Deepfake Audio Detection

Madeeha B. Khan
University at Albany, madeehabk3@gmail.com

Sanjay Goel
University at Albany, SUNY, goel@albany.edu

Jaswant Katar Anandan
University at Albany - SUNY, jkataranandan@albany.edu

Jersey Zhao
Albany university, jzhao20@albany.edu

Ramavath Rakesh Naik
University at Albany, rrnaik@albany.edu

Deepfake Audio Detection

Completed Research

Jaswant Katar Anandan
University at Albany, SUNY
jkataranandan@albany.edu

Madeeha Khan
University at Albany, SUNY
mkhan8@albany.edu

Rakesh Naik
University at Albany, SUNY
rrnaik@albany.edu

Jersey Zhao
University at Albany, SUNY
jzhao20@albany.edu

Dr. Sanjay Goel
University at Albany, SUNY
goel@albany.edu

Abstract

Deepfakes, algorithms that use Machine Learning (ML) to generate fake yet realistic content, represent one of the premier security challenges in the 21st century. Deepfakes are not limited to just videos, as deepfake audio is a fast-growing field with an enormous number of applications. Recently, multiple Convolutional Neural Network (CNN) based techniques have been developed that generate realistic results that are difficult to distinguish from actual speech. In this work, we extracted audio features from real and synthesized audio files and determined that Mel-Frequency Cepstral Coefficients (MFCCs) in synthesized audio show a significant difference from the MFCCs in real audio. Using Deep Neural Networks (DNNs), experiments were conducted to train classifiers to detect synthesized audio in different datasets, with highly successful results.

Keywords

Audio Deepfake, Machine Learning, MFCC.

Introduction

The ability to create audio and visual content that the average user cannot easily identify as “fake” poses a serious threat to society. Such content, which lacks any journalistic rigor, can be used to disseminate misinformation on everything ranging from elections to treatments for diseases. Widely used social media sites such as Facebook and Twitter allow information to travel around the world quickly, with very little verification that the information is accurate. Research has shown that fake news travels faster on Twitter than actual news – and by a significant margin (Dizikes, 2018). There have been numerous examples of deepfakes going viral on social media websites, including the infamous 2018 deepfake of former United States President Barack Obama, in which actor Jordan Peele’s face was digitally reconstructed onto Obama’s face using artificial intelligence (AI) (Almars, 2021). In April 2020, a political group in Belgium released a deepfake video of the Belgian prime minister claiming that the COVID-19 pandemic was the result of exploitation and destruction of the natural environment by humans. Deepfakes have also been used for financial scams such as tricking employees of firms into wiring money to bank accounts they think are legitimate (Noone, 2021). While deepfakes such as those may have limited impact, a future deepfake video or “leaked audio” of a major international figure or CEO announcing the failure of a vaccine could have deeper ramifications.

Synthetic (or deepfake) speech can be created using open-source toolkits that may include “basic cut-and-paste waveform fusion techniques” (Borrelli et al, 2021) as well as vocoders that use the source-filter model of a speech signal (Borrelli et al, 2021). The latest milestones in deepfake audio synthesis are “WaveNet (a vocoder developed by DeepMind in 2016) and Tacotron (a text-to-speech algorithm created by Google in 2017)” (David, 2021).

Generative Adversarial Networks (GANs) have been a game changer in the development of deepfake media content. GANs are comprised of two neural networks working in tandem and competing with each other such that one network attempts to generate fake content while the other one attempts to detect if the generated content is fake. By competing with each other in a zero-sum game where one networks gain is the other networks loss, the quality of the fakes continues to improve until it is almost indistinguishable to a human (Engler, 2020).

The tools developed to detect deepfakes (Goled, 2021), while useful, often work effectively only for a small subset of the deepfake audio. Often, the tools used to detect deepfakes are then leveraged by programs seeking to create even better more undetectable fake versions. Techniques for deepfake detection are based on classification where they start by extracting features from the media source (audio or video) and then training classifiers using a dataset of fake and real media content to detect fakes. This research identifies key audio features that can be used to distinguish between real and synthesized audio and to train DNNs to classify real and fake audio files. The rest of the paper is organized into six sections: literature review, feature extraction, feature selection, dataset exploration, results, and conclusion.

Literature Review

There are four different modes of deepfakes based on the media source: audio, video, image and text. In our research, we focus on audio deepfakes. In the past decades, a multitude of research has been conducted to find better techniques to detect synthesized audio. In this section we summarize prior work done in the field, which can be grouped into the following domains: analyzing the statistical features of audio signals, monitoring neuron behaviour, exploiting affective cues between visual and audio modalities, and training neural networks.

The Bispectral Analysis (AlBadawy et al, 2019) and The NTU Approach (Xiao et al, 2015) both use the statistical features of audio signals to detect deepfake audio. The Bispectral Analysis research found that audio synthesized using DNNs has unusual spectral correlations not observed in human speech. By computing the bicoherence magnitude and phase for each audio clip, they obtained four statistical moments for each feature: mean, variance, skewness, and kurtosis, and trained a collection of five separate logistic regression classifiers. If the maximum classification score across all five classifiers was above a specified threshold, then the audio was classified as synthesized. Tests conducted using Bispectral Analysis resulted in an area under the curve (AUC) of 0.99, with AUC decreasing as additive noise increased (AlBadawy et al, 2019). The NTU Approach (Xiao et al, 2015) used high dimensional magnitude and phase-based features and long-term temporal information to detect spoofed audio. This research built a multiple component system, each using specific type of features. For each component system, a Multilayer Perceptron (MLP) was trained to predict the posterior probability of the input feature patch extracted from the audio clips and their scores were averaged to produce the final score for detection. This research had an equal error rate (EER) of 0.29% for known spoofing types, and a 5.23% EER for unknown spoofing types (Xiao et al, 2015).

The DeepSonar (Wang et al, 2020) approach monitored neuron behaviors of Speaker Recognition (SR) system, i.e., a DNN, to discern AI-synthesized fake voices. They adopted a DNN-based SR system to capture the layer-wise neuron behaviors for both real and fake voices and determine the activated neurons with designed neuron coverage criteria. The captured neuron behaviors were formed as input feature vectors for training a simple supervised binary-classifier based on shallow neural networks to predict whether a clip of voice is a human speech or synthesized. Experimental results showed that DeepSonar has an average accuracy higher than 98.1% and an EER lower than 2% (Wang et al, 2020).

The Affective Cues method (Mittal et al, 2020) focused on video deepfakes and exploited the relationship between the visual and audio modalities extracted from the same video (Mittal et al, 2020). Affective cues are specific features that convey rich emotional and behavioral information to human observers and help them distinguish between different perceived emotions (Wang et al, 2017). These affective cues comprise of various positional and movement features, such as dilation of the eye, raised eyebrows, volume, pace, and tone of the voice. The research exploited this correlation between modalities and affective cues to classify real and fake video. Results from tests conducted on two different datasets show an AUC of 84.4% and 96.6%, respectively (Mittal et al, 2020).

The Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection (Chintha et al, 2020) research developed two methods for audio spoof detection: CRNNSpoof and WIRENetSpoof. CRNNSpoof is a convolution-recurrent neural network with raw audio as the input and uses five 1-D convolution layers to learn useful representations. WIRENetSpoof is a convolution neural network and "the input audio is either clipped or repeated to a fixed length of four seconds before the log-mel spectrogram is obtained" (Chintha et al, 2020). Experimental results showed that CRNNSpoof and WIRENetSpoof performed with an EER of 4.27% and 7.37% respectively on the evaluation set (Chintha et al, 2020).

While these research studies have had successful results, they also have a few limitations. The Bispectral Analysis study relies on the fact that "current speech-synthesis algorithms introduce specific and unusual higher-order bispectral correlations that are not typically found in human speech" (AlBadawy et al, 2019). However, if new techniques are created that can generate synthesized audio without these higher-order bispectral correlations, then the research solution will fail to detect the synthesized audio. The NTU Approach results show that it worked on "most known and unknown spoofing types" (Xiao et al, 2015) except for one type of spoofing speech - the S10 spoofing speech type in the dataset used. One of the limitations discussed in the DeepSonar research is that if synthetic audio generating techniques add "additional loss function by modeling the neuron behaviors" (Wang et al, 2020), or introduce real-world noises at a high magnitude, then the tool's performance decreases. The Affective Cues method only works on video deepfakes and cannot be applied to just audio deepfakes. And lastly, in the Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection study, "both models fail to perform better than the given baseline and benchmark methods on the development set" Chintha et al, 2020).

None of the research studies above extracted audio features to aid in the detection of deepfake audio. In our research, we built three different ML models and tested them on different datasets to determine which one consistently performed better than the others. We also analyzed different features that can be extracted from audio files and determined that MFCCs show the highest difference between real and synthesized audio files.

Feature Extraction

Most of the datasets for fake and real audio available online are in the .wav file format. As a first step, we extracted relevant features from the raw audio data, such as images and coefficients, by converting the raw audio clip into a signal. To perform this conversion, we identified the Sample Rate at which the raw audio is loaded into the module and converted it into an audio signal (Velardo, 2020). The signal is based on the time/amplitude format, where the x-axis represents the time and the y-axis represents the amplitude, as shown in Figure 1.

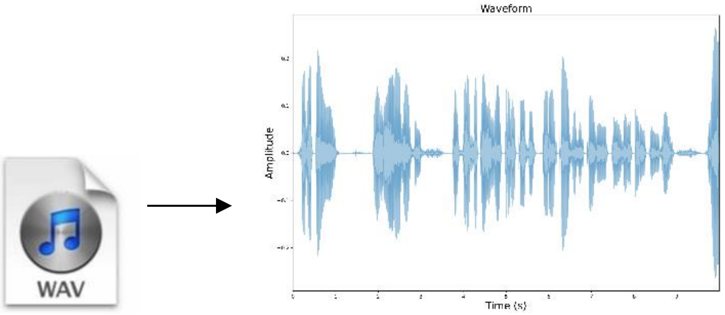


Figure 1: Converting a .wav file into an audio signal

Audio signals contain hundreds of features that can be extracted, such as audio brightness, depth, roughness, and hardness etc. This makes it difficult to identify the optimal feature(s) that would be instrumental in training ML models to detect synthesized audio. This limitation is reduced by extracting the magnitude of the frequencies in the audio signal. We converted the time/amplitude of the signal into a frequency/magnitude spectrum using Fast Fourier transform, which converts time-based signals into frequency-based magnitudes and helps identify which frequencies are essential in a given audio file (Velardo, 2020), as shown in Figure 2.

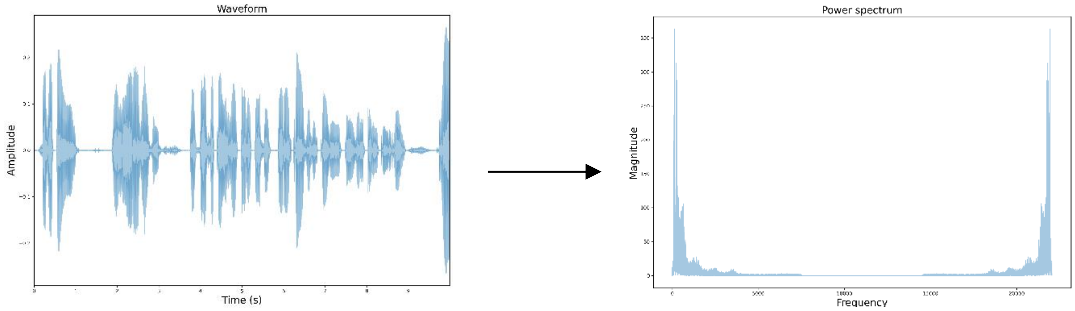


Figure 2: Converting an audio signal into a spectrum

The Fast Fourier transform output consists of a mirror image of the low and high frequencies. For computational purposes, we only utilized the first half of the spectrum (see Figure 3).

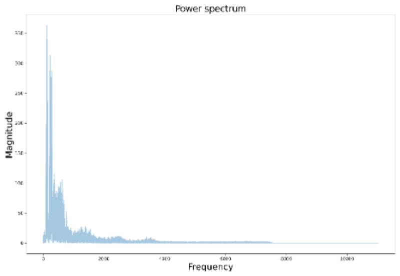


Figure 3: Splitting the spectrum to extract the first half

The main limitation of this spectrum is that it is not time based. Without the timeline, it is difficult to separate the data and its processing. Therefore, we used the short time Fourier transform to convert the frequency / magnitude format into time / frequency / magnitude format, where the time is plotted on the x-axis, frequency distribution is plotted on the y-axis and the magnitude is represented through the color intensity (Velardo, 2020) as shown in Figure 4. This time-based spectrogram clearly represents the magnitude of a particular frequency at a particular time. Using this time-based spectrogram, we analyzed the audio files and extracted the following features: MFCCs, RMS, Zero Crossing Rate, Chroma Frequency and Spectral Roll off.

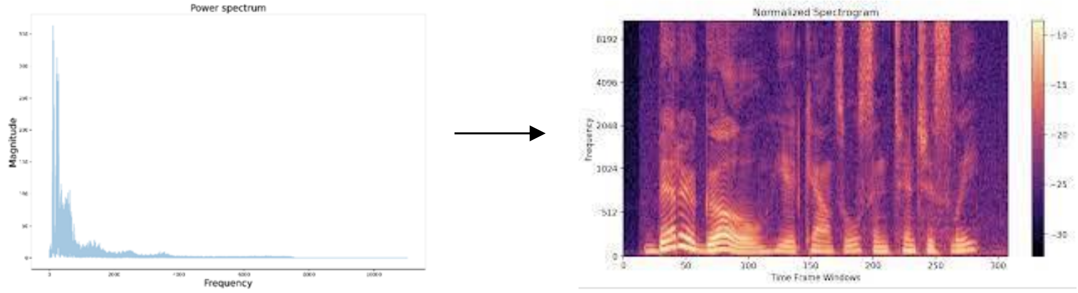


Figure 4: Converting the spectrum into a spectrogram

Feature Selection

We trained five different Recurrent Neural Network (RNN) models, one for each of the five extracted audio features and observed that the ML model with MFCCs as the independent variable had the highest precision, accuracy, and recall, as illustrated in Table 1.

Features	Output	Precision	Accuracy	Recall	F-1 Score
MFCCs	Fake	0.95	0.97	0.99	0.97
	Real	0.99	0.97	0.95	0.97
RMS	Fake	0.65	0.67	0.77	0.71

	Real	0.7	0.67	0.57	0.63
Zero Crossing	Fake	0.61	0.61	0.73	0.67
	Real	0.61	0.61	0.47	0.53
Chroma Frequency	Fake	0.69	0.72	0.82	0.75
	Real	0.76	0.72	0.6	0.67
Spectral Roll off	Fake	0.58	0.57	0.62	0.6
	Real	0.57	0.57	0.52	0.54

Table 1: Classification report for five RNN models

Therefore, we identified MFCCs (using 13 coefficients) as the significant feature to be used in training a classifier to detect synthesized audio. Sounds generated by humans are filtered by the shape of the vocal tract, including the tongue and the teeth; and this shape determines what audio comes out. Determining the shape accurately gives an accurate representation of the phoneme being produced (Bhat, 2020). "The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the role of MFCCs is to accurately represent this envelope" (Bhat, 2020). MFCCs capture the timbral and textural aspects of sound from an audio and are based on the frequency domain, which is useful in extracting information from the audio. MFCCs approximate the human auditory system, which is very useful for the classification of the audio. It consists of 13 to 40 coefficients which are calculated in each frame of the given audio. Using a sample fake and real audio file, we generated a time-based spectrogram and the MFCCs spectrogram. The plots derived from the audio data are represented in logarithmic scale, and the results are represented in Table 2 below. From the results in Table 2, we can observe that there are significant differences between a real audio file and a fake audio file, and this difference can be leveraged in training a classifier.

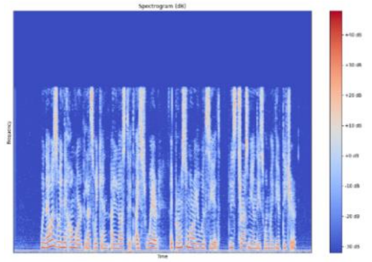
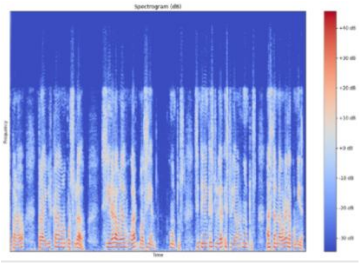
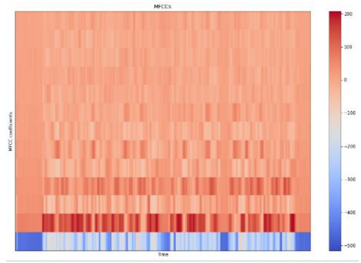
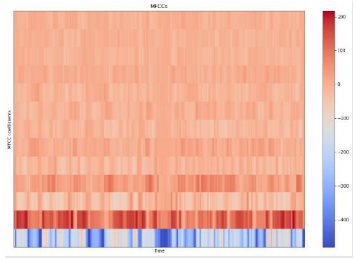
	Real Audio	**Fake Audio**
Time-based Spectrogram		
MFCCs Spectrogram		

Table 2: The time-based spectrogram and MFCCs spectrogram for a sample fake audio and real audio file

The number of MFCCs generated by an audio clip depends on the length of the audio clip, i.e., a two-second audio clip produces 87 MFCCs while a three-second audio clip produces 109 MFCCs. This poses a challenge since ML models need a standardized input (same shape / size). To overcome this, we divided each audio clip into segments of standard size (i.e., 2 seconds), so that each segment generates the same number of MFCCs. If the audio clip's length is an odd number of seconds (i.e., 31 seconds), the last one second is dropped off. The segments are then fed to the models and the resulting predictions are analyzed to identify the majority / max count which is used as the final prediction (0 – fake or 1 - real) for the audio clip, (see Figure 5).

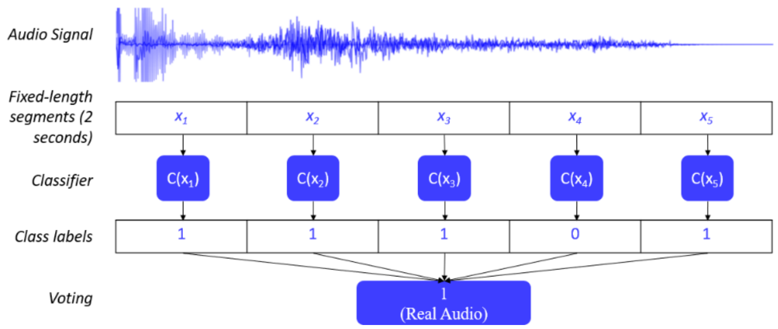


Figure 5: Segmenting the data to feed into the classifier and using the max count as the final prediction

Dataset Exploration

To train our ML models, we identified publicly available datasets that contain synthetic audio created based on the latest technology and were diverse and balanced in gender and sources to train our ML models. We analyzed different audio datasets available online and noticed that some datasets were either too small, biased or didn't have the data formatted for easy use. The H-Voice dataset (Ballesteros et al, 2020) consisted of histograms of original and fake voice recordings obtained by Imitation and Deep Voice, but this research required audio files in the .wav format. The FakeAVCeleb dataset (Khalid et al, 2021) contained 490 real and 25,000 fake audio files, and the unbalanced distribution would have resulted in a poor performance on the minority class. We selected the following datasets to train our model with varying results:

Dataset 1: FoR: A Dataset for Synthetic Speech Detection

This dataset was created in 2019 as a part of a research project to provide a base for other studies in speech synthesis and synthetic speech detection. It "contains more than 198,000 utterances from the latest deep-learning speech synthesizers as well as real speech" (Reimao et al, 2019). It includes 87,000 synthetic utterances as well as 111,000 real utterances. The fake audio is generated by speech synthesis technologies such as Google Test to Speech (TTS), Amazon TTS, Microsoft TTS, and Baidu TTS, using Neural Network architectures. The real audio is obtained from open-source speech datasets as well as other sources of real speech, such as TED Talks and YouTube videos. Both real and fake audio were converted to a mono (single channel) using the SoX tool so that the channels don't become a distinguishing factor. The dataset consisted of multiple versions, with different levels of pre-processing, and we utilized the dataset version that contained the real and fake audio clips balanced in terms of gender and class and normalized in terms of sample rate, volume, and number of channels. It also truncated all files at 2 seconds so that length of the file doesn't become a distinguishing factor.

Dataset 2: Fake Voice Recordings (Imitation)

This dataset (Rodríguez et al. 2019) contains fifty original voice recordings, and fifty fake voice recordings created using the imitation algorithm to facilitate machine learning projects. While the dataset is small, it contains a range of sources, genders, etc. and is ideal for training simple classifiers.

Dataset 3: ASVspoof 2015

This dataset was created in 2015 to be used in the first Automatic Speaker Verification Spoofing (ASVspoof) and Countermeasures Challenge. The dataset is quite large and contains real speech collected from "106 speakers (45 male, 61 female) and with no significant channel or background noise effects" (Wu et al, 2015). A number of spoofing algorithms were used to generate the synthesized audio from the genuine data.

Results

We used three ML models: Multilayer Perceptron (MLP), CNN, and Recurrent Neural Networks (RNN) to build our classifiers, using the MFCCs extracted from the audio files as the independent variables. We used deep learning algorithms for the classification models because most of the latest synthesized audio is created using deep learning algorithms. We chose the most commonly used deep learning algorithms and analyzed their performance to determine which model has the highest precision, accuracy, and recall.

Multi-Layer Perceptron (MLP)

In the MLP model, the MFCCs are the independent variables, and the algorithm classifies the audio file as fake or real. The input (MFCCs) is first flattened into the input shape of [87,13] where the 87 represents the number of MFCCs produced for each 2 seconds of audio data and the 13 represents the number of coefficients of each MFCCs' values. The number of MFCCs for each audio depends upon the duration of the audio. They are flattened so that each value is fitted into each node of a MLP structure, and the activation function is applied. Our model consists of three hidden dense layers where the 1st layer consists of 512 nodes with "relu" activation. Similarly, the 2nd layer has 256 nodes with "relu" activation, and the final dense layer is shortened into 64 nodes with "relu" activation. The output layer has two nodes each representing 0 (fake) and 1 (real) respectively, as illustrated in Figure 6.

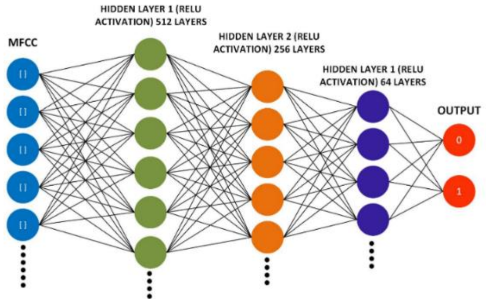

HIDDEN LAYER 1 (RELU ACTIVATION) 512 LAYERS
HIDDEN LAYER 2 (RELU ACTIVATION) 256 LAYERS
MFCC
HIDDEN LAYER 1 (RELU ACTIVATION) 64 LAYERS
OUTPUT
[]
[]
[]
[]
[]
0
1

Figure 6: Architectural design of our MLP Model

The classification report for the MLP model for all datasets is shown in Table 3.

Dataset	Feature	Output	Precision	Accuracy	Recall	F-1 Score
Dataset 1	MFCC	Fake	0.93	0.94	0.95	0.94
		Real	0.95	0.94	0.93	0.94
Dataset 2	MFCC	Fake	0.66	0.72	0.87	0.75
		Real	0.82	0.72	0.58	0.68
Dataset 3	MFCC	Fake	0.90	0.93	0.97	0.93
		Real	0.97	0.93	0.90	0.93

Table 3: Classification Matrix for MLP with MFCC as the independent variable

Convolutional Neural Network (CNN)

This deep learning algorithm is popularly used to classify image datasets (Thomas, 2019), since it uses three properties (pixel value of a row, pixel value of a column and number of RGB channels) of each pixel in an image. Since the MFCCs of each audio file are a two-dimensional array [87,13], they need to be converted into a three-dimension format. This can be done by adding 1 as a default channel value for all the audios in the dataset. Our CNN model consists of three convolutional layers in total, which has 32 filters and works with "relu" activation. The first two convolutional layer are made up of (3,3) kernel size with Max pooling layer of (3,3) kernel size and of (2,2) strides. The last convolutional layer is made of (2,2) kernel size with Max pooling layer of (2,2) strides. Each layer is batch normalized for faster training and to achieve high learning rates. The output from the convolutional layers is then flattened into dense layer of 64 nodes with "relu" activation and finally it outputs the results as 0 (Fake) and 1 (Real) using "softmax" activation. The data processing is repeated for 30 epochs to improve the benchmark of the results, (see Figure 7).

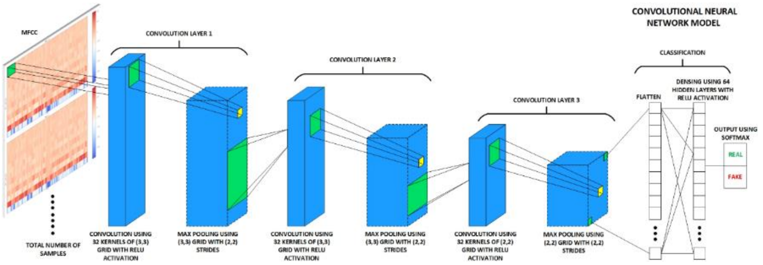

CONVOLUTIONAL NEURAL NETWORK MODEL
MFCC
CONVOLUTION LAYER 1
CONVOLUTION LAYER 2
CONVOLUTION LAYER 3
CLASSIFICATION
FLATTEN
DENSING USING 64 HIDDEN LAYERS WITH RELU ACTIVATION
OUTPUT USING SOFTMAX
REAL
FAKE
TOTAL NUMBER OF SAMPLES
CONVOLUTION USING 32 KERNELS OF (3,3) GRID WITH RELU ACTIVATION
MAX POOLING USING (3,3) GRID WITH (2,2) STRIDES
CONVOLUTION USING 32 KERNELS OF (3,3) GRID WITH RELU ACTIVATION
MAX POOLING USING (3,3) GRID WITH (2,2) STRIDES
CONVOLUTION USING 32 KERNELS OF (2,2) GRID WITH RELU ACTIVATION
MAX POOLING USING (2,2) GRID WITH (2,2) STRIDES

Figure 7: Architectural design of our CNN Model

The classification report for the CNN model for all datasets is shown in Table 4.

Dataset	Feature	Output	Precision	Accuracy	Recall	F-1 Score
Dataset 1	MFCC	Fake	0.95	0.97	0.99	0.97
		Real	0.99	0.97	0.94	0.97
Dataset 2	MFCC	Fake	0.95	0.93	0.90	0.93
		Real	0.91	0.93	0.95	0.93
Dataset 3	MFCC	Fake	0.99	0.99	0.99	0.99
		Real	0.99	0.99	0.99	0.99

Table 4: Classification Matrix for CNN with MFCC as the independent variable

Recurrent Neural Network – Long Short-Term Memory (RNN-LSTM)

This deep learning algorithm is popularly used to process audio data (Hewahi et al, 2019). Many music producing software use this algorithm to produce new musical notes depending on the previous note. The main specialty of this classification algorithm is that each node of the RNN network produces output that depends on the output of the previous node. Our RNN-LSTM model uses two layers; the 1st layer is a sequence-to-sequence layer that consists of 64 nodes, and the second layer is a sequence-to-vector layer of 64 nodes. The output from the two layers is passed into a dense layer of 64 nodes and "relu" activation. Finally, the output from the previous layer results in 0 (Fake) and 1 (Real) using "softmax" activation, as illustrated in Figure 8.

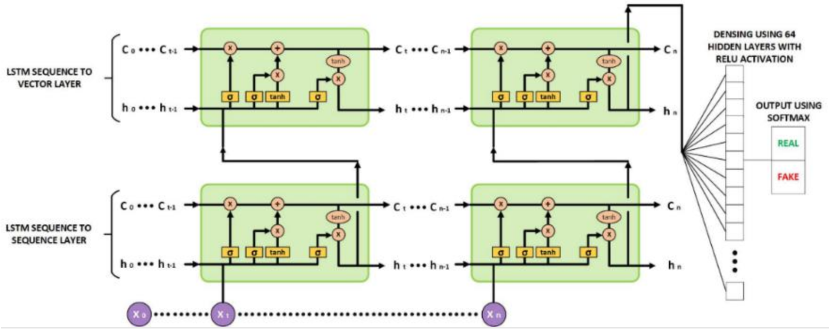


Figure 8: Architectural design of our RNN – Long Short-Term Memory Model

The classification report for the RNN-LST model for all datasets is shown in Table 5.

Dataset	Feature	Output	Precision	Accuracy	Recall	F-1 Score
Dataset 1	MFCC	Fake	0.97	0.98	0.99	0.98
		Real	0.99	0.98	0.97	0.98
Dataset 2	MFCC	Fake	0.86	0.82	0.72	0.79
		Real	0.79	0.82	0.90	0.84
Dataset 3	MFCC	Fake	0.97	0.98	0.99	0.98
		Real	0.99	0.98	0.97	0.98

Table 5: Classification Matrix for RNN-LSTM with MFCC as the independent variable

Evaluating the Performance

The CNN and RNN models performed exceptionally well for Datasets 1 and 3 (the large datasets), with Area under the Curve (AUC) scores of [0.997 (CNN - Dataset 1), 0.997 (RNN – Dataset 1)] and [0.999 (CNN – Dataset 3) and 0.998 (RNN – Dataset 3)] respectively. But their performance on Dataset 2 (the small dataset) was a little lower, with Area under the Curve (AUC) scores of [0.971 (CNN - Dataset 2) and 0.929 (RNN – Dataset 2)] respectively. Since Dataset 2 is very small, the decreased performance can result from underfitting.

However, while the MLP model performed slightly lower than the CNN and RNN models for all datasets, it had a high performance for both Datasets 1 and 3, with AUC scores of [0.984 (MLP - Dataset 1) and 0.988

(MLP – Dataset 3)] respectively. The performance decreased significantly for Dataset 2, with an AUC score of [0.779 (MLP - Dataset 2)].

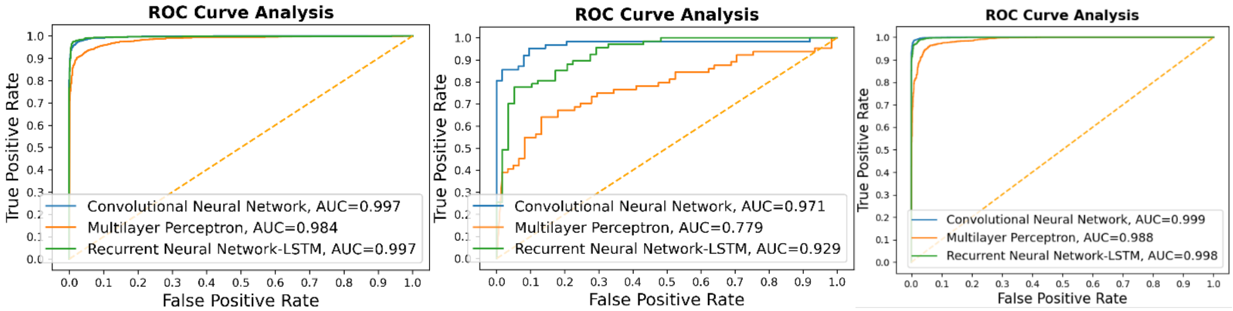


Figure 9: AUC for Dataset 1 (left), Dataset 2 (middle) and Dataset 3 (right)

Small training datasets usually result in decreased performance, because "over-constrained model(s) will underfit the small training dataset, whereas under-constrained model(s) will likely overfit the training data" (Brownlee, 2020). Datasets 1 and 3 were larger (1.5 GB and ~5 GB respectively) compared to Dataset 2 (131 MB), and so Dataset 2 consistently performed lower than the other datasets for each classifier. Additionally, the MLP model was small with few layers and parsing a large dataset through it can lead to overfitting, which can result in a slight decrease in the performance of the model, as compared to the other classifiers.

Conclusion

In this research, we explored different audio features and determined that MFCCs are a key feature in distinguishing between real and synthesized audio. We used diverse datasets, which included both the synthesized audio created from the latest speech synthesis technologies and had a good range of gender, class etc. We trained three DNN models to classify real and synthesized audio. The research found that the CNN model, with 99% precision, 99% accuracy and 99% recall and the RNN model, with 97% precision, 98% accuracy and 99% recall can identify synthesized audio with high precision, accuracy, and recall. This provides a novel technique for detecting fake audio and can help combat the increasing cases of scamming and mis/disinformation. This method can be implemented to detect synthesized audio in real time with high precision, accuracy, and recall, and would be the first step in defending against deepfake cyber threats.

The research has produced some promising results, but further work is needed to expand the scope of the work and improve the performance of the models. An area of further research would be to vary the time length used to segment the audio clips (current research uses 2 seconds) and evaluating if/how that affects the performance of the classifiers. The research can also be further augmented by using other datasets to train and test the models, and evaluation how their performance changes. Lastly, this research focused on using deep neural network to build classifiers and could be enhanced by using other classifiers and evaluating their performance.

REFERENCES

AlBadawy, Ehab A., et al. "Detecting AI-Synthesized Speech Using Bispectral Analysis." *CVPR Workshops* (2019).

Almars, A. (2021) Deepfakes Detection Techniques Using Deep Learning: A Survey. *Journal of Computer and Communications*, 9, 20-35. doi: 10.4236/jcc.2021.95003.

Ballesteros L., Dora, M., Rodriguez, Y., Renza, D. (2020), "H-Voice: Fake voice histograms (Imitation+DeepVoice)", *Mendeley Data*, V4, doi: 10.17632/k47yd3m28w.4.

Bhat, S. (2020, August 6). From COMET_ML import experiment. *Medium*. Retrieved February 25, 2022, from https://medium.com/soundrecognition-using-mfccs/from-comet-ml-import-experiment-b2827b85fe32.

Borrelli, C., Bestagini, P., Antonacci, F., Sarti, A., & Tubaro, S. (2021). Synthetic speech detection through short-term and long-term prediction traces. *Eurasip Journal on Information Security*, 2021(1). https://doi.org/10.1186/s13635-021-00116-3.

Brownlee, J. (2020, August 25). Impact of dataset size on Deep Learning Model Skill and performance estimates. *Machine Learning Mastery*. Retrieved April 25, 2022, from https://machinelearningmastery.com/impact-of-dataset-size-on-deep-learning-model-skill-and-performance-estimates/.

Chintha, A., Thai, B., Sohrawardi, S. J., Bhatt, K., Hickerson, A., Wright, M., & Ptucha, R. (2020). Recurrent convolutional structures for audio spoof and video deepfake detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5), 1024-1037.

David, D. (2021, May 10). Analyzing the rise of Deepfake Voice Technology. *Forbes*. Retrieved February 18, 2022, from https://www.forbes.com/sites/forbestechcouncil/2021/05/10/analyzing-the-rise-of-deepfake-voice-technology/?sh=33df4bb86915.

Dizikes, P. MIT News Office. (2018, March 8). Study: On twitter, false news travels faster than true stories. *MIT News | Massachusetts Institute of Technology*. https://news.mit.edu/2018/study-twitter-false-news-travels-faster-true-stories-0308.

Engler, A. (2020, May 6). Fighting deepfakes when detection fails. *Brookings*. https://www.brookings.edu/research/fighting-deepfakes-when-detection-fails/.

Goled, S. (2021, January 13). Top ai-based tools & techniques for deepfake detection. *Analytics India Magazine*. Retrieved April 24, 2022, from https://analyticsindiamag.com/top-ai-based-tools-techniques-for-deepfake-detection/.

Hewahi, N., AlSaigal, S., & AlJanahi, S. (2019, August 6). Generation of music pieces using Machine Learning: Long short-term memory neural networks approach. *Taylor & Francis Online*. Retrieved February 25, 2022, from https://www.tandfonline.com/doi/full/10.1080/25765299.2019.1649972.

Khalid, H., Tariq, S., and Woo, S. 2021. FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset.

Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020, October). Emotions Don't Lie: An Audio-Visual Deepfake Detection Method using Affective Cues. *In Proceedings of the 28th ACM international conference on multimedia* (pp. 2823-2832).

Noone, G. (2021, March 15). Audio deepfake scams: The growing threat explored. *Tech Monitor*. https://techmonitor.ai/techonology/cybersecurity/growing-threat-audio-deepfake-scams.

Reimao, R. and Tzerpos, V. "FoR: A Dataset for Synthetic Speech Detection," 2019 *International Conference on Speech Technology and Human-Computer Dialogue* (SpeD), 2019, pp. 1-10, doi: 10.1109/SPED.2019.8906599.

Rodríguez, Y.; Ballesteros L., Dora, M., Renza, D. (2019), “Fake voice recordings (Imitation)”, *Mendeley Data,* V1, doi: 10.17632/ytkv9w92t6.1.

Thomas, C. (2019, May 27). An introduction to Convolutional Neural Networks. *Towards Data Science*. Retrieved February 25, 2022, from https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7.

Velardo, V.; “Preprocessing audio data for Deep Learning”. (2020). *YouTube*. Retrieved December 9, 2021, from https://youtu.be/Oa_d-zaUti8.

Wang, C., Zhou, Z., Jin, X., Fang, Y., and Lee, M. 2017. The influence of affective cues on positive emotion in predicting instant information sharing on microblogs: Gender as a moderator. *Information Processing & Management* 53, 3 (2017), 721–734.

Wang, R., Juefei-Xu, F., Huang, Y., Guo, Q., Xie, X., Ma, L., and Liu, Y. 2020. DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. *Proceedings of the 28th ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, 1207–1216.* DOI :https://doi.org/10.1145/3394171.3413716.

Wu, Zhizheng; Kinnunen, Tomi; Evans, Nicholas; Yamagishi, Junichi. (2015). Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2015) Database, [dataset]. University of Edinburgh. *The Centre for Speech Technology Research (CSTR).* https://doi.org/10.7488/ds/298.

Xiao, X., Tian, X., Du, S., Xu, H., Siong, C.E., & Li, H. (2015). Spoofing speech detection using high dimensional magnitude and phase features: the NTU approach for ASVspoof 2015 challenge. *INTERSPEECH*.