Defend Data Poisoning Attacks on Voice Authentication

Ke Li
Vanderbilt University
ke.li.1@vanderbilt.edu

Cameron Baird
Vanderbilt University
cameron.baird@vanderbilt.edu

Dan Lin
Vanderbilt University
dan.lin@vanderbilt.edu

ABSTRACT

With the advances in deep learning, speaker verification has achieved very high accuracy and is gaining popularity as a type of biometric authentication option in many scenes of our daily life, especially the growing market of web services. Compared to traditional passwords, “vocal passwords" are much more convenient as they relieve people from memorizing different passwords. However, new machine learning attacks are putting these voice authentication systems at risk. Without a strong security guarantee, attackers could access legitimate users' web accounts by fooling the deep neural network (DNN) based voice recognition models. In this paper, we demonstrate an easy-to-implement data poisoning attack to the voice authentication system, which can hardly be captured by existing defense mechanisms. Thus, we propose a more robust defense method, called Guardian, which is a convolutional neural network-based discriminator. The Guardian discriminator integrates a series of novel techniques including bias reduction, input augmentation, and ensemble learning. Our approach is able to distinguish about 95% of attacked accounts from normal accounts, which is much more effective than existing approaches with only 60% accuracy.

KEYWORDS

voice authentication, deep neural networks, data poisoning attacks

ACM Reference Format:
Ke Li, Cameron Baird, and Dan Lin. 2022. Defend Data Poisoning Attacks on Voice Authentication. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Speaker verification (or voice authentication) is a process that verifies the identity of the speaker based on his/her voice. To some people, such “vocal passwords" might not seem to be as common as PIN codes and facial authentication. However, speaker verification has already been adopted in many scenes for a long time. Since the 1980s, law enforcement and jurisdiction departments have utilized voice verification technologies to identify suspects and provident crimes [2, 42]. Financial service institutions have also used voice authentication as one of the verification methods for years [38]. Today, with the fast growth of Internet-of-Things (IoT) and voice assistance systems such as Apple Siri, Google Assistant, and Amazon Echo, speaker verification is increasing in popularity. By simply saying “Hi" to the system, a user can easily access his/her account and receive personalized services.

Compared to the traditional passwords or PIN codes, using the "vocal passwords" would be much more convenient for customers to access their smart devices as well as a large number of web services available on the market. With vocal passwords, customers will no longer need to create and remember various passwords for different accounts; simultaneously, it helps mitigate the risks of leaking or forgetting the passwords. Compared to the recent facial authentication systems, which bring similar convenience, speaker verification has its own unique advantages. First, the voiceprint of a human is quite stable after adulthood [14, 26] whereas facial features may change once a while because of various factors, such as aging, growing beard, or wearing new makeup. That means users might need to update facial authentication more frequently than vocal authentication in order to ensure accuracy. Second, the hardware cost for deploying voice authentication is lower than that of facial authentication. Voice data is typically smaller than facial data, and hence needs less storage space. Microphones used to collect voices are also cheaper than high-resolution cameras needed for facial authentication. These advantages are propelling the growth of the global voice biometrics market which was valued at USD 0.69 billion in 2018 and is expected to reach USD 3.91 billion by the year 2026 [1].

Speaker verification technology has been investigated for nearly 40 years, ranging from the earlier MFCC [28, 29, 50] and GMM [6] models to the state-of-the-art deep neural network (DNN) models such as D-vector [17] and Deep-Speaker [24]. The DNN-based models have exhibited high accuracy (e.g., 95%) of voice verification. While enjoying the burgeoning performance, DNN-based models are known to be much more vulnerable to new machine learning attacks such as adversarial input attacks and data poisoning attacks than traditional speaker verification models [6, 29, 48]. Both kinds of attacks aim to mislead the DNN models to misclassify the input data. Adversarial input attacks [27, 45] achieve the goal by perturbing the input data while data poisoning attacks use poisoned training data to manipulate the victim DNN model. For facial or voice authentication systems that are developed upon DNN models, such machine learning attacks impose severe threats to the web service quality and customer information security. For example, some general attacks attempt to lower the overall accuracy of the authentication system and cause a large number of legitimate users not able to log into their accounts. Targeted attacks are even more concerning as attackers may impersonate a legitimate user to access the user's account. Although some countermeasures have been proposed to defend these attacks on DNN-models for image classification and facial recognition [8], they do not work well in defending voice recognition models (as shown in our experiments) due to the fundamental structural differences between the image

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

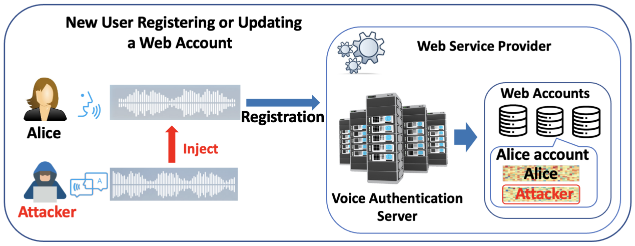


Figure 1: Data Poisoning Attack on Voice Authentication Systems

data and voice data. Also, it is worth noting that adversarial input training [11, 27, 45] is not an applicable defense for this targeted data poisoning attack since the poisoned data is still a normal audio file and does not contain any perturbed values.

In this work, we aim to tackle a challenging data poisoning attack on the voice authentication process whereby an attacker intends to gain access to the targeted victim's account through voice authentication. As shown in Figure 1, the targeted data poisoning attack may occur during the stage of new user registration or user account update. Specifically, to use the voice authentication, the user needs to provide several different utterances to let the web service authentication system learn his/her voice features. If during the uploading of these training audio files, an attacker injects or replaces some of the user's audio files with his own, our experiments found out an astounding fact that the voice authentication system would be easily misled to consider both user's and the attacker's voices as legitimate and grant access to both the real user and the attacker when hearing their voices. In other words, the attacker would be able to peek and use the user's account without being noticed by the real user until the damage is done. It is worth noting that such an attack is not hard to implement. It is similar to the recent discussion on injection attacks on facial authentication [8], where an attacker may exploit the vulnerabilities of the victim's home network and router to inject malicious packets [33, 40].

In order to protect the integrity of the voice authentication from the aforementioned targeted data poisoning attack, an intuitive idea could be to compare the attacker's audio files with the real user's audio files, and check if there are any differences that can be utilized to filter out the attacker's audio files. Unfortunately, experiments show that no significant differences in the data distribution of the attacker's and the victim's raw audio files can be found from observing the popular t-distributed stochastic neighbor embedding (t-SNE). Alternatively, one may think of checking the differences in the feature vectors generated by the voice recognition system. Again, no significant differences can be identified by using t-SNE. This indicates that more advanced approaches are needed to detect the attackers. In this paper, we propose a deep neural network (referred to as Guardian) that is capable of distinguishing the feature vectors of poisoned audio files from those of non-poisoned audio files with more than 95% accuracy. Our approach is generic to any DNN-based voice authentication model. In the experiments, we select the popular Deep Speaker model [24] which has a very high voice recognition accuracy (95%) as the attacker's target. Since targeted data poisoning attacks typically aim at only a few victims at a time to avoid the overall accuracy degeneration of the speaker verification model which otherwise will raise system alerts, the ratio between the poisoned data and non-poisoned data is very low. In order to avoid domain bias, we trained multiple speaker verification models with different victims selected for each model, and collected a balanced set of poisoned and non-poisoned feature vectors for the Guardian network's training. Furthermore, we propose an input augmentation approach that combines the poisoned and non-poisoned feature vectors in a way that can better help Guardian network learn the differences between them. The Guardian network contains two convolutional layers and 2 fully connected layers. Once the Guardian network is fully trained, it will not need to be retrained when the speaker verification model is launched in the field for user registration. Given a new user registration input, the speaker verification model will feed the feature vectors of user's utterances to the Guardian network which will then output a decision whether the user files contain poisoned data or not. To sum up, we have made the following contributions:

- We studied the impacts of targeted data poisoning attacks on voice authentication. Our experiments demonstrate a very high success rate of such attacks.
- We designed a Guardian neural network that can effectively defend the targeted data poisoning attack.
- We conducted extensive experimental studies on real datasets. The results show that our proposed Guardian network is much more accurate than statistical outlier detection approaches, traditional machine learning algorithms, and the latest defense mechanisms for CNN-based facial authentication models.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 demonstrates the targeted data poisoning attacks. Section 4 presents the proposed Guardian network. Section 5 reports the experimental results. Section 6 conducts the security analysis. Finally, Section 7 concludes the paper.

2 RELATED WORK

Existing machine learning attacks can be classified into three main categories, which are adversarial input attacks [11, 27, 45, 48], data poisoning attacks [7, 8, 22, 43, 47, 48] and model stealing attacks [21, 30]. Since our work is defending a type of data poisoning attack, we mainly review the existing defenses against data poisoning attacks in the following.

2.1 Data Poisoning Attacks

The data poisoning attack happens during the model training stage whereby the attacker injects poisoned training samples in order to mislead the classifier to assign wrong labels to some testing data. The typical attack procedure is the following. In many applications of machine learning, such as authentication systems, the training data is non-stationary. Both the joining of new users and the leaving of old users will affect the distribution of the whole dataset. In order to handle such non-stationary data distribution, the classifier typically needs to be retrained periodically [23]. When the classifiers are retrained on new samples collected during network operation, it gives the attacker a chance to inject poisoned samples into the training dataset.

According to the effect of the attack, the data poisoning attacks can be divided into two categories: availability attacks and integrity attacks [5, 18, 25]. The availability attack could be considered as an untargeted attack that does not aim at a particular target as it just aims to degrade the overall performance of the classifier. On the contrary, the integrity attacks do not want to affect the overall performance of the classifier to arouse alert. Instead, the integrity attacks have clear targets for which they want to misclassify. Our work is tackling such targeted attacks.

There are two main approaches to generating the poisoned training samples [5, 41]. One is to perturb the original sample using algorithms such as the well-known Fast Gradient Sign Attack (FGSM) [13] to make the classifier assign attacker desired label to the perturbed sample. The other approach is to manipulate the training sample by changing its correct label to a wrong one [43]. The data poisoning attack launched to the voice authentication system belongs to this second category. However, we would like to stress that there has not been any study on such attacks on voice authentication systems yet, not to mention the corresponding defense mechanisms.

2.2 Existing Defenses Against Data Poisoning Attacks

A popular defense mechanism against data poisoning attacks is adversarial input training [4, 13, 27, 44, 45, 48]. The key idea is to add one more class label and train the classifier using the perturbed samples generated by the same algorithm that the attacker may use such as FGSM or PGD (projected gradient descent) [13, 16]. The goal is to help the classifier learn the features of poisoned data along with other normal data so that the classifier may be able to distinguish the poisoned data from normal data in the future. However, such defense mechanisms will not be applicable in our attack scenario. This is because adversarial input training uses the real samples injected with carefully crafted noises that can mislead the classifier. In our attack scenario, the attacker does not insert any noise into his/her voice file. The attacker simply labels his/her voice file using the victim's identity. If one wants to directly apply the adversarial input training here, some voice files need to be randomly selected to pretend to be attackers and be labeled as “adversarial." Note that these voice files are normal audio files without noises. This type of training is simply telling the voice recognition model to classify the preselected attacker files as “adversarial" while the voice recognition model gains no knowledge about what the true attackers' voice features may look like in the real world.

Another well-known defensive approach is outlier detection, also known as data sanitization [12, 18, 22, 32, 34, 43, 44]. For example, Dai et al. [10] found that poisoned data and non-poisoned data may follow different distributions, and employ principal component analysis (PCA) to filter out the poisoned data. Other types of classifiers such as SVM and KNN have also been explored to detect the outliers, i.e., poisoned data [22, 35–37, 39]. However, as shown in our experimental studies, all of these outlier detection techniques are not effective in identifying the attacked account from normal accounts due to the highly similar data distributions between the feature vectors from the attacked accounts and the normal accounts.

The most related work to ours is by Cole et al. [8] who examines a targeted data poisoning attack in facial authentication systems by assuming that attackers may inject their own facial images into the user registration phase similar to the injection of attacker's audio files in our scenario. They propose a DNN model called DEFEAT to distinguish the attacked accounts from the normal accounts. However, our work is more advanced than the DEFEAT model in several aspects. Specifically, the DEFEAT model mainly utilizes fully connected layers while fully-connected-layer based structure is not effective in detecting attacked accounts in voice authentication systems as shown in our experimental studies. Our proposed Guardian network not only leverages convolutional layers but also incorporates new input augmentation and ensemble learning techniques which lead to 90% detection accuracy.

3 DATA POISONING ATTACK ON VOICE AUTHENTICATION SYSTEMS

In this section, we first introduce our threat model, and then present the results of the targeted data poisoning attacks against DNN-based voice recognition systems.

3.1 Threat Model

In this work, we follow the same threat model by the recent work of targeted data poisoning on facial authentication [8]. The attacker's goal is to deceive the voice authentication system into recognizing the attacker's voice and the legitimate user's voice (the victim) as the same so that the attacker can gain the access to the victim's web account via voice authentication. It is assumed that the attacker has compromised the victim's home network [3, 33, 40] and is able to inject malicious messages when the victim communicates with web services. The attack may trick the user to update his/her registration information by replacing the normal access page to the web service with a registration update request, just like a traditional password update request. Once the victim starts to update his/her voice authentication, the attacker will replace a few of the victim's audio files with his own audio files to be sent to the web service provider. In this targeted data poisoning attack, there are three parties:

- **Normal users (U_n)**: Normal users are those who have not been attacked and whose voice authentication is correctly performed by the voice recognition systems.
- **Victim users (U_v)**: Victim users are those whose voice authentication has been poisoned by an attacker.
- **Attackers (U_a)**: Attackers are those who conduct the data poisoning attack on the victim user's registration.

As a result of the attack, the voice authentication system at the web service provider site will be trained using both the victim's and the attacker's audio files to register the victim's account. The attacker does not need to know any specific parameters of the voice authentication model at the server side. Such an attack is considered successful if both the victim and the attacker can access the same account via the voice authentication (as illustrated in Figure 2). In other words, the attacker would be able to use his/her own voice to log into the victim's web account from any places later on.

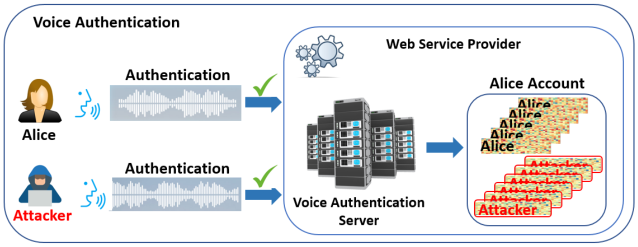
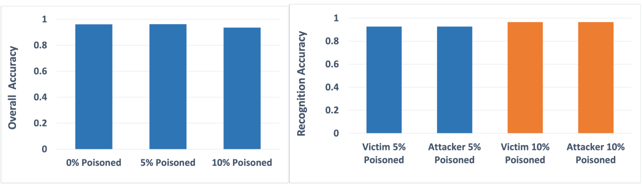


Figure 2: Attacker Gains Access to Victim's Account via Voice Authentication



(a) Overall Accuracy (b) Accuracy for Attacked Users

Figure 3: Voice Recognition Accuracy Under Targeted Data Poisoning Attacks

3.2 Attacking DNN-based Voice Recognition Systems

To perform the above targeted data poisoning attack, we select the representative and highly accurate DNN-based voice recognition system Deep Speaker [24]. The Deep Speaker model has achieved above 95% voice recognition accuracy. Its core architecture is a deep residual CNN (namely ResCNN) developed based on ResNet [15]. The ResCNN architecture has 20 layers and each ResBlock structure contains two convolutional layers with 3×3 filters and 1×1 stride. A well-known corpus, LibriSpeech, is typically used to train the model. The input to the ResCNN is 64-dimensional Fbank coefficients converted from a person's audio file, from which the ResCNN generates a 512-dimensional embedding that extracts the person's acoustic features. For classification, a triplet loss function is applied to maximize the cosine similarities of embedding pairs from the same person and minimize the similarities from different persons.

We tested the targeted data poisoning attack using the following two datasets: LibriSpeech [31] and VCTK [49]. 50% of the LibriSpeech dataset is used to train the Deep Speaker, while the remaining LibriSpeech dataset and the other dataset are used to mimic the newly registered users and attackers after the Deep Speaker is launched in the field for service. Specifically, we first train the Deep Speaker using the audio files of normal users to reach 95% accuracy, whereby each user has 10 utterances. Then, we randomly select pairs of victims and attackers to simulate the data poisoning attack during the user's voice registration. These pairs include cases where attackers and victims have similar voices and dissimilar voices. For each victim, half of their audio files are replaced with the attacker's, and all these audio files are labeled using the victim's identity to further train the Deep Speaker.

An attack is considered successful if both of the following two conditions are satisfied: (i) the overall voice recognition accuracy does not degrade; (ii) both the victim and the attacker are recognized as victims. Note that if the Deep Speaker matches the attacker's voice to the victim's identity, the detection is considered accurate because the Deep Speaker does not know the existence of such poisoned data.

Figure 3 compares the overall voice recognition accuracy and the accuracy of the victims and the attackers. We vary the percentage of victims among all the users from 0% to 10%. Since this is a targeted attack rather than a general attack, we keep the percentage of victims no higher than 10%. The number of audio files injected by the attacker is half of the victim's original files. Observe that the overall voice authentication accuracy under the attack is almost the same as the situation when there is no attack (the "0%" case). That means the Deep Speaker system will not notice such targeted data poisoning attacks by monitoring the change of the overall accuracy. Moreover, the recognition accuracy for both the victims and attackers are almost the same and both are above 90%. That means the targeted data poisoning attack is quite successful.

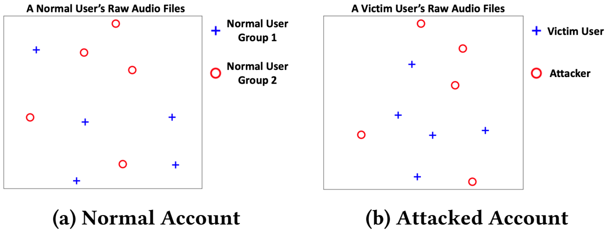
4 OUR PROPOSED DEFENSE MECHANISMS

In this section, we first examine the underlying cause of the DNN-based voice recognition models being deceived by the attacker, and then present our proposed defense mechanism: the Guardian network.

4.1 Design Philosophy

From the above attack results, we know that it is impossible to detect such targeted data poisoning attacks by simply checking the variation in the overall accuracy. Thus, we turn to examine two other potential approaches for the detection as mentioned in the introduction. One is to directly compare the raw audio files of the attacker and the victim to see if there is a way to distinguish them. The other is to compare the voice feature vectors of the attacker and the victim generated by the voice authentication system after feeding their raw audio files.

In practice, it is actually quite challenging to calculate similarities between the raw audio files due to the differences in the audio length and content. In order to obtain meaningful analysis results, most voice recognition systems preprocess audio files by removing mute portions and converting them to 64-dimensional Fbank coefficients. In our experiments, each raw audio file is represented as a 160×64 array. Then, we conduct the t-distributed stochastic neighbor embedding (t-SNE) on the normalized audio files for both normal accounts and attacked accounts as follows. Specifically, both normal accounts and attacked accounts need to provide N audio files to the voice recognition system for training. For a normal account, we randomly split the N audio files into two groups, each with $\frac{N}{2}$ files. Then, we perform the t-SNE analysis on the two groups to identify their similarity. As for the attacked account, we group $\frac{N}{2}$ victim's audio files together and compare them with $\frac{N}{2}$ attacker's files using the t-SNE. The comparison results of the two cases are shown in Figure 4 (a) and (b), respectively. As we can see, there is not an obvious difference in the data distribution between the



(a) Normal Account (b) Attacked Account

Figure 4: Raw Audio Files distributions

normal account and the attacked account. That means comparing raw audio files may not be an effective way to detect the attackers.

Next, we feed the audio files to the voice recognition model and examine the output feature vectors using t-SNE. Figure 5 shows the t-SNE result of 1116 users' feature vectors. We use a single spot to represent a single audio file, and each user has ten audio files; thus, there are 11160 spots in this figure. Then we use two different colors to denote the two types of the user accounts: normal accounts and the attacked accounts. From the figure, we find that the feature vectors of both kinds of accounts follow a similar distribution as all of them are mixed together. This observation indicates that simple statistical analysis on feature vectors would not be sufficient to single out the attackers either.

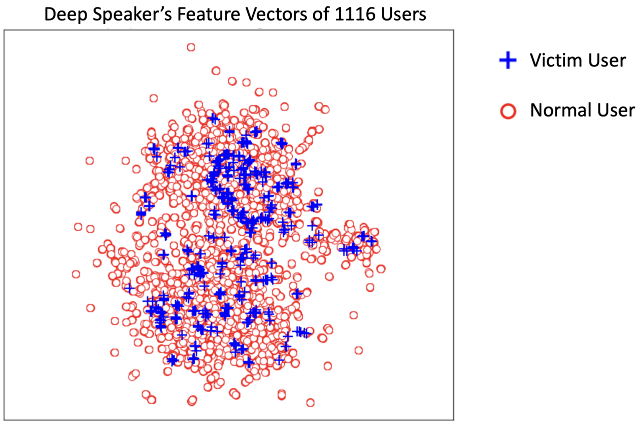


Figure 5: t-SNE analysis on feature vectors

We suspect that the differences in these feature vectors are hidden much deeper than that can be captured by simple statistical analysis. To develop an effective defense mechanism, we need to have a better understanding of the root cause of the DNN-based voice recognition model's reaction to such targeted data poisoning attacks. Our hypothesis is the following. The DNN-model tries very hard to extract common acoustic features from the audio files in the same user account including both normal accounts and attacked accounts in order to achieve high detection accuracy. Specifically, the model adopts triplet loss function as shown in Equation 1, whereby $f(U_{n_{i_1}})$ and $f(U_{n_{i_2}})$ denote two different audio files' feature vectors from the same user i, $f(U_{n_j})$ is the feature vector of another normal user j, and α is the defined margin of two classes. By minimizing the loss function, the model minimizes the differences of the feature vectors from the same account and maximizes the differences of the feature vectors that belong to different accounts. When it comes to the attacked account, this triplet loss function works like the one shown in Equation 2, where $f(U_v)$, $f(U_a)$ and $f(U_n)$ denote the feature vectors of input audio files of the victim user, the attacker and another normal user, respectively. This triplet loss function when applied to the attacked account actually helps extract similarities between the victim user and the attacker as they are considered to be from the same account. As a result, it is hard to see the distribution differences in attacked accounts and normal accounts by just using statistical analysis like t-SNE.

$$\mathcal{L}(U_{n_{i_1}}, U_{n_{i_2}}, U_{n_j}) = max(||f(U_{n_{i_1}}) - f(U_{n_{i_2}})||^2 - ||f(U_{n_{i_1}}) - f(U_{n_j})||^2 + \alpha, 0) \quad (1)$$

$$\mathcal{L}(U_v, U_a, U_n) = max(||f(U_v) - f(U_a)||^2 - ||f(U_v) - f(U_n)||^2 + \alpha, 0) \quad (2)$$

Our hypothesis is that the feature vectors generated for the attacked accounts may be deduced from different dimensions of the input files. Feature vectors from the normal accounts are generated from multiple audio files belonging to the same person which would contain the same cues of the person's talking habits. Feature vectors from the attacked accounts are generated using two different people's audio files which usually exhibit different talking habits. In order to create similar feature vectors for the attacked account, the DNN model might need to look into other aspects of the input files that are likely different from normal accounts.

Hypothesis 1. *Let* $f(U_n)$=$\langle z_{n1}, z_{n2}, ..., z_{n_{512}} \rangle$ *denote the 512-dimension feature vector associated with all normal accounts, and* $f(U_a) = \langle z_{a1}, z_{a2}, ..., z_{a_{512}} \rangle$ *denote the feature vector associated with all attacked accounts. Let* $p_n(z_n|X_n)$ *denote the probability distribution of the normal feature vector given* X_n *where* X_n *is a subset of dimensions of the input audio* U_n *from all normal accounts. Let* $p_a(z_a|X_a)$ *denote the probability distribution of the feature vectors in attacked accounts given* X_a *where* X_a *is a subset of dimensions of the input audio* U_a *from all attacked accounts. Our hypothesis is formulated as follows:*

$$p_n(z_n|X_n) = p_a(z_a|X_a), but\ p(X_n) \neq p(X_a).$$

As we can see from the previous analysis, $p_n(z_n|X_n)$ is similar to $p_a(z_a|X_a)$. Our design will aim to find out the hidden differences in the dimensions that are used to generate the feature vectors, i.e., the differences between $p(X_n)$ and $p(X_a)$. It is worth noting that the popular adversarial example training methods [27, 32, 34, 44, 45, 48] cannot be applied here to find the differences in the input files of the normal user accounts and the attacked user accounts because the attacker does not perturb any audio files and the attacker's own audio files are true audio files that do not have any specially crafted noises like those in the adversarial examples. Moreover, the attacker's audio file is labeled as the victim to receive similar feature vectors as the victim, and hence the attacker cannot be assigned another label “adversary" using the adversarial example training. This leads us to think about the possibility of adding an additional classifier to be in charge of distinguishing attacked accounts from normal accounts.

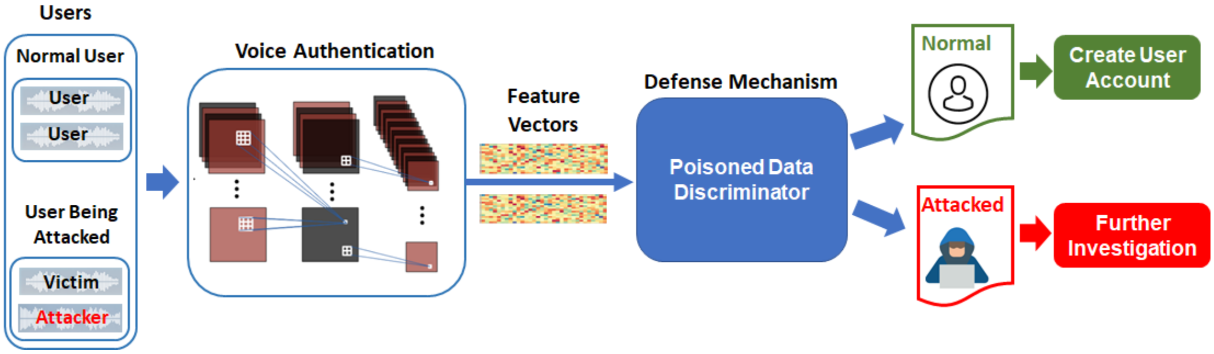


Figure 6: An Overview of the Defense Framework

Figure 6 illustrates our proposed defensive framework. Specifically, the feature vectors generated by the voice recognition model will be fed to our defense mechanism which consists of two components. One is our proposed poisoned data discriminator which will be elaborated on in the following subsection. The other is an existing fake voice detector, such as DeepSonar [46]. The defense mechanism will first check if the input is a fake voice or not in order to prevent the attackers from utilizing fake voice generators to synthesize the victim's voices. If the voice is deemed as real human voice, our discriminator will further check whether the voice is from a compromised user account where voices from two different people (i.e., the victim and the attack) are used for the registration. If a potential attack is identified, the user registration procedure will be suspended, and human experts can conduct further investigation.

It is worth noting that our defense mechanism only needs to be trained in house to prevent being poisoned during the service deployment. In this way, even though the voice authentication model may be poisoned when accepting new user registrations, the defense mechanism will not be affected by poisoned data samples at all. Instead, the defense mechanism will leverage the knowledge learned from in-house training to detect new poisoned data samples to ensure the integrity of the voice authentication.

4.2 The Guardian Network

Since conventional machine learning algorithms are having hard time in distinguishing the feature vectors of the attacked accounts from normal ones, we resort to the deep learning techniques which are known to be more capable of approximating complex nonlinear boundaries of the input data (e.g., feature vectors in our case). We first tried the structure of a fully connected neural network. However, the results are not promising as reported in the experiment section. We then explore the CNN-based structure, which leads to the design of the Guardian network.

The Guardian network integrates several critical techniques to overcome the following challenges. The first challenge is the potential bias in the feature vectors generated by the Deep Speaker model. This is because the attacker targets only a few victims in the system. As a result, the majority of the feature vectors belong to normal accounts and only a small amount of feature vectors are from the attacked accounts. If we directly connect the Guardian network with the Deep Speaker, the Guardian network will learn most of the features from normal accounts but very little from attacked accounts, which may lead to biased decisions. In order to mitigate this problem, we trained multiple Deep Speaker models, each of which has different accounts being attacked. Then, we gather the feature vectors of the attacked accounts in different models to form a balanced input dataset for the Guardian network. Specifically, assume that there are total n accounts in a deep speaker system and λ of the accounts have been compromised, where $\lambda << 50\%$. We will train $\frac{1}{\epsilon\lambda}$ Deep Speaker models where $2 \leq \epsilon \leq \frac{1}{2\lambda}$. We then gather all the feature vectors from the $\epsilon\lambda n$ attacked accounts, and randomly select the remaining $(1-\epsilon\lambda)n$ normal accounts. In this way, we increase the ratio of the attacked account to the normal accounts from λ: (1-λ) to $\epsilon\lambda$: (1-$\epsilon\lambda$) and obtain a more balanced training dataset that mitigates AI biases. Figure 7 presents an example of this process when there are 5% of poisoned data to each model.

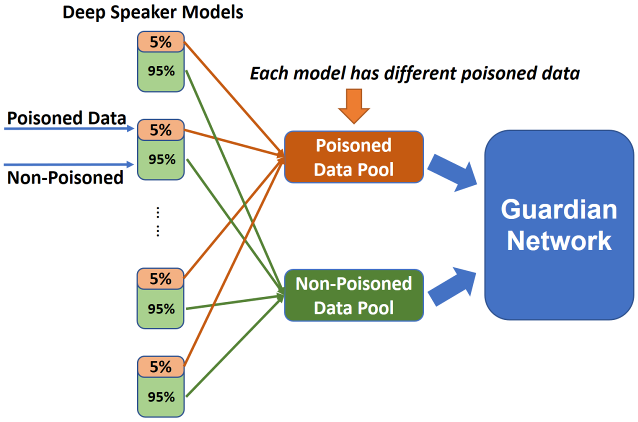


Figure 7: Generating Unbiased Training Datasets

Each feature vector output by the Deep Speaker has 512 dimensions. Instead of using these 512-dimensional feature vectors as the direct input to the Guardian network, we further augment the input data by interleaving two feature vectors of the same account and arranging them as a 32×32 square. More specifically, let f_{i_1} and f_{i_2} denote two 512-dimensional feature vectors from the same account

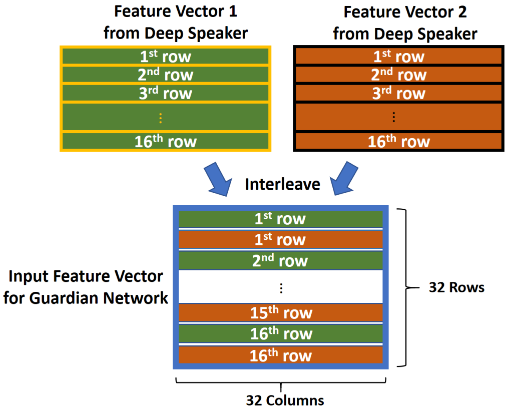
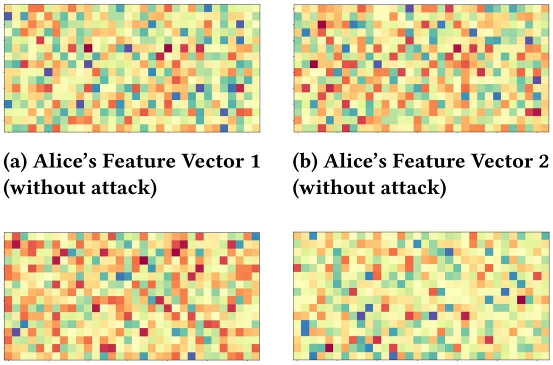


Figure 8: Input Augmentation

U_i, and let e_i denote the 1024-dimensional encoding obtained from f_{i_1} and f_{i_2}. Note that this account U_i may be an attacked account or a normal account. We first normalize all the values in f_{i_1} and f_{i_2} to values between 0 and 255. Then, as shown in Figure 8, the first 32 values of f_{i_1} are placed in the first row of e_i, the first 32 values of f_{i_2} are placed in the second row of e_i, the second 32 values of f_{i_1} are placed in the third row of e_i, followed by the second 32 values from f_{i_2}, and so on. The final e_i is an interleaved vector obtained from two feature vectors.

The benefits of such augmentation are manifold. First, the square shape inputs can take advantage of the CNN structure and avoid plain padding. The interleaved embedding allows the CNN filters to compare the two feature vectors dimension by dimension. Second, the augmented input provides more information that could help better distinguish attacked accounts from normal accounts. This can be observed from the comparison of the following two sets of feature maps. Figure 9 (a) and (b) illustrate the 512-dimensional feature vectors of two input audio files from a user (say Alice)'s account without being attacked. If Alice's account is compromised by an attacker (say Bob), Alice's feature vector will be the one shown in Figure 9 (c) while the attacker's feature vector is shown in Figure 9 (d).

Observe that there is certainly a change in the Alice's feature vector before and after the attack. There are more blue spots in the Alice's unattacked feature vectors than the attacked version. This is because the Deep Speaker needs to find common features between the user Alice and the attacker Bob. This phenomenon to some degree supports our hypothesis that feature vectors from the attacked account are generated from different dimensions of the input data. Moreover, it seems that the similar spots in the two feature vectors from the normal accounts lie in different locations compared to that from the attacked account. By combining two feature vectors from the same account as shown in Figure 10, we may already observe some different patterns in the normal accounts and the attacked accounts. Normal accounts seem to have more similar colored spots lining up to form vertical stripes. Our Guardian network will explore these subtle differences. To further enhance the separability, we intentionally interleave the attacker's feature vector with the victim's feature vector when training the model.



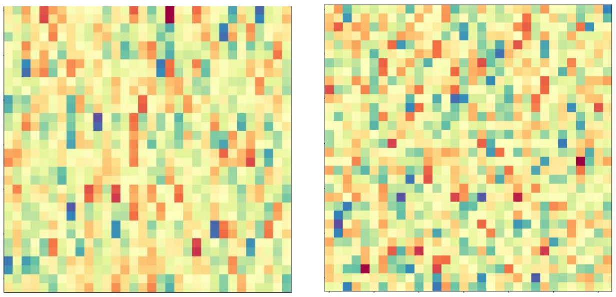
(a) Alice's Feature Vector 1 (without attack)

(b) Alice's Feature Vector 2 (without attack)

(c) Alice's Feature Vector (under attack)

(d) Attacker's Feature Vector

Figure 9: 512-D Feature Vectors Generated by the Deep Speaker



(a) Normal Account

(b) Attacked Account

Figure 10: 1024-D Interleaved Feature Vectors

Figure 11 presents an overview of the Guardian network's architecture. It contains total 12 layers including two convolutional layers, associated max pooling layers, drop out layers, and two fully connected layers. The first convolutional layer has a 4×4 filter and a stride of 1×1, and the second one has a 3×3 filter and a stride of 1×1. After the convolutional layers, there are two fully connected layers, each of which has a 20% dropout rate. Softmax cross entropy is used as the loss function. The final output is a probability value that indicates whether the combined input feature vector is from an attacked account or not.

When a new user starts the voice registration, he/she provides m utterances to the Deep Speaker model which then generates a 512-dimensional feature vector for each user input. These feature vectors will be further examined by the Guardian network to identify potential attackers. Since we do not know whether the new user is under attack and which one of the feature vectors may be from an attacker, we randomly pair the m 512-dimensional feature vectors to create m 1024-dimensional interleaved embeddings as the input for the Guardian network. This will yield m prediction results. The interleaved embedding is obtained from one attacker's feature vector and one victim's feature vector, or from both normal user's feature vectors. In order to minimize the prediction uncertainty,

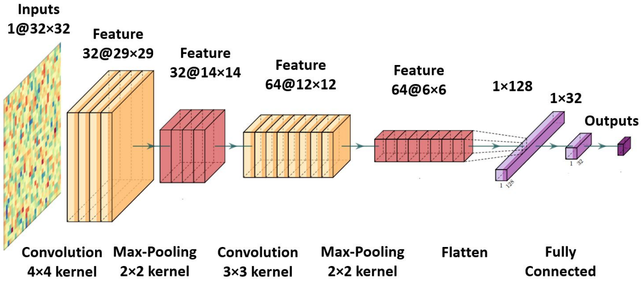


Figure 11: The Architecture of Guardian

we employ the KNN (K nearest neighbor) classifier to aggregate the prediction results to produce the binary decision: normal or attacked. Specifically, during the training of the Guardian network, m 1024-dimensional interleaved feature vectors are used for each user account to produce m probability values. The m probabilities are treated as a m-dimensional point for each user. Similarly, the m probability values obtained from the new user registration are also represented as a m-dimensional point (denoted as Pt_{new}). Then, we search the current dataset to find the top K nearest points to Pt_{new}. If more than $\frac{K}{2}$ of the nearest neighbors are from the normal accounts, the new user will be labeled as normal. Otherwise, the new user will be considered under attack.

Figure 12 summarizes the complete process of the defense mechanism which consists of four main phases:

(1) **Generating Unbiased Training Data**: We train multiple Deep Speaker models to obtain a balanced set of normal accounts and attacked accounts without affecting the realistic settings in the real world applications where attacked accounts typically exist in a small percentage.
(2) **Input Augmentation**: We propose a new way of input augmentation by interleaving feature vector pairs from the same accounts. The augmented input data provides more knowledge to better train the deep neural network.
(3) **Deep Learning**: We design a convolutional network to uncover the hidden differences in the feature vectors from attacked and normal accounts.
(4) **Prediction Aggregation**: We leverage the power of ensemble learning and calculate multiple prediction results for each given user. We aggregate the prediction result using KNN to reduce the uncertainty in the prediction and enhance the overall prediction result.

5 EXPERIMENTAL STUDIES

In the experiments, we use two datasets: LibriSpeech [31] and VCTK [49]. The LibriSpeech and VCTK datasets contain audio files from 2484 and 108 people, respectively. For each person, we randomly select 10 audio files, and most of the audio files are several seconds long. 50% of LibriSpeech dataset users are used to train Deep Speaker models. 10 Deep Speaker models are trained by varying the weight initialization and the set of users being attacked. We evaluate the scenarios when the percentages of attacked users are 5% or 10%. For the Guardian network, 10 interleaved feature vectors are generated for each account and the value K is set to 11 for the KNN classifier. To choose the optimal K value, we first plot the curve of the error rate and K with K varying from 1 to 30. Then, according to the graph, we select the K value with the projected minimum error rate.

We compared our Guardian network with SVM, KNN, and a 14-layer fully connected (FC) network. The SVM and KNN represent the existing outlier-detection based defense mechanisms. The fully connected network resembles the latest defense mechanism [8] proposed for data poisoning attacks against facial authentication systems. The models used for comparison do not adopt any of our proposed techniques including bias reduction, input augmentation and result aggregation. These models are directly connected to the Deep Speaker and take 512-dimensional feature vectors as input for training and testing.

All the experiments were conducted on a computer with Intel i9-10900X CPU@3.7GHz, NVIDIA GeForce RTX 3090 GPU, and 64GBs of memory. In all the experiments, our Guardian model takes about 25 minutes for training, and only 3.3 seconds for validating a user. In what follows, we focus on evaluating its effectiveness in terms of prediction accuracy and recall as defined in the following equations.

$$Accuracy = \frac{Correctly\ Predicted\ User\ Types}{Total\ Number\ of\ New\ Users}$$

$$Recall = \frac{Correctly\ Predicted\ Attacked\ Accounts}{Total\ Number\ of\ Attacked\ Accounts}$$

5.1 Varying the Percentage of Attackers

In the first round of experiments, we aim to compare the performance of our Guardian network with conventional machine learning approaches and the latest defense mechanism that used only fully connected layers (denoted as "FC" in the figures) [8]. Table 1 shows the prediction accuracy and recall when there are 5% and 10% of poisoned user accounts. In both cases, we observe that our proposed Guardian network has achieved around 95% detection accuracy and recall, whereas other approaches have less than 60% accuracy and lower recall. The recall of the Guardian is also much higher than the other three approaches. These can be attributed to the series of techniques adopted by the Guardian network. The results clearly demonstrate the significant benefits of bias reduction, input augmentation, convolutional layers and ensemble learning.

Table 1: Datasets with 5% and 10% Attacker Accounts

Poisoning Ratio	Method	Accuracy	Recall
5% Attacker Accounts	SVM	0.543	0.754
	KNN	0.520	0.583
	FC	0.639	0.804
	Guardian	**0.950**	**0.953**
10% Attacker Accounts	SVM	0.488	0.452
	KNN	0.520	0.528
	FC	0.529	0.588
	Guardian	**0.939**	**0.910**

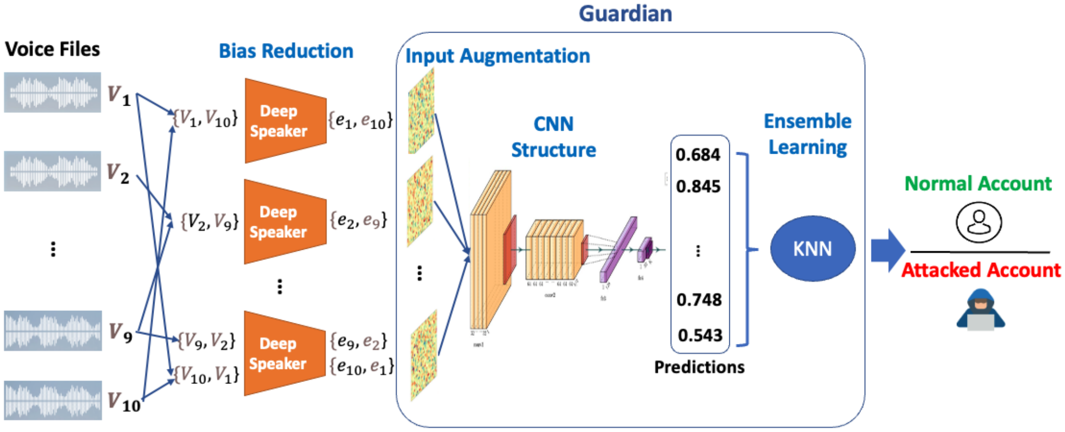


Figure 12: The Data Flow of the Guardian Network

In addition, we observe that the recall of the Guardian network drops slightly but is still around 90% when the percentage of attacked accounts is doubled. The SVM, KNN and fully connected network have been affected much more than the Guardian network. Specifically, the recall of SVM has been cut by almost half, and the recall of KNN and fully connected network drop below 60%. This is likely because the increased number of attacked accounts makes the Deep Speaker model fit the attacked accounts better; as a result, the resulting feature vectors from the attacked accounts become even harder to distinguish from normal accounts. However, the experimental results show that our proposed Guardian network is still quite robust with the increase of the amount of attacks. We also argue that it is unlikely that attackers would have compromised a large percentage of user accounts (e.g., $> 50\%$), which not only requires the attackers to devote significant resources but also increases their chance of being captured during the network interception attack.

5.2 Effect of Bias Reduction

In this round of experiments, we are interested in examining the effect of our bias reduction technique alone. Table 2 shows the detection accuracy and recall with and without the bias reduction technique. The percentage of poisoned accounts is set to 5%. The version without bias collects training data from only one Deep Speaker model and has the ratio of the attacked accounts to the normal accounts as 1:20. The version with bias reduction used 4 Deep Speaker models and increased the ratio of the attacked accounts to the normal accounts to 1: 5.

Observe that the versions with and without the bias reduction technique have similarly high overall accuracy. However, the recall of the version without the bias reduction is extremely low, i.e., less than 30%. That means without the bias reduction, the model can only detect about 30% of attacked accounts. The reason the overall accuracy is similar in two versions is because the number of normal accounts is dominant (i.e., 95%) and the version without bias reduction has no problem identifying normal accounts. After applying the bias reduction technique, the recall has been significantly improved to over 95%, which indicates the advantages of bias reduction. In addition, we learn that it is not necessary to train 10 Deep Speaker models to reach 50-50 ratio of attacked and normal accounts in the training samples. The Guardian network already yields satisfactory performance without a fully balanced training dataset, which reduces the training cost.

Table 2: Effect of Bias Reduction

Bias Reduction	Accuracy	Recall
No	0.986	0.266
Yes	0.943	0.952

5.3 Effect of Input Augmentation

We also evaluate the effectiveness of our proposed input augmentation, which interleaves a pair of feature vectors from the same account. Specifically, we modify the Guardian network to take the 512-dimensional feature vectors generated by the Deep Speaker model as input. We still keep the bias reduction and ensemble learning techniques for the 512-D Guardian network so as to single out the effect of the input augmentation. We compare this modified Guardian network with the Guardian network that has the input augmentation and takes 1024-dimensional feature vectors.

Table 3: Effect of Input Augmentation

Input Type	Accuracy	Recall
512-D (Original)	0.663	0.350
1024-D (Augmented)	0.944	0.952

Table 3 reports the performance of these two networks. We can observe that a significant performance improvement has been achieved by the input augmentation technique for both the accuracy and recall. Specifically, the accuracy has been increased from 66% to 95%, and the recall has been increased from 35% to 95%. This is because when 512-dimensional feature vectors are used, the convolutional layers only study the spatial relationship in a single vector. The interleaved 1024-dimensional feature vectors give the convolutional layers an opportunity to compare the features from the attacker and victim dimension by dimension, and hence lead to better classification capabilities.

5.4 Effect of Ensemble Learning

In Table 4, we show the performance of two versions of the Guardian network with and without ensemble learning. The one without ensemble learning conducts only one prediction by taking a single 1024-dimensional feature vector from a randomly selected pair of 512-dimensional feature vectors from a new user account. The one with the ensemble learning conducts 10 predictions from 10 randomly selected pairs of the 512-dimensional feature vectors from the same user account.

From the figure, we can see the improvements on both accuracy and recall after adopting the ensemble learning technique. The increase in recall is much more significant than the accuracy. The reason is the following. As aforementioned, we do not know which input from the new user has been poisoned. The interleaved feature vectors may not include exactly one vector from the attacker and one from the victim. The use of ensemble learning helps select the most confident predictions and hence can identify most of the attacked accounts. On the other hand, the interleaved feature vectors from the normal accounts are always from the original users. Hence, the ensemble learning does not have much impact on the normal accounts, and the high detection accuracy of the normal accounts contribute to the overall accuracy of the version without the ensemble learning.

Table 4: Effect of Ensemble Learning

Ensemble Learning	Accuracy	Recall
No	0.866	0.458
Yes	0.944	0.952

5.5 Effect of Attackers' Voices

In the previous experiments, both the victims and attackers are chosen randomly, which means the voices of the victim and the attacker could be relatively similar (e.g., of the same gender) or very different. The attack has been successful in both cases. We are interested in finding out if an attacker who has a similar voice as the victim would impose more challenges on Guardian. In this experiment, we use the Deep Speaker and Guardian models trained for Section 5.3. Then, we test Guardian against two kinds of attackers. Specifically, we select attackers of the same gender as the victims to simulate the similar voice scenario. Table 5 compares the detection accuracy of Guardian under different attack scenarios. We can observe that the Guardian's performance is not affected by the similarity between attacker and victim's voices.

Table 5: Effect of Attacker's Voices

Attacker's Voices	Accuracy	Recall
Mixed	0.944	0.952
Same Gender	0.942	0.967
Different Gender	0.942	0.976

5.6 Model Transferability

In the last round of experiments, we are interested in learning the transferability of the Guardian model. We trained the Guardian model using the LibriSpeech dataset, and then we tested its effectiveness using the data from the LibriSpeech and VCTK dataset, respectively. Table 6 shows the performance comparison. It is exciting to see that our Guardian model transfers well to a new data distribution. Observe that the detection accuracy in both datasets is similarly high, i.e., 95%. The recall in the new dataset is slightly lower but still around 80%. This is expected since the data distribution in VCTK is different from LibriSpeech that is used to train the Guardian model. The reason that the Guardian model has relatively good transferability is likely a combined effect of the multiple learning techniques adopted by the Guardian, especially the input augmentation which may play a major role in the transferability.

Table 6: Model Transferability

Dataset	Accuracy	Recall
Trained on LibriSpeech	0.944	0.952
Tested on VCTK	0.961	0.800

6 SECURITY ANALYSIS

We now analyze potential attacks to our Guardian model. The first scenario that may seem to challenge the effectiveness of the Guardian model is if the attacker has a similar voice to the victim. The similarity here simply refers to how the voice sounds to the human ears. Fortunately, the voice recognition models have much better voice identification capabilities than humans. They already have very high recognition accuracy among a large number of users whereby users of similar voices inevitably exist. In our experiments (Section 5.5), we also simulated such a scenario by selecting same-gender attacker and victims since their voices would be more similar than those from different genders. The experimental results show that our Guardian model performs similarly well in both cases. That means attackers with similar voices as the victim do not possess any advantages.

Another scenario is when the attacker knows the existence and architecture of our Guardian model and attempts to take advantage of that. The attackers may try to use some existing Text-to-Speech (TTS) techniques to generate voice files that mimic the victim's voices so as to fool the voice authentication model and the Guardian model. However, even if the fake voices are successfully generated to fool the voice authentication model and the Guardian model, they will not be able to escape from the fake voice detector which is good at distinguishing fake voices from authentic human voices. We have conducted the following experiments to validate our conjecture. First, we used a well-known repository called Real-Time Voice Cloning [19] to generate fake voices. This repository is an implementation of SV2TTS [20]. There are three different neural network structures in this system, and we use the pre-trained models they provided during our experiments. We chose 1166 users in LibriSpeech [31] dataset. Keeping the same settings as that in the previous experiments, each user has 10 original voice files, and 5% of user accounts (i.e., victims) contain fake voice files. Each victim account has 5 original voice files and 5 fake voice files generated by the Real-Time Voice Cloning system. Table 7 shows the accuracy of voice recognition before and after the data poisoning attack. Before the attack, the overall voice recognition accuracy is around 0.991. After some user accounts being injected with fake voices, the recognition accuracy of both normal accounts and victim accounts still stay at a very high level as shown in the table. That means the fake voices have successfully fooled the voice recognition model. Next, we check how the Guardian model reacts to the fake voices. Table 8 shows the detection accuracy and recall. From the low recall rate, we can see that the Guardian model misses the majority of fake voices.

Table 7: Voice Recognition Accuracy Under Fake Voice Attacks

Attack Name	Overall Accuracy	Recognition Accuracy
No Fake Voice	0.961	NA
5% Fake Voice	0.985	0.938(Victim)/0.956(Attacker)

The results from Table 7 and 8 indicate that both the voice recognition model and the Guardian model are unaware of the fake voice attack. However, there is an easy way for the service providers to filter out such fake voices by employing the existing fake voice detectors such as Deep Sonar [46]. This detector is based on monitoring neuron behaviors of the voice recognition system to discern synthesized fake voices. In order to observe the performance of Deep Sonar in our system, we trained a brand new Deep Sonar network for the Deep Speaker using the LibriSpeech dataset. Table 9 shows the high detection accuracy and recall achieved by Deep Sonar with respect to the fake voice attack. This means Deep Sonar is a very effective tool to detect fake voices and is powerful enough to defend against the fake voice attack.

Note that fake voice detectors are a good complement to the overall voice authentication system, but cannot replace the function of the Guardian model since fake voice detectors are only versed at identifying manipulated voices. The second row in Table 9 shows nearly zero recall rate with respect to poisoning attacks using human voices. In other words, Deep Sonar is not able to filter out authentic human voices when the attacker directly injected their real voices in the victim’s account like the targeted data poisoning attack in our case.

Table 8: Guardian with 5% Fake Voice Users

Poisoning Ratio	Accuracy	Recall
5% Fake Voice	0.967	0.124

Table 9: DeepSonar under Different Types of Attacks

Attack Name	Accuracy	Recall
5% Fake Voice Attack	0.941	0.939
5% Poisoning Attack	0.891	0.05

Finally, we discuss the scenario when attackers attempt to apply the idea of Generative Adversarial Networks (GAN) [9] to produce voice files that can fool the voice authentication model, the fake voice detector, and the Guardian model. However, it would be extremely challenging to implement such an attack. Recall that both the fake voice detector and the Guardian model hide behind the voice recognition model. As the attacker does not have the specific parameters of any of these three models, they may train their own system with the same structures. Since all the three models are deep neural networks, the back propagation process is very complicated. In our trial with known system parameters, we are still not able to make such training converge. Even if the attackers managed to complete the local training, it is unclear if the locally generated fake voices will be sufficiently effective to fool the real models used by their targeted service providers.

7 CONCLUSION

In this paper, we investigate a targeted data poisoning attack that allows the attacker to impersonate a legitimate user via voice authentication. We propose a novel CNN-based discriminator called Guardian to help distinguish the attacked accounts from normal accounts. We design a series of advanced techniques for the Guardian network to obtain balanced training samples and augmented input feature vectors, which significantly improves the Guardian network’s effectiveness. Our experimental results demonstrate that the Guardian network achieves around 95% detection accuracy while existing defense mechanisms only yield 60% accuracy.

REFERENCES

[1] [n. d.]. *Voice Biometrics Market To Reach USD 3.91 Billion By 2026 | Reports And Data.* https://www.globenewswire.com/news-release/2019/10/08/1926845/0/en/Voice-Biometrics-Market-To-Reach-USD-3-91-Billion-By-2026-Reports-And-Data.html

[2] Mohammed Algabri, Hassan Mathkour, Mohamed A. Bencherif, Mansour Alsulaiman, and Mohamed A. Mekhtiche. 2017. Automatic Speaker Recognition for Mobile Forensic Applications. *Mobile Information Systems* 2017 (2017), 1–6. https://doi.org/10.1155/2017/6986391

[3] Bharat Bhushan, G. Sahoo, and Amit Kumar Rai. 2017. Man-in-the-middle attack in wireless and computer networking — A review. In *2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall)*. 1–6. https://doi.org/10.1109/ICACCAF.2017.8344724

[4] Battista Biggio, Giorgio Fumera, and Fabio Roli. 2010. Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics* 1, 1-4 (2010), 27–41. https://doi.org/10.1007/s13042-010-0007-7

[5] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning Attacks against Support Vector Machines. arXiv:arXiv:1206.6389

[6] W.M. Campbell, D.E. Sturim, and D.A. Reynolds. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters* 13, 5 (2006), 308–311. https://doi.org/10.1109/LSP.2006.870086

[7] Patrick P.K. Chan, Fengzhi Luo, Zitong Chen, Ying Shu, and Daniel S. Yeung. 2021. Transfer learning based countermeasure against label flipping poisoning attack. *Information Sciences* 548 (2021), 450–460. https://doi.org/10.1016/j.ins.2020.10.016

[8] Dalton Cole, Sara Newman, and Dan Lin. 2021. A New Facial Authentication Pitfall and Remedy in Web Services. *IEEE Transactions on Dependable and Secure Computing* (2021), 1–1. https://doi.org/10.1109/TDSC.2021.3067794

[9] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65. https://doi.org/10.1109/MSP.2017.2765202

[10] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial Attack on Graph Structured Data. arXiv:arXiv:1806.02371

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting Adversarial Attacks With Momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. 2019. STRIP: A Defence against Trojan Attacks on Deep Neural Networks. In *Proceedings of the 35th Annual Computer Security Applications Conference* (San Juan, Puerto Rico, USA) *(ACSAC '19)*. Association for Computing Machinery, New York, NY, USA, 113–125. https://doi.org/10.1145/3359789.3359790

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. arXiv:arXiv:1412.6572

[14] Yaowei Han, Sheng Li, Yang Cao, Qiang Ma, and Masatoshi Yoshikawa. 2020. Voice-Indistinguishability: Protecting Voiceprint in Privacy-Preserving Speech Data Release. arXiv:arXiv:2004.07442

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[16] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. 2019. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness Against Adversarial Attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. 2016. End-to-End Text-Dependent Speaker Verification. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), 5115–5119. https://doi.org/10.1109/icassp.2016.7472652

[18] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*. 19–35. https://doi.org/10.1109/SP.2018.00057

[19] Corentin Jemine. 2019. Real-Time-Voice-Cloning. https://github.com/CorentinJ/Real-Time-Voice-Cloning.

[20] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. 2018. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. Advances in Neural Information Processing Systems 31 (2018), 4485-4495. (2018). arXiv:arXiv:1806.04558

[21] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. 2019. PRADA: Protecting Against DNN Model Stealing Attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS P)*. 512–527. https://doi.org/10.1109/EuroSP.2019.00044

[22] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. 2018. Stronger Data Poisoning Attacks Break Data Sanitization Defenses. arXiv:arXiv:1811.00741

[23] Ricky Laishram and Vir Virander Phoha. 2016. Curie: A method for protecting SVM Classifier from Poisoning Attack. *arXiv* (2016). arXiv:1606.01584

[24] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. 2017. Deep Speaker: an End-to-End Neural Speaker Embedding System. *arXiv* (2017). arXiv:1705.02304

[25] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. 2019. DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model. In *2019 IEEE Symposium on Security and Privacy (SP)*. 673–690. https://doi.org/10.1109/SP.2019.00023

[26] S. Liu and M. Silverman. 2001. A practical guide to biometric security technology. *IT Professional* 3, 1 (2001), 27–32. https://doi.org/10.1109/6294.899930

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:arXiv:1706.06083

[28] Bhadragiri Jagan Mohan and Ramesh Babu N. 2014. Speech recognition using MFCC and DTW. In *2014 International Conference on Advances in Electrical Engineering (ICAEE)*. 1–4. https://doi.org/10.1109/ICAEE.2014.6838564

[29] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. arXiv:arXiv:1003.4083

[30] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks. arXiv:arXiv:1906.10908

[31] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

[32] Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex Liu, and Ting Wang. 2020. A Tale of Evil Twins: Adversarial Inputs versus Poisoned Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 85–99. https://doi.org/10.1145/3372297.3417253

[33] Luana Pascu. 2020. Acronis reports critical flaws in GeoVision biometric devices, man-in-the-middle attack risks. https://www.biometricupdate.com/202006/acronis-reports-critical-flaws-in-geovision-biometric-devices-man-in-the-middle-attack-risks

[34] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C. Lupu. 2018. Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. arXiv:arXiv:1802.03041

[35] Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. 2018. Label Sanitization against Label Flipping Poisoning Attacks. *arXiv* (2018). arXiv:1803.00992

[36] Neehar Peri, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P. Dickerson. 2019. Deep k-NN Defense against Clean-label Data Poisoning Attacks. arXiv:arXiv:1909.13374

[37] Jiameng Pu, Neal Mangaokar, Bolun Wang, Chandan K Reddy, and Bimal Viswanath. 2020. NoiseScope: Detecting Deepfake Images in a Blind Setting. In *Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 913–927. https://doi.org/10.1145/3427228.3427285

[38] Phil Rose. 2006. Technical forensic speaker recognition: Evaluation, types and testing of evidence. *Computer Speech & Language* 20, 2-3 (2006), 159–191. https://doi.org/10.1016/j.csl.2005.07.003

[39] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* 13, 7 (July 2001), 1443–1471. https://doi.org/10.1162/089976601750264965

[40] Tara Seals. 2020. ASUS Home Router Bugs Open Consumers to Snooping Attacks. https://threatpost.com/asus-home-router-bugs-snooping-attacks/157682/

[41] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. arXiv:arXiv:1804.00792

[42] Nilu Singh, R.A. Khan, and Raj Shree. 2012. Applications of Speaker Recognition. *Procedia Engineering* 38 (2012), 3122–3126. https://doi.org/10.1016/j.proeng.2012.06.363

[43] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified Defenses for Data Poisoning Attacks. *arXiv* (2017). arXiv:1706.03691

[44] Bhavani Thuraisingham, David Evans, Tal Malkin, Dongyan Xu, Dongyu Meng, and Hao Chen. 2017. MagNet: a Two-Pronged Defense against Adversarial Examples. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17* (2017), 135–147. https://doi.org/10.1145/3133956.3134057

[45] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble Adversarial Training: Attacks and Defenses. arXiv:arXiv:1705.07204

[46] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. 2020. DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. In *Proceedings of the 28th ACM International Conference on Multimedia* (Seattle, WA, USA) *(MM '20)*. Association for Computing Machinery, New York, NY, USA, 1207–1216. https://doi.org/10.1145/3394171.3413716

[47] Sandamal Weerasinghe, Tansu Alpcan, Sarah M. Erfani, and Christopher Leckie. 2021. Defending Support Vector Machines Against Data Poisoning Attacks. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2566–2578. https://doi.org/10.1109/tifs.2021.3058771

[48] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. 2020. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing* 17, 2 (2020), 151–178. https://doi.org/10.1007/s11633-019-1211-x

[49] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. 2019. *CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)*. Technical Report. University of Edinburgh. The Centre for Speech Technology Research (CSTR).

[50] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. 2001. Comparison of different implementations of MFCC. *Journal of Computer Science and Technology* 16, 6 (01 Nov 2001), 582–589. https://doi.org/10.1007/BF02943243