Synthetic Speech Detection: A Courtroom Perspective

Amit Ghimire*
Department of Computer Science,
University of British Columbia
Vancouver, Canada
amitghi@student.ubc.ca

Joseph Janssen*
Department of Earth, Ocean and
Atmospheric Sciences, University of
British Columbia
Vancouver, Canada
Institute of Applied Mathematics,
University of British Columbia
Vancouver, Canada
joejanssen@eoas.ubc.ca

James Tang*
Department of Computer Science,
University of British Columbia
Vancouver, Canada
tangytob@student.ubc.ca

ABSTRACT

Legally obtained audio is admissible in criminal court, though it first must be verified. As the technology for producing synthetic audio via machine learning becomes better and more accessible, this verification process will require more care. The problem of receiving an audio clip of a claimed specific person and evaluating its authenticity has not been well studied, thus we build random forests and deep neural network models to solve this problem. Further, the differences between training a model on a specific person versus training a model on multiple people are explored. For limited datasets, as is the case in our scenario, neural networks with precomputed features outperform both random forests and end-to-end models. We show that both random forests and neural networks with manual feature extraction work well for detecting synthetic speech if samples from the synthesis algorithm are included in the training set. Random forest shows variation in the specific versus general speaker model whereas neural networks with precomputed features has similar performance among both general and specific speaker models. Both neural networks and random forests fail to perform well when testing on unknown synthesis algorithms so we propose further approaches to improve generalizability.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Applied computing** → *Law*.

KEYWORDS

synthetic audio, neural networks, random forests, courtroom, signal processing

ACM Reference Format:
Amit Ghimire, Joseph Janssen, and James Tang. 2021. Synthetic Speech Detection: A Courtroom Perspective. In *SPML 2022: International Conference on Signal Processing and Machine Learning, August 04–06, 2022, Dalian, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Authors contributed equally to this research.

SPML 2022, August 04–06, 2022, Dalian, China

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

1 INTRODUCTION

A Pakistani Tik-Toker is caught up in an extortion case, a UFC fighter is accused of attempted murder, and a young Georgia man confesses to a murder [13, 25, 26]. What do all these incidences have in common? They all relied on recorded audio evidence. But, something significant separates the Georgia case from the others. The audio recording in the Georgia case was found to be fake and recorded instead by a malicious human impersonator. As advancements in machine learning and artificial intelligence continue, these dangerous but easily detectable human impersonators will be replaced by intelligent deepfake generators which can impersonate anyone after training on a few seconds of real audio [17].

Audio information is essential for criminal investigations. It can provide information about criminal whereabouts, the type of weapons used, and the type of vehicle. Most importantly, audio evidence is useful for speech and talker identification which plays an important role for establishing suspects, associates, or further crimes such as threats or planning a crime [7]. Even if an audio clip isn't used for criminal investigation, it may still contain potentially libelous and sensitive information. Before any of this becomes admissible in court, the audio often must be enhanced, analyzed, and authenticated. There are many ways in which audio can be authenticated including trying to find unexplained and sudden changes in background noise or voice frequency, visual inspection, metadata analysis. These are all done by human experts [7]. Though some software-based automatic speaker verification systems exist, their ability to detect synthetic audio remains low [36].

As of 2016, some machine-generated speech could already fool human detectors over 10% of the time [36], and as of 2019 the percentage rose to 45% [24]. Indeed, this is a problem only robust machine learning and signal processing can solve. With the rapid development of deepfake audio datasets and even with the continuation of synthetic audio detection challenges such as ASVspoof [17, 23, 38], synthetic audio detection and anti-spoofing techniques are all getting a lot of research attention [6, 8, 9, 15, 16, 30, 39]. However, none of the papers have classified real and fake audio from a specific person. Receiving audio from a specific person is much more realistic in the courtroom setting, thus we would like to test to see if models focused on a specific person can perform better compared to models more generally trained to find spoofed audio. Clearly, more research is needed to develop robust, accurate,

and easily trainable audio-only person-specific deepfake detection algorithms [17], thus this paper will take a step in this direction. Our problem is posed as follows: given a piece of audio submitted as evidence, can we determine if it came from the proposed person or it is synthetic.

The remainder of this paper is organized as follows: Section 2 presents previous works related to synthetic audio detection. Section 3 describes our data sources and proposed methods. Section 4 shows the design and results of our experiments. Section 5 discusses the results of our paper and provides limitations and directions for future work. Finally, Section 6 gives our conclusions.

2 RELATED WORK

Currently, most deepfake detection algorithms are applied to video and try to detect the differences between mouth movements and audio [18, 34]. However, this cannot be done when a corresponding video does not exist or is not available. Thus, detecting if an audio file is fake is considerably more difficult compared to when both audio and video data are available. This has been demonstrated by testing three sets of models trained on FakeAVCeleb, which is a racially unbiased dataset containing over 20,000 fake audio and video clips [17]. The models trained with only audio had an accuracy of 76%, while the ensemble model had 78% accuracy and the video-only model had 81% accuracy [16]. It was concluded that further work needed to be done for audio-only deepfake detection.

The FoR dataset was created by Reimao and Tzerpos and contains over 87,000 synthetic utterances from 33 people and 7 different text-to-speech algorithms [23]. Interestingly, in the same paper, they show that the deep learning model VGG19 is robust to noise under most circumstances, but when noise levels exceed 35%, classification accuracy drops off sharply. In a more detailed analysis of the same dataset, the same author provides comparisons between many statistical learning and deep neural network models in their ability to detect synthetic audio. Among statistical learning methods, random forests (98.5%) show much higher validation accuracies compared to naive Bayes (79.2%), SVMs (92.1%), and decision trees (93.2%). All of these highest accuracies came from using Mel-frequency cepstral coefficients (MFCC) which performed much better compared to short-time Fourier transforms (STFT) and Fourier transforms (FFT), and slightly better compared to Mel-spectrograms (Mel) and constant-Q transform coefficients (CQT). About half of the deep learning architectures outperformed random forests on the validation set, including VGG16 (100%), VGG19 (99.9%), InceptionV3 (99.9%), MobileNet (99.2%), and XceptionNet (98.6%). The above deep learning frameworks preferred using STFT compared to Mel, MFCC, and CQT. On the test set which contained an unknown attack not included in training, random forests, again, outperforms the other statistical learning methods, though the accuracy drops to 86.9% of which was obtained using CQT variables. Evaluating the deep learning models on the test set also concluded the CQT variables were the best at generalizing to the unknown attack. VGG19 and MobileNet obtained accuracies between 90-92%.

As a result of the ASVspoof competitions, there has been a notable amount of exploration of different deep learning architectures for spoofed audio detection. It is generally agreed that CQT and its derivatives are great variables for detecting synthetic audio, however this has recently been disputed [12]. A relatively small fully end-to-end res-net model achieved a 1% error rate and showed great generalizability [12]. Graph attention networks were used to obtain less than a 1% equal error rate for the ASVspoof competition dataset [15].

3 METHODS

3.1 Dataset

We will use Tacotron 2, an attention-based convolutional neural network model [27] to generate voice clips of Donald Trump, George Takei, and Drake. Data for these celebrities is generated by the fakeyou.com text-to-speech website (https://fakeyou.com/), which uses a Tacotron 2 model. We also gather real versions of their voices by accumulating YouTube clips from a wide range of speeches and interviews. The Tacotron 2 generated data uses the audio transcript of the real clips to generate the deepfakes. We did not have data from Trump in the FakeAVCeleb dataset. This was a minor error on our part because we wanted three celebrities which existed in FakeAVCeleb and Tacotron 2. We downloaded Trump data from both datasets, but later realized the Trump data in FakeAVCeleb was for Melania Trump, not Donald Trump.

To generate additional fake voice clips for an unknown attack, we use the FakeAVCeleb dataset to test our model on Drake and Takei voice samples [17]. The voices in this dataset are generated by SV2TTS [14], which is a neural network model based on transfer learning, LSTMs, and attention.

For the Tacotron 2 data, the voice clips that correspond to each text source are joined together. For FakeAVCeleb data, the voice clips that corresponds to each speaker are joined together. The joined clips are then resampled at 32,000 hertz. Then, the joined audio is cut into numerous 2-second-long clips. Any sample that is shorter than 2 seconds is 0 padded to fit this criteria. The number of voice clips for each speaker is given in Table 1.

Table 1: The number of real and fake clips for each speaker.

	Number of clips		
	Donald Trump	George Takei	Drake
Real	785	676	283
Tacotron	400	264	241
FakeAVCeleb	0	19	34

3.2 Feature extraction

Because of the limited amount of data we have for training our learning algorithms, we decided to extract features from the audio clip based on expert knowledge. We used magnitude-based and phase-based features, extracted from each audio clip, which are further discussed below. For the feature based models, the samples are converted into features, The size of the feature vectors for MFCC, GD, and CQT are 4914, 63488, and 21168, respectively. The *mfcc*, *delta*, and *cqt* functions within the Librosa python package were used to extract the features from audio clips. The function to extract modifed group delay cepstral coefficents was created

manually and is available on GitHub (along with the rest of our code for generating the data and results) at https://github.com/joej1997/SyntheticSpeech.

3.2.1 Mel-Frequency Cepstral Coefficients (MFCC):. MFCCs are coefficients that are computed from the cepstral representation of the Mel-spectrogram of an audio clip. MFCCs are the most common feature vectors used in speaker verification systems and understood to represent the vocal tract of a speaker as a filter in a source-filter model of speech generation [16, 17, 24, 36, 37]. This is a speech parameterization technique which considers the logarithmic perception of audio in both the amplitude and frequency modes. MFCCs are computed by transforming the log power spectrum of audio signal into mel-scale, i.e, transforming the frequency of the signal into logarithmic scale. The discrete cosine transform is applied to this mel-scaled log power spectrum and the first 13 coefficients are stored as MFCCs. We also computed delta and delta-delta coefficients to capture dynamic feature within a short time frame. Delta and delta-delta coefficients can be interpretated as first and second order derivatives of the feature and helps to capture short term temporal transitions in the features.

3.2.2 Constant-Q Transform Coefficients (CQT):. The CQT is a spectral representation of a signal in which the frequency is logarithmic. It is related to discrete Fourier transforms (DFT) but has a few important differences. The transform exhibits a reduction in frequency resolution with higher frequency bins, which is desirable for auditory applications. As with the case with MFCCs, logarithmic representation of frequency means the transform mirrors the human auditory system, whereby at lower-frequencies spectral resolution is better, whereas temporal resolution improves at higher frequencies [35]. We decided to use CQT because of its better generalization performance to unknown speech synthesis algorithms [24]. Again, delta and delta-delta coefficients were computed to be used as additional features for our learning algorithms.

3.2.3 Group Delay: Group delay is defined as the negative derivative of the phase of the Fourier transform of a signal. It is the time-domain delay of each frequency component of the signal, as a function of frequency [32]. It represents the spectral phase information as a feature in speaker recognition. Most speech synthesis and voice conversion systems use a simplified, minimum phase model which may introduce artefacts into the phase spectrum [36]. Group delay can be used to detect artefacts in the phase spectra of synthetic speech [36]. We decided to use this feature because phase-based features have proven to be useful in speaker verification systems [19]. For numerical computation of group delay of audio clip, following equation was used:

$$D = \mathrm{Re}(\frac{Y(w)}{X(w)}) \tag{1}$$

where X(w) and Y(w) are the Fourier transforms of the signal, x(n) and scaled signal, $n \cdot x(n)$, respectively [28].

3.3 Random forests

Developed in 2001 by Leo Breiman [4], random forests is a statistical learning algorithm which builds off previous research in decision trees, bagging, and randomly chosen covariates [2, 3, 5]. Decision trees continue to remain popular because of their flexibility and ease of interpretation, however, fully grown regression trees such as those used in random forests have extremely high variance. This high variance contributes to poor prediction on a test set because of the bias-variance decomposition of errors. To reduce the variance, many trees are grown in parallel and the final class is chosen with majority voting. The success of random forests is dependent on how independent the individual trees are from eachother and the strength of the individual trees [2, 4]. To induce independence, trees are grown on via random samples of the training set, and splits at each node are chosen from a random subset of explanatory variables. Though random forests are more favorable for classification problems, they are also widely used in regression.

We use random forests as it has shown some success in previous research in detecting fake audio and generalizing to unknown attacks [24]. For our experiments, we use the *RandomForestClassifier* function in the sklearn package [22]. We use default hyperparameters.

3.4 Principal component analysis

Principal component analysis (PCA) is a method for finding a more "fundamental set of independent variables" among the current set of independent variables, especially under circumstances of correlation and measurement error [11, 21]. The principal components are obtained by minimizing the sum of squared perpendiculars between the observed points and the hyperplane of interest. Our purpose for using PCA is to reduce the dimensions of the input space of random forests before training. The number of PCs, or dimensions, we use is determined by the commonly used "elbow" method [20, 33], where the location between the end of the steep decrease and the beginning of the gradual decrease is found on the plot of variance explained and principal component number. An example of the "elbow" is demonstrated in Figure 1. In all cases, we first center our data by subtracting by the column means of the training set and dividing by the column standard deviations of the training set. We then find that 50 principal components is an appropriate number to use based on Figure 1. PCA has been shown to be useful in applications to synthetic speech detection [37].

3.5 Deep learning

We test a variety of deep learning approaches for ASV using an end-to-end model called Time-domain Synthetic Speech Detection Net (TSSDnet) [12], and multiple feature-based models. All the deep learning model we test are based on the ResNet architecture described in Hua et al. [12]. The base ResNet model starts with a convolutional layer followed by a max pooling layer. This is then followed by four ResNet style modules followed by an average pooling layer. The final layers consist of three fully connected linear layers. TSSDnet directly takes in raw waveforms in the time domain and uses 1D convolutions while the feature-based models uses the same architecture but take in GD, MFCC, or constant-Q feature tensors and uses 2D convolutions.

3.5.1 Loss function. We use a data augmentation technique called mixup regularization [10], which was used in the TSSDnet paper to increase resilience to unknown attacks [12]. Mixup regularization creates new training examples as a convex combination of existing

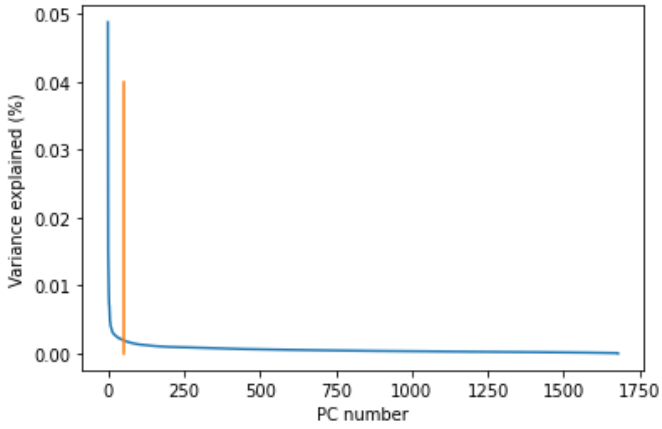

Figure 1: The blue line shows the % of variance explained for the principal component. The orange line shows our cutoff before which we keep the data, and after which we discard the data.

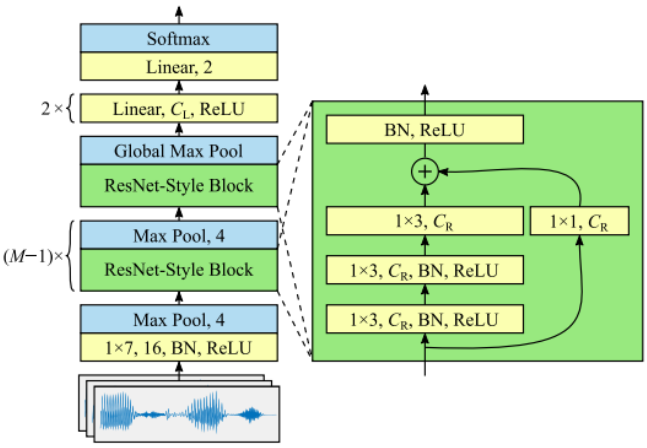

Figure 2: ResNet end-to-end ASV model architecture.

training examples and labels. The examples are combined as shown in eq:2.

$$\tilde{x_{ij}} = \lambda x_i + (1-\lambda)x_j$$
$$\tilde{y_{ij}} = \lambda y_i + (1-\lambda)y_j \quad (2)$$

Where x is the training sample and y is the label which is either 0 or 1.

The loss function uses the standard cross entropy loss, defined as eq:3

$$\mathrm{CE}(y,p) = (-y\log(p) - (1-y)\log(1-p)) \quad (3)$$

Where p is the probability predicted on the training example x after softmax.

The resulting loss function is a combination of cross entropy losses eq:4

$$\mathrm{CE}_{\mathrm{mixup}}(y_i, y_j, \tilde{p_{ij}}) = \lambda CE(y_i, \tilde{p_{ij}}) + (1-\lambda)CE((y_j, \tilde{p_{ij}}) \quad (4)$$

where $\tilde{p}$ is the probability that is computed by the network on training example $\tilde{x_{ij}}$ after soft max.

3.5.2 Deep learning training details. TSSDnet and the MFCC ResNet use a batch size of 16, constant-Q net uses a batch size of 8, and GD net uses a batch size of 2. The different batch sizes are due to GPU memory constraints when dealing with larger feature vectors. MFCC can use a larger batch size because the size of its feature vector is smaller than GD or CQT. TSSDnet can use a larger batch size because it only uses 1D convolutions on raw audio rather than 2D convolutions. Each model is trained with a learning rate of 0.001 over 100 epochs using the Adam optimizer [1], which were the same training hyperparameters used by the TSSDnet paper [12].

4 EXPERIMENTS AND RESULTS

We present the main results of our experiments on random forests and deep learning with various features on the Tacotron 2 and FakeAVCeleb dataset.

4.1 Data partition and evaluation strategy

We evaluate the different models using total accuracy which is (correct predictions)/(total predictions). We will also report the confusion matrix in the form shown in eq:5:

$$\begin{pmatrix} \text{True positive} & \text{False positive} \\ \text{False negative} & \text{True negative} \end{pmatrix}. \quad (5)$$

We first trained our learning algorithms on training data from a specific person, and tested on different audio clips from the same speaker and the same audio synthesis algorithm. We thought this might be closer to our courtroom scenario to have limited training data from a specific person. Then, in the second stage, we combined training data from all three celebrities to train our models to test the performance of a general model trained on multiple speakers. For unknown attacks, the same data partition strategy was adopted for speakers but the test set contains samples from FakeAVCeleb dataset which was not used while training the models. The unknown attack test was not done for Trump due to the lack of testing data.

4.2 Random forest results

We first trained random forest models based on the data partition strategy described above. For training random forest models, we used a combined feature vector by concatenating all the features (MFCC, CQT and GD) for a single audio clip. The results for our random forest models for both known and unknown attacks are summarised in Table 2. As we can see from the table, random forest works well for known attacks whereas the performance drops significantly when tested on unknown attacks. For known attacks, model trained on a specific speaker performed better than the general speaker model. However, for unknown attacks, the general model performed better than the specific speaker models, but in both cases, the accuracy is very low.

Table 2: Test accuracies and confusion matrices (True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN)) for random forests.

Test Results: Random Forests					
	Accuracy [%]	TP	FP	FN	TN
Trump (Known)	88	68	9	11	92
Drake (Known)	87	41	4	8	45
Takei (Known)	84	68	9	11	92
General (Known)	79	107	75	15	230
Drake (Unknown)	33	0	89	6	47
Takei (Unknown)	65	10	47	3	86
General (Unknown)	70	60	86	31	214

4.3 Deep learning results

The deep learning models are based on the pytorch implementations of TSSDnet that are publicly available on GitHub https://github.com/ghuawhu/end-to-end-synthetic-speech-detection. All the results are generated on 3 Nvidia GeForce RTX 2080 Ti GPUs.

4.3.1 Known Attacks. For known attacks, we trained four different model types including, an end-to-end model, a GD model, a MFCC model, and a constant Q transform model. For each type of model, we trained them on Tacotron 2 data for each speaker and then for all speakers. This results in three specific speaker models and one general speaker model for each model type. The results of the tests with known attacks are given in table 3. Despite the generalization ability of end-to-end models demonstrated by Hua et al [12], the end-to-end model performs quite poorly compared to the feature models due to the limited amount of data. The best model overall is the GD, which has the highest general accuracy (98.6 percent) and higher specific speaker accuracy, beaten only by MFCC on Trump by only 1.1%.

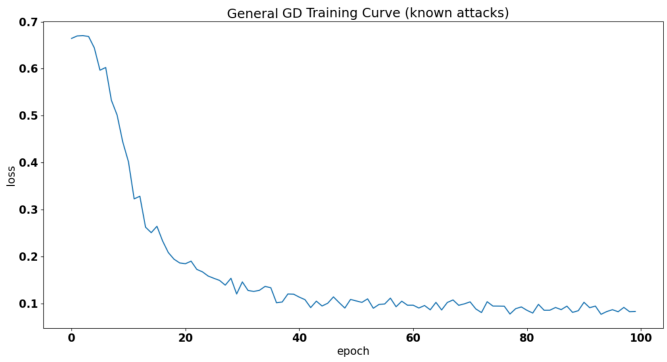

Figure 3: Training curve for general GD model for known attacks

4.3.2 Unknown attacks. For experiments with unknown attacks, we train our models on the same Tacotron 2 data, but we test against FakeAvCeleb, which we denote as our unknown attack. We did not have any Trump samples in FakeAvCeleb so we exclude Trump from these experiments. The results are given in Table 4. All the models suffer from the unseen attack due to having not been trained on FakeAvCeleb. Drake models in particular perform poorly. For example, GD on drake results in 100 percent accuracy in for known attacks, but only 42.3 percent accuracy for unknown attacks. GD performs the best in the general case and for Takei data, resulting in 79.4% and 94.5% accuracy. CQT performs better than GD on Drake, resulting in 66.2% accuracy while GD achieves 42.3% accuracy. Like in the known attacks experiments, end-to-end models performs poorly overall due to the limited dataset.

Table 3: Test accuracies and the confusion matrices (True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN)) for Neural Networks.

Test Results (Known Attacks): Neural Nets					
	Accuracy [%]	TP	FP	FN	TN
TSSDnet (Trump)	42.8	0	0	103	77
TSSDnet (Drake)	54.1	53	45	0	0
TSSDnet (Takei)	59.7	89	60	0	0
TSSDnet (general)	57.4	245	182	0	0
MFCC (Trump)	**100**	**103**	**0**	**0**	**77**
MFCC (Drake)	83.7	47	10	6	35
MFCC (Takei)	76.5	89	35	0	25
MFCC (general)	80.5	245	83	0	99
Constant Q (Trump)	99.4	103	1	0	76
Constant Q (Drake)	79.6	33	0	20	45
Constant Q (Takei)	91.9	89	12	0	48
Constant Q (general)	92	233	22	12	160
GD (Trump)	98.9	101	0	2	77
GD (Drake)	**100**	**53**	**0**	**0**	**45**
GD (Takei)	**94**	**81**	**1**	**9**	**59**
GD (general)	**98.6**	**242**	**3**	**3**	**176**

5 DISCUSSION

In this section, we highlight findings along with limitations of our current work and suggest possible considerations for future work for more robust and accurate synthetic speech detection.

5.1 Findings

Confirming with findings from previous studies [24], both random forests and neural networks with manual feature extraction showed good performance for seen synthesis algorithms for both specific and general speaker models. The accuracy also dropped for unseen algorithms for both random forests and neural networks similar to [24] but performance for unseen algorithms are comparatively worse possibly because of limited data. Phase information proved to be useful for synthetic speech detection for known synthesis algorithms, especially when used to train neural nets, but performance is highly speaker dependent for unknown algorithms. This feature also showed the best performance for unknown attacks. End-to-end neural network models showed no learning ability contrary to results in other works [12]. This was expected because we

Table 4: Test accuracies and confusion matrices (True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN)) for Neural Networks.

Test Results (Unknown Attacks): Neural Nets					
	Accuracy [%]	TP	FP	FN	TN
TSSDnet (Drake)	37.3	53	89	0	0
TSSDnet (Takei)	61	89	57	0	0
TSSDnet (general)	62.7	245	146	0	0
MFCC (Drake)	37.3	53	89	0	0
MFCC (Takei)	56.2	82	57	7	0
MFCC (general)	63.2	245	144	0	2
Constant Q (Drake)	**66.2**	**20**	**15**	**33**	**74**
Constant Q (Takei)	76.7	89	34	0	23
Constant Q (general)	62.9	219	119	26	27
GD (Drake)	42.3	53	82	0	7
GD (Takei)	**94.5**	**84**	**3**	**5**	**54**
GD (general)	**79.8**	**239**	**73**	**6**	**73**

fed the model much less data samples than is usually required for deep neural networks to learn features from raw data.

5.2 Limitations

5.2.1 Limited Data: We had a limited amount of data samples for our training in comparison to most other synthetic speech detection datasets. Our data was also limited in variation in speech synthesis algorithms. Further, only male celebrities were considered in this study.

5.2.2 No Hyperparameter tuning: We used the default hyperparameters while training our random forest and neural network models. Our focus was on studying how performance would differ by training on single or multiple speakers, thus raw performance was not our goal. Hence, we did not tune the hyperparameters of our models which could have demonstrated better raw accuracies.

5.3 Future Works

5.3.1 Data: Future synthetic speech detection systems should be composed of more data while also being comprehensive with respect to different genders, ethnicities, accents, and languages. Data augmentation can also be used to increase training samples as it has been proven to increase robustness and generalizability [31]. A larger variety of speech synthesis algorithms for both training and testing would be appropriate for future studies. Training datasets for specific people or celebrities with large availability of audio recordings can also be created to explore the speaker dependency aspect of synthetic speech detection.

5.3.2 Explore other learning techniques: We only used random forests and the TSSDnet neural network architecture. Future works should explore other learning algorithms and network architectures while also tuning hyperparameters to get better performance. For example, for our courtroom scenario of detecting fake audio from a specific person, transfer learning could be tested by training a single speaker classifier fed by a general model trained on multiple speakers. There has been some initial research on the usefulness of transfer learning, but more is needed [29]. Furthermore, other audio features such as long term temporal features of an audio clip should be explored in conjunction with the features we presented [36, 37]. As multi-modal data has been proven to show better performance[17], emotion correlation in speech audio and corresponding text could also provide interesting and useful information.

5.3.3 Other potential applications: Although our courtroom scenario presents a critical area of application of such synthetic speech detection systems, the dire consequence of erroneous results in life or death scenarios could mean that these systems will be deemed not ready for implementation in such cases. One potential area of application where we believe the technology acceptance could face less friction is a web browser plugin to detect fake audio content on the web. The plugin can be used to to prevent users from unknowingly consuming misinformation by warning them that the audio content could be synthetic or manipulated.

6 CONCLUSION

We explored possible machine learning algorithms for real and fake audio classification for a specific person, with a courtroom perspective. Random forests with a combination of three feature vectors worked well for known attacks but the accuracy drops significantly for unknown attacks. For known attacks, random forests works better for models trained with specific people whereas the performance is better in the general model for unknown attacks.

With the limited data we have, end-to-end neural network models do not work well. However, any of MFCC, GD, or CQT features makes the accuracy of neural networks much better with GD showing the best performance. Neural network models show similar or better performance for the general model when compared with the specific speaker model. We believe this is because neural networks usually require a large database of training data. Neural network models perform better than random forests for both known and unknown attacks but the performance for unknown attacks is still worse than known, with exception of Takei data trained with GD features. This speaker dependency could also be an interesting phenomenon to be explored further.

We hope our study will be helpful for others trying to develop more robust audio deepfake detection systems for person-specific use cases such as a courtroom scenario.

REFERENCES

[1] 2014. Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980 cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[2] Yali Amit and Donald Geman. 1997. Shape quantization and recognition with randomized trees. *Neural computation* 9, 7 (1997), 1545–1588.

[3] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[4] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[5] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. 1984. Classification and regression trees. Wadsworth Int. *Group* 37, 15 (1984), 237–251.

[6] Xinhui Chen, You Zhang, Ge Zhu, and Zhiyao Duan. 2021. UR channel-robust synthetic speech detection system for ASVspoof 2021. *arXiv preprint arXiv:2107.12018* (2021).

[7] Hasan Fayyad-Kazan, Ale Hejase, Imad Moukadem, Sondos Kassem-Moussa, et al. 2021. Verifying the Audio Evidence to Assist Forensic Investigation. *Computer and Information Science* 14, 3 (2021), 1–25.

[8] Yang Gao, Jiachen Lian, Bhiksha Raj, and Rita Singh. 2021. Detection and Evaluation of human and machine generated speech in spoofing attacks on automatic speaker verification systems. In *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 544–551.

[9] Yang Gao, Tyler Vuong, Mahsa Elyasi, Gaurav Bharaj, and Rita Singh. 2021. Generalized Spoofing Detection Inspired from Audio Generation Artifacts. *arXiv preprint arXiv:2104.04111* (2021).

[10] Yann N. Dauphin David Lopez-Paz Hongyi Zhang, Moustapha Cisse. 2018. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=r1Ddp1-Rb

[11] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology* 24, 6 (1933), 417.

[12] Guang Hua, Andrew Beng Jin Teoh, and Haijian Zhang. 2021. Towards End-to-End Synthetic Speech Detection. *IEEE Signal Processing Letters* 28 (2021), 1265–1269. https://doi.org/10.1109/LSP.2021.3089437

[13] S.E. Jenkins. [n. d.]. LCSO: Recorded 'confession' in Kendrick Johnson case fake. *WTXL Tallahassee* ([n. d.]). https://www.wtxl.com/news/local-news/lcso-recorded-confession-in-kendrick-johnson-case-fake

[14] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558* (2018).

[15] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. 2021. AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks. *arXiv preprint arXiv:2110.01200* (2021).

[16] Hasam Khalid, Minha Kim, Shahroz Tariq, and Simon S Woo. 2021. Evaluation of an Audio-Video Multimodal Deepfake Dataset using Unimodal and Multimodal Detectors. In *Proceedings of the 1st Workshop on Synthetic Multimedia-Audiovisual Deepfake Generation and Detection*. 7–15.

[17] Hasam Khalid, Shahroz Tariq, Minha Kim, and Simon S Woo. 2021. FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset. *arXiv preprint arXiv:2108.05080* (2021).

[18] Pavel Korshunov and Sébastien Marcel. 2018. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685* (2018).

[19] Hema Murthy, Padmanabhan R., and Parthasarathi S.H.K. 2009. Robustness of phase based features for speaker recognition. , 2355 - 2358 pages.

[20] Ganesh R Naik, Suviseshamuthu Easter Selvan, Massimiliano Gobbo, Amit Acharyya, and Hung T Nguyen. 2016. Principal component analysis applied to surface electromyography: a comprehensive review. *IEEE Access* 4 (2016), 4025–4037.

[21] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2, 11 (1901), 559–572.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[23] Ricardo Reimao and Vassilios Tzerpos. 2019. FoR: A dataset for synthetic speech detection. In *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 1–10.

[24] Ricardo Amaral Martins Reimao. 2019. Synthetic speech detection using deep neural networks. (2019).

[25] Jessa Schroeder. [n. d.]. Horrifying death threats and abuse UFC fighter Rachael Ostovich endured 'at the hands of her husband' is revealed in secret audio recording by a witness who heard her screams. *Daily Mail* ([n. d.]). https://www.dailymail.co.uk/news/article-6521737/Audio-reveals-extend-abuse-Rachael-Ostovich-endured-husband.html

[26] Muhammad Shahzad. [n. d.]. Audio clip of Ayesha, Rambo 'discussing extortion from suspects' surfaces. *Tribune Pakistan* ([n. d.]). https://tribune.com.pk/story/2324284/audio-clip-of-ayesha-rambo-discussing-extortion-from-suspects-surfaces

[27] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4779–4783.

[28] Julius O. Smith. accessed (date accessed). *Introduction to Digital Filters with Audio Applications*. `http://ccrma.stanford.edu/~jos/filters/`. online book.

[29] Nishant Subramani and Delip Rao. 2020. Learning efficient representations for fake speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5859–5866.

[30] Hemlata Tak, Jee-weon Jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans. 2021. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. *arXiv preprint arXiv:2107.12710* (2021).

[31] Hemlata Tak, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans. 2021. RawBoost: A Raw Data Boosting and Augmentation Method applied to Automatic Speaker Verification Anti-Spoofing. *arXiv preprint arXiv:2111.04433* (2021).

[32] Tharmarajah Thiruvaran, Eliathamby Ambikairajah, and Julien Epps. 2007. Group delay features for speaker recognition. In *2007 6th International Conference on Information, Communications Signal Processing*. 1–5. https://doi.org/10.1109/ICICS.2007.4449768

[33] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. 2003. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*. Springer, 91–109.

[34] RLMAPC Wijethunga, DMK Matheesha, Abdullah Al Noman, KHVTA De Silva, Muditha Tissera, and Lakmal Rupasinghe. 2020. Deepfake Audio Detection: A Deep Learning Based Solution for Group Conversations. In *2020 2nd International Conference on Advancements in Computing (ICAC)*, Vol. 1. IEEE, 192–197.

[35] Wikipedia contributors. 2021. Constant-Q transform — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Constant-Q_transform&oldid=1050196207 [Online; accessed 12-December-2021].

[36] Zhizheng Wu, Phillip L De Leon, Cenk Demiroglu, Ali Khodabakhsh, Simon King, Zhen-Hua Ling, Daisuke Saito, Bryan Stewart, Tomoki Toda, Mirjam Wester, et al. 2016. Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24, 4 (2016), 768–783.

[37] Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. 2013. Synthetic speech detection using temporal modulation feature. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 7234–7238.

[38] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. 2021. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. *arXiv preprint arXiv:2109.00537* (2021).

[39] Zhenyu Zhang, Xiaowei Yi, and Xianfeng Zhao. 2021. Fake Speech Detection Using Residual Network with Transformer Encoder. In *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*. 13–22.