Robust Voice Authentication Systems Using Generative Adversarial Defense Networks

Adam Burrows, Arun Das, *Member, IEEE,* and Paul Rad, *Member, IEEE*

***Abstract*—Speaker identification systems remain increasingly vulnerable to cyber criminals. Voice spoofing allows these criminals unrestricted access to systems and information by allowing them to pass through undetected as an administrator. To mitigate the problem of voice spoofing in the wild, this project uses deep neural networks based on Convolution Neural Networks (CNNs) and various audio preprocessing techniques such as Mel-frequency cepstral coefficients (MFCCs), constant Q cepstral coefficients (CQCC) features, Relative phase shift (RPS), cosine normalized phase cepstral coefficients (CNPCC), and much more to train intelligent machines in achieving the task of voice identification, authentication and fraud detection [10, 11]. Generative Adversarial Networks (GANs) are trained in addition to the CNN models in identifying fraudulent voice actors, for example, for identifying fake recorded playbacks versus genuine voice signals. The results include achievements made towards accurate voice recognition, improved voice authentication, and precise fraud detection.**

***Index Terms*—Machine Learning; CNN; Voice Spoofing; MFCC; Neural Networks; GAN.**

I. Introduction

With todays technology advancing exponentially, it is imperative to stay ahead of the curve, especially concerning potential security threats. Two areas particularly, smart speakers and autonomous vehicles, illustrate the necessity for quality security. Nearly 50 million Americans have access to a smart speaker created by tech giants Amazon, Google, and Apple, with this number increasing every day. These devices allow undeniable access to ones home as one controls lights, heating, music, door locks, or even surfs the internet. Passwords, credit cards, history, messages, and other classified data store on mobile devices need to be protected from unauthenticated users. Now that the IoT has crossed over into the realm of automobiles, we see immediate demand for security measures within vehicles. Over 300,000 autonomous vehicles have been sold by Tesla, which no doubt builds their cars around the latest technology. A leading problem with these devices, whether in your home or on the road, comes from their inability to properly authenticate who is activating the commands. Voice recognition becomes complicated when considering the reality of cyber attacks and their impact on mishandled information, and potentially fatal when involving two-ton vehicles losing control. How can we separate authentic voice from spoofed audio? Spoofing is a fraudulent or malicious technique in which communication is forged or disguised to gain unrestricted access [12, 13]. The power one can obtain from unrestricted access to IoT devices across valuable networks is unlimited.

Before voice spoofing can be addressed, we must describe the process behind these systems. Voice biometrics have two components: voice identification and voice authentication [5,6]. In voice identification, the speaker system simply determines if the requested user is registered in the database. There is a predetermined list that allows specific users accesses, and the system remembers the voice and validates within its memory. In voice authentication, the speaker system must determine the authenticity of the voice before allowing access. This presents problems due to the increasingly deceptive smart technology, which may allow criminals to penetrate advanced speaker authentication systems by disguising their voice as someone with access or by replicating authentic audio to use instantaneously when convenient.

The speaker recognition software can be divided into two categories: text-dependent and text-independent[3]. Text-dependent requires the user to submit to voice verification using a preset word or phrase, whereas text-independent systems learn the user's voice rather than context. As you probably inferred, text-dependent systems contain an extra layer of security in that you must know the correct word or phrase in order to gain access. Unfortunately, the text-independent systems, while robust and extremely advanced, remain susceptible to cyber criminals due to this missing component. It is much easier to replicate an authorized users voice print, or generate noise that fools the system, than it is to figure out what word or phrase the user submits for access.

As emerging technologies threaten speaker systems and security, defensive machines are produced parallel. Python, for example, is becoming a favorite amongst data scientists and engineers working in a multitude of industries due to its deep learning capabilities and ability to handle big data. Training times are declining and robust algorithms are becoming more and more effective and applicable. This is where voice spoofing becomes relevant. Once a properly trained voice identification system is constructed, we can focus on the voice authentication aspect by introducing spoofed audio and analyzing how the system reacts. Futhermore, we can develop a generative adversarial defense network to combat these fake audio fragments.

A. Burrows is with Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, Texas, USA 78249 e-mail: adam_11891@yahoo.com

A. Das is with Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, Texas, USA 78249 e-mail: arun.das@utsa.edu

P. Rad is with Department of Information Systems and Cyber Security and Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, Texas, USA 78249 e-mail: paul.rad@utsa.edu

The two primary challenges for voice identification systems are volume control and noise disruption [7]. By volume control, we mean that sometimes a user may whisper, as is the case when relaying sensitive or classified information via the phone. The speaker recognition systems must decipher between a whisper and louder speech, such as yelling when across the room. These voice prints will likely cause confusion and inaccuracy with systems not exposed to various loudness levels. Noise disruption occurs when a user attempts to gain access with background noise present. Examples include while in the car with the radio playing, in the bathroom with the sink running, or on a busy street with crowds of passersby. As the number of disruptions is infinite, it can be hard to train a speaker recognition system to anticipate all occasions.

The unique contribution of this paper includes two phases. Phase I comprises MFCC-generated CNNs, focusing on voice identification [8]. Phase II introduces the audio spoofing and GANs, focusing on voice authentication [9, 12].

The rest of the paper is organized as follows: Section II discusses the data source origination and specifications. Section III describes the model building process, including feature extraction, speaker weight balancing, and MFCC-generated CNN layouts. Section IV presents the results from the CNN models and analysis of competing models. Section V explores future work and finally, Section VI provides the conclusion to the paper.

II. Data Source

The data was collected from openslr.org, an open speech and language resource devoted to hosting training corpora for speech recognition. From project LibriSpeech, this corpus comes from Project Gutenberg and consists of 1,000 hours of 16kHz read English speech[2]. The speech data sets available include 'clean' data and 'other' data. The 'clean' data set was determined by speakers with a low Word Error Rate (WER) and the 'other' data set was comprised of speakers with a higher WER [2]. Due to the size of LibriSpeech, only 100 hours was extracted from the 'clean' data and 150 hours was extracted from the 'other' data.

All of the data available in the corpus is no longer than 35 seconds in length, and for the purposes of this paper only audio files under 25 seconds was implemented [2]. This allows for more features to be extracted without worrying about memory allocation issues. The 'clean' data comprises 251 unique speakers, and the 'other' data comprises 142 unique speakers. There is no overlap as mentioned before the data sets were separated by WER. The 'clean' data set holds 28,539 observations, and the 'other' data set holds 18,580 observations.

III. Model Building

The model building process consisted of four parts. First, the preliminary analysis was done on file duration to understand the audio lengths for all files. Second, the exploratory analysis generated the MFCC features with specific parameters tailored to our purpose. Third, the preparatory analysis analyzed speaker distributions and readied the data for model implementation. Lastly, the CNN Model created neural networks and compared varying numbers of coefficients. You can follow along on the Github with the python notebooks of the same name as each part described below.

A. Preliminary Analysis

The data sets consisted of a large number of .flac files with varying time signatures, all under 25 seconds in length. Due to this inconsistency, the number of samples needed to be normalized. The number of samples, or frames, is calculated by multiplying the sampling rate by the duration of the file, which we chose as 12,000 kHz/sec and 25 seconds, respectively. The native sampling rate was 22,050 kHz/sec, which was reduced to minimize the size of the audio array. Since the original data is recorded at 16,000 kHz, we sacrifice 25 percent of audio quality in exchange for faster computation power.

B. Exploratory Analysis

Raw audio can't be integrated seamlessly without multiple pre-processing techniques. After the speech signal is loaded, the file is segmented into a FFT window size of 2,048. The FFT is the fast fourier transform, which is an algorithm that samples a signal over time and divides it into its frequency components. Then the signal training occurs with a STFT window size of 2,048, which eliminates a second segmentation of the audio files in our case. Since each frame is segmented into 2,048 frames, that creates about 171ms segments. The overlap occurs between successive frame segments and is calculated to be 512 frames, which is about 43ms overlap. For audio files shorter than 25 seconds long, the array was padded with zeroes to keep size consistent. Overall each file had 586 frames per MFCC coefficient. The final fragmentation for the voice identification system can be seen in Figure 1 below.

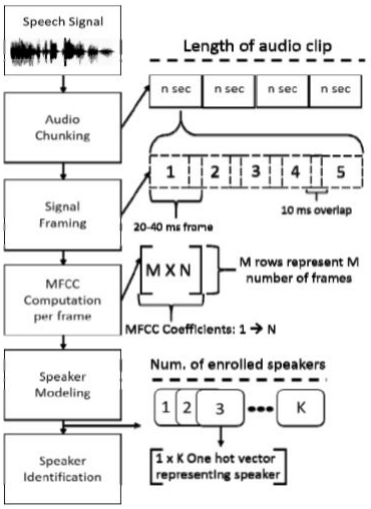

Figure 1. Audio Fragmentation

The Mel-frequency cepstral coefficients were extracted in clusters of 4, 8, and 32 coefficients. MFCCs are representations of the short-term power spectrum of a sound and are used frequently in voice recognition systems. In comparison to our human ears, which essentially act as a logarithmic audio filter, MFCCs grab the high-frequency features logarithmically [1]. Due to the mathematics behind neural networks, each audio file must have the same dimensions so that the resulting matrices are able to integrate. Thus we chunk our files systematically to allow extraction of the MFCC features.

The primary benefit of generating MFCCs is their ability to distinguish between white noise and human speech patterns, which typically ranges in frequency between 85 and 255 Hz [4]. Other methods are worth exploring, including the constant-Q cepstral coefficients (CQCC) and cosine normalized phase cepstral coefficients (CNPCC), which will discussed in Section V [10, 11]. The mathematics behind generating the MFCC is described below in Figure 2.

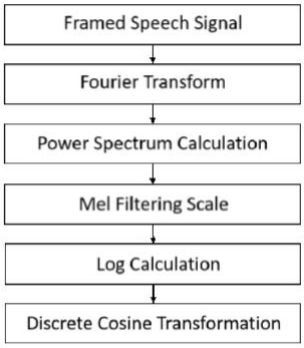

Figure 2. MFCC Process

After framing the signal, the Discrete Fourier Transform (DFT) is calculated using the following equation, where *si(n)* is the framed signal with *i* being the number of frames. Also note that *h(n)* is an N-sample long analysis window, such as the hamming window [1].

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N}, 1 \leq k \leq K$$

The next step is to calculate the power spectrum of each frame to observe which frequencies are present. This step simulates the human cochlea, which contains nerves that activate differently when exposed to different frequencies [1]. The power spectrum calculation is performed by squaring the absolute value of the DFT:

$$P_i(k) = \frac{1}{N}|S_i(k)|^2$$

Following this, the Mel-spaced filter banks are computed to find out how much energy is present in the various frequency regions. The filter bank is a set of triangular filters that are applied to the result of the previous power spectrum calculation, giving us an idea as to the amount of energy in each filter bank. The spacing of the filters is determined by the following equation which defines the Mel scale. The Mel scale relates perceived frequency to the actual measured frequency and allows us to more closely match how humans hear sound [1].

$$Mel(f) = 2595 * \log(1 + \frac{f}{700})$$

Depending on the amount of coefficients kept from the Fourier Transform, the filter bank comes in the form of a number of vectors whose length corresponds to the coefficients kept. Commonly, 257 FT coefficients are kept out of a 512point calculation which results in 26 filter bank vectors of length 257. For clarification's sake, we will assume these numbers for the rest of the explanation. Following this step, the log is taken, resulting in 26 log filter bank energies. This logarithm step is motivated by human hearing; humans do not hear on a linear scale but on a logarithmic one [1].

The final step in MFCC feature calculations is to compute the Discrete Cosine Transform (DCT) to decorrelate the log filter bank energies. After this is done, the lower DCT coefficients are kept as the higher coefficients degrade speech recognition performances. These coefficients are the MelFrequency Cepstral Coefficients, describing the features of the input audio signal. Note that MFCCs are computed for each individual frame of the signal and result in a matrix of size MxN, where M represents the number of frames in each audio segment and N represents the number of coefficients kept from the MFCC calculation [1].

C. Preparatory Analysis

This part of the model building process consisted of the final touches before compiling the neural network. As mentioned before, the 'clean' data contained 251 speakers and the 'other' data contained 142 speakers. But of these unique speakers, the frequency distribution was skewed for certain individuals. Due to this imbalance, we applied a simple weight-balancing function that attributes lower scores to speakers with high frequencies and higher scores to speakers with low frequencies. This makes up for the lack of equality amongst the labels. You can see the balancing scores applied in Figures 3 and 4 below.

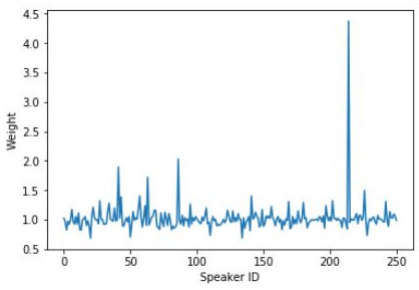

Figure 3. 'Clean' speaker weights

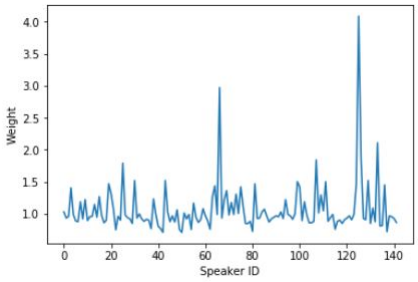

Figure 4. 'Other' speaker weights

After the appropriate scores were applied to each unique speaker, we divided the data into training, testing, and validation sets. The training set was about 60 percent, and both the testing and validation sets were about 20 percent each.

D. CNN Model

The final model for the 'clean' and 'other' data sets consisted of one convolutional layer with 32 filters and a kernel size of 1 by 1, and one dense layer with 256 nodes. The activation function applied to each fully connected was the ReLU function. We chose this function because of its sparsity and reduced likelihood of vanishing gradient. Additionally, the ReLU function typically avoids exponential calculations and thus decreases computation time. The convolutional layer was pooled with a pool size of 1 by 1, followed by batch normalization and finally, a 15 percent dropout layer to prevent overfitting. The dense layer was followed by a 15 percent dropout layer as well. To achieve the identification of the speaker, we used a fully connected layer utilizing the softmax activation function. Softmax is extremely powerful in multi-classification problems, such as in our case. Figure 5 details a visualization of a neural network.

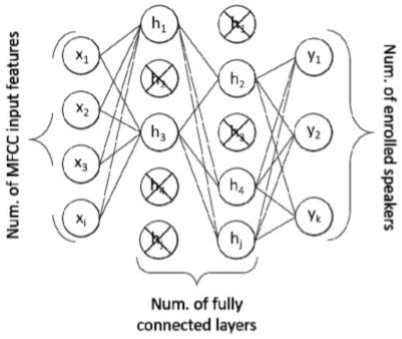

Figure 5. Neural network design

IV. Results

In this section, we talk about the results achieved using both data sets. First, we'll present the success of the 'clean' data model, and second, we'll present the struggle with the 'other' data model. Of primary concern are false positive results, followed by false negatives. In speaker recognition systems, false positives and false negatives both indicate incorrectly guessing the speaker when it comes to speaker identification. For speaker verification though, these are completely different. False positives indicate the system granted a user's access to the wrong speaker. This could cause security risks that leads to the exploitation of sensitive information. False negatives indicate the system denied access to an authorized user, which is frustrating, but not detrimental to one's sensitive information.

A. 'Clean' Model

We analyzed the number of MFCCs generated to determine the optimal number of coefficients for accurate speaker identification. Using 4, 8, and 32 MFCCs, we found that the lower number of MFCCs plays a significant role in identifying speakers correctly. The sweet spot is likely somewhere between 16 and 32, but for the purpose of this paper, given the scope, we settled for these attributes. Figure 6 below details the comparison of the number of MFCC features.

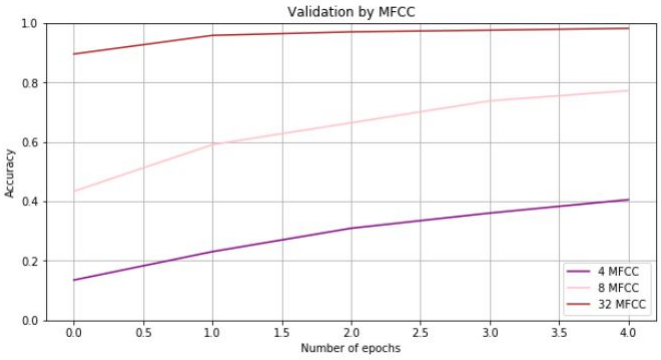

Figure 6. Test accuracy by MFCCs

While 32 coefficients clearly outperforms the 4 and 8, this is computationally exhausting and time consuming. More than eight hours were required to extract 32 coefficients for each audio file.

The final model utilizing 32 Mel-frequency cepstral coefficients resulted in nearly 99 percent testing accuracy. We evaluated this model on a validation set that performed with 98 percent accuracy and a loss of 0.07. Figure 7 details the accuracy and loss metrics for the final model.

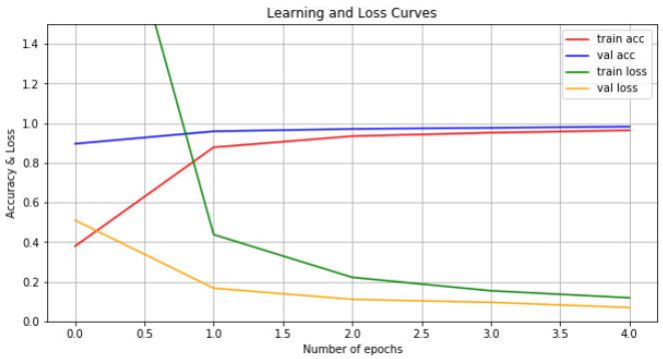

Figure 7. Final 'clean' model

B. 'Other' Model

Again we determined the optimal number of Mel-frequency cepstral coefficients to generate for our neural network, which

can be seen in Figure 8. Recall the nature of this data set: the speakers were classified into this set by a higher Word Error Rate (WER), which signifies the presence of possible volume control issues, noise disruption, or lack of audio quality. This data was incorporated to exist as a validation set in itself, and we'll talk more about its future use in Section V.

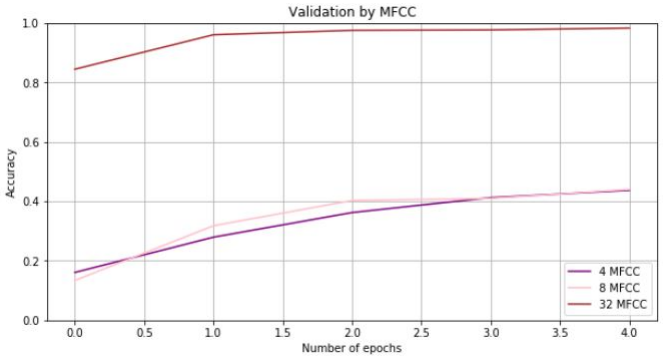

Figure 8. Test accuracy by MFCCs

The results are similar to the 'clean' model performance. 32 Mel-frequency cepstral coefficients performs significantly better, which makes sense considering this data requires more help due to the higher WER of the speakers.

The final 'other' model can be seen as only slightly less accurate than the 'clean' model. With an accuracy of 98 percent and loss value over 0.07, this model is eye to eye with the model with better quality speakers. No doubt the number of MFCC features plays a vital role in speaker identification, as proved with both data sets in our paper.

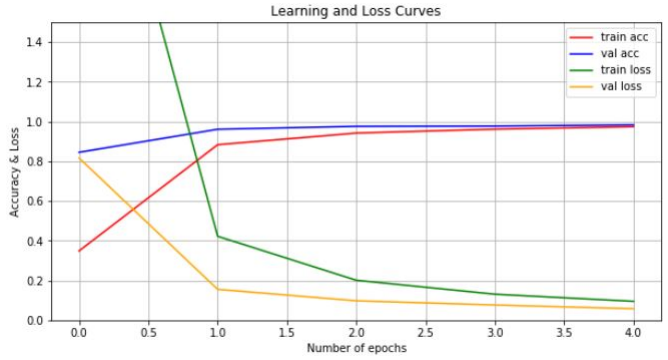

Figure 9. Final 'other' model

V. Future Work

As far as speaker identification goes, steps can be taken to improve results closer to 100 percent accuracy. Frame chunking, signal overlap, and feature extraction are a few concepts to explore deeper. These parameters are extremely robust to slight changes. It's important to incorporate data sets comprising data quality issues including speaker volume control and background noise disruptions to allow for the models to anticipate such imperfections. Other features including the CQCC and CNPCC would be interesting to explore.

Following the speaker identification success, further progress can be made in the realm of speaker authentication. Generally speaking, we want to introduce spoofing as a tactic for cyber criminals to infiltrate inaccessible systems. From here we can develop generative adversarial networks (GAN) that will mitigate this growing concern.

VI. Conclusion

In this paper we discussed the two components of a speaker recognition system. Identification, which was modeled using convolutional neural networks for two data sets, one clean and the other less convenient, and authentication, which connects to the spoofing concept for gaining unauthorized access to another's identity. The data sets came from LibriSpeech, containing 251 and 142 unique speakers, respectively. It was determined that 32 Mel-frequency cepstral coefficients resulted in optimal speaker identification, with 2,048 frame chunking and 512 frame stepping. The final models resulted in nearly 99 percent testing accuracy after only 5 epochs, with a minimal loss value of 0.07. One step made towards spoofing was generating .tiff formatted images for each audio file. From here we can advance towards developing a GAN model for the speaker authentication phase.

Acknowledgment

The authors would like to thank the Open Cloud Institute for the time allowed on the cloud-computing framework and for the sizable memory allocation.

References

[1] A. Boles and P. Rad, *Voice Biometrics: Deep Learning-based Voiceprint Authentication System* UTSA 2017.

[2] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, *LibriSpeech: An ASR Corpus Based on Public Domain Audio Books* The John Hopkins University, Baltimore, MD.

[3] Kinnunen, Tomi, and H. Li. *An Overview of Text-independent Speaker Recognition: From Features to Supervectors* Speech communication 52, no. 1 (2010): 12-40.

[4] Ronald Baken and Robert F. Orlikoff, *Clinical Measurement of Speech and Voice* Cengage Learning, 2000.

[5] Douglas A. Reynolds and Richard C. Rose. *Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker Models* IEEE transactions on speech and audio processing 3, no. 1 (1995): 72-83.

[6] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models* Digital signal processing 10, no. 1-3 (2000): 19-41.

[7] Homayoon Beigi, *Speaker Recognition: Advancements and Challenges* INTECH Open Access Publisher, 2012.

[8] Md. Rashidul Hasan, Mustafa Jamil, and Md Golam Rabbani Md Saifur Rahman, *Speaker identification Using Mel Frequency Cepstral Coefficients* variations 1, no. 4 (2004).

[9] Hector Delgado, Massimiliano Todisco, Md Sahidullah, Nicholas Evans, Tomi Kinnunen, Kong Aik Lee, and Junichi Yamagashi, *ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements* EURECOM, France, MULTISPEECH, France, University of Eastern Finland, NEC Corporation, Japan, National Institute of Informatics, Japan, University of Edinburgh, U.K.

[10] Massimiliano Todisco, Hector Delgado and Nicholas Evans, *Constant Q Cepstral Coecients: A Spoong Countermeasure for Automatic Speaker Verication* EURECOM, France.

[11] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, Vadim Shchemelinin1, *Audio Replay Attack Detection with Deep Learning Frameworks* ITMO University, Russia, STC-Innovations Ltd., Russia.

[12] Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, Xiaolin Li, *Adversarial Examples: Attacks and Defenses for Deep Learning* National Science Foundation Center for Big Learning, University of Florida.
[13] Nicholas Carlini and David Wagner, *Audio Adversarial Examples: Targeted Attacks on Speech-to-Text* University of California, Berkeley.