under Grant 61572170, Natural Science Foundation of Hebei Province under Grant F2021205004, Science and Technology Foundation Project of Hebei Normal University under Grant L2021K06, Science Foundation of Returned Overseas of Hebei Province Under Grant C2020342, and Key Science Foundation of Hebei Education Department under Grant ZD2021062.

References

[1] R. M. Hanifa, K. Isa, and S. Mohamad, "A review on speaker recognition: technology and challenges," *Comput. Elec. Eng*, vol. 90, no.3, pp.107005, 2021.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint*, arXiv: 1412.6572, 2014.

[3] T. Kinnunen, Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4401-4404.

[4] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2014, pp. 1-5.

[5] J. Lindberg, and M. Blomberg, "Vulnerability in speaker verification-a study of technical impostor techniques," in *Proceedings of the European Conference on Speech Communication and Technology*, 1999, pp.1211-1214.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint*, arXiv: 1312.6199, 2013.

[7] Y. Gong, and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications, " *arXiv preprint*, arXiv: 1711.03280 2017.

[8] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples, " in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 1962-1966.

[9] G. Chen, Z. Zhao, F. Song, S. Chen, L. Fan, and Y. Liu, "AS2T: arbitrary source-to-target adversarial attack on speaker recognition systems, " *arXiv preprint*, arXiv: 2206.03351, 2022.

[10] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceedings of the Workshop of the 5th International Conference on Learning Representations*, 2017, pp. 99-112.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, " Towards deep learning models resistant to adversarial attacks," *arXiv preprint*, arXiv: 1706.06083, 2017.

[12] J. Li, X. Zhang, J. Xu, S. Ma, and W. Gao, "Learning to fool the speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 2937-2941.

[13] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "AdvPulse: universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1121-1134.

[14] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," in *Proceedings of the 2020 IEEE ICASSP*, 2020, pp. 1738-1742.

[15] S. Baluja, and I. Fischer, "Learning to attack: adversarial transformation networks," in *Proceedings of the Annual AAAI Conference on Artificial Intelligence*, 2018, pp. 2687-2695.

[16] P. Tang, J. Lou, and L. Xiong, "Generating adversarial examples with distance constrained adversarial imitation networks," *IEEE Trans. Depend. Sec. Comput*, vol. 19, no. 6, pp. 4145-4155, 2022.

[17] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE international conference on acoustics, speech, and signal processing*, 2002, pp. 4072-4075.

[18] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Proc*, vol. 19, no. 4, pp. 788-798, 2011.

[19] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Dig. Sig. Proc*, vol. 10, no. 1, pp. 19-41, 2000.

[20] M. Ravanelli, and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021-1028.

[21] J. Lan, R. Zhang, Z. Yan, J. Wang, Y. Chen, and R. Hou, "Adversarial attacks and defenses in speaker recognition systems: a survey," *J. Sys. Arch*, vol. 127, no. 5, pp. 102526, 2022.

[22] C. Yan, X. Ji, K. Wang, Z. Jiang, and W. Xu, "A survey on voice assistant security: attacks and countermeasures," *ACM Comput. Surv*, vol. 55, no. 4, pp. 1-36, 2022.

[23] X. Zhang, Y. Xu, S. Zhang, and X. Li, "A highly stealthy adaptive decay attack against speaker recognition," *IEEE Access*, vol. 10, no. 11, pp. 118789-118805, 2022.

[24] W. Zhang, S. Zhao, L. Liu, J. Li, X. Cheng, T. Zheng, and X. Hu, "Attack on practical Speaker Verification System Using Universal Adversarial Perturbations," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 2575-2579.

[25] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "Sirenattack: generating adversarial audio for end-to-end acoustic systems," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 357-369.

[26] G. Chen, Z. Zhao, F. Song, S. Chen, L. Fan, F. Wang, and J. Wang, "Towards understanding and mitigating audio adversarial examples for speaker recognition," *arXiv preprint*, arXiv: 2206.03393, 2022.

[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 *2021 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 694-711.

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

[29] B. Zheng, P. Jiang, Q. Wang, Q. Li, C. Shen, C. Wang, Y. Ge, Q. Teng, S. Zhang, and S. Zhang, "Black-box adversarial attacks on commercial speech platforms with minimal information," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 86-107.

[30] J. Deng, L. Dong, R. Wang, R. Yang, and D. Yan, "Decision-based attack to speaker recognition system via local low-frequency perturbation," *IEEE Signal Process Lett*, vol. 29, no. 6, pp. 1432-1436, 2022.

[31] H. Tan, L. Wang, H. Zhang, J. Zhang, M. Shafiq, and Z. Gu, "Adversarial attack and defense strategies of speaker recognition systems: a survey," *Electronics*, vol. 11, no. 14, pp. 2183, 2022.

[32] J. Li, X. Zhang, C. Jia, J. Xu, L. Zhang, Y. Wang, and W. Gao, "Universal adversarial perturbations generative network for speaker recognition," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1-6.

[33] L. Zhang, Y. Meng, J. Yu, C. Xiang, B. Falk, and H. Zhu, "Voiceprint mimicry attack towards speaker verification system in smart home," in *Proceedings of the 39th IEEE Conference on Computer Communications (INFOCOM 2020)*, 2020, pp. 377-386.

[34] H. Tan, K. Liang, Y. Lee, C. Li, Y. Li, and J. Wang, "Speech separation using augmented-discrimination learning on squash-norm embedding vector and node encoder," *IEEE Access*, vol. 10, no. 7, pp. 102048-102063, 2022.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.