Synthetic Speech Detection Using Neural Networks

Ricardo Reimao
Electrical Engineering and Computer Science
York University
Toronto, Canada
rreimao@yorku.ca

Vassilios Tzerpos
Electrical Engineering and Computer Science
York University
Toronto, Canada
bil@yorku.ca

***Abstract*—Computer generated speech has improved drastically due to advancements in voice synthesis using deep learning techniques. The latest speech synthesizers achieve such high level of naturalness that humans have difficulty distinguishing real speech from computer generated speech. These technologies allow any person to train a synthesizer with a target voice, creating a model that is able to reproduce someone's voice with high fidelity. This technology can be used in several legit commercial applications (e.g. call centres) as well as criminal activities, such as the impersonation of someone's voice.**

In this paper, we analyze how synthetic speech is generated and propose deep learning methodologies to detect such synthesized utterances. Using a large dataset containing both synthetic and real speech, we analyzed the performance of the latest deep learning models in the classification of such utterances. Our proposed model achieves up to 92.00% accuracy in detecting unseen synthetic speech, which is a significant improvement from human performance (65.7%).

***Index Terms*—synthetic speech detection, deep neural networks, machine learning, text to speech**

I. Introduction

Synthetic speech refers to any utterance generated by a computer. With the advent of deep learning, synthetic speech is getting closer to a natural sounding voice. Some of the state-of-art technologies achieve such a high level of naturalness that humans have difficulty distinguishing real speech from computer generated speech. Moreover, these technologies allow a person to train a speech synthesizer with a target voice, creating a model that is able to reproduce someone's voice with high fidelity.

Such technologies can have negative consequences, since one could maliciously impersonate someone's voice. An example would be training a model with the voice of a famous person and then using this model to generate an utterance with malicious content to defame the person publicly.

During our experiments, we surveyed human subjects to understand their susceptibility to impersonation using speech generated by the latest deep learning technologies. Our results show that, on average, one in three synthetic speech utterances were perceived as real by our participants. This reinforces the importance of a system that is able to distinguish between human- and computer-generated speech.

In this paper, we analyze how synthetic speech is generated and propose approaches to detect such synthesized utterances.

The first step toward training a synthetic speech detection system is to collect a significant amount of computer-generated speech as well as real human speech. For our research we created a dataset, called *Fake or Real (FoR)*, which contains more than 84,000 synthetic utterances, as well as more than 111,000 real utterances [1]. The FoR dataset is publicly available to the community (http://bil.eecs.yorku.ca/datasets) under the GNU GPLv3 license.

Next, we developed several potential solutions for a synthetic speech detection classifier. Although previous research has achieved very good results in the past [2], [3], they did not include the latest state-of-art machine-learning TTS systems, such as DeepVoice 3 [4] and Google Wavenet [5].

We achieve up to 92.00% accuracy on unseen utterances using a deep-learning model, which is considerably higher than using non-deep-learning methods (86.94%) and significantly higher than average human accuracy (64.83%). These results show that deep learning models can be a good solution for the synthetic speech detection problem.

The remainder of this paper is organized as follows: Section II presents an overview of the FoR dataset. In Section III, we present a series of experiments with the objective of analyzing and identifying potential models for synthetic speech detection. Finally, Section IV concludes the paper.

II. The FoR Dataset

Although several synthetic speech datasets were proposed in the past [2], [3], [6], they do not focus on synthetic speech generated by the latest deep learning-based speech synthesis algorithms. Moreover, the number of utterances in previously published datasets is typically not sufficient to train complex neural network models [7].

For our experiments, we created a new synthetic speech dataset. The *Fake or Real (FoR)* dataset is composed of more than 87,000 synthetic utterances and 111,000 real utterances from a large variety of individuals. It contains utterances from state-of-the-art speech synthesis algorithms, i.e. utterances with naturalness similar to real human speech. Also, our dataset contains a large number of data points and, according to our experiments, is sufficiently large to train complex models, such as InceptionV3 [8], without overfitting.

In this paper. we use the following two versions of the FoR dataset: a) *for-2seconds*. This version is balanced in terms of gender and class and normalized in terms of sample rate,

volume and number of channels. All files are truncated at 2 seconds, b) *for-rerecorded*. A re-recorded version of the *for-2seconds* version, to simulate a scenario where an attacker sends an utterance through a voice channel (i.e. a phone call or a voice message).

More information about the collection and processing methods can be found at the FoR dataset publication [1].

III. Experiments

With the dataset completed, the next step consists in performing experiments to compare ways of detecting synthetic speech. We start this section by presenting the performance of humans in the synthetic speech detection task. In Section III-B we present a series of experiments to analyze the accuracy of several detection methods against the for-2second dataset. Also, we present an analysis of a real-world attack scenario using the re-recorded dataset (for-rerecorded). For both dataset versions we also present the performance of the proposed models on a totally unseen TTS algorithm.

A. Synthetic Speech Detection by Humans

As one of our goals is to present a synthetic speech detection technique that performs better than humans, the first step is to analyze the human performance in this task. Similar to a Turing test, the idea is to play utterances and ask participants to judge if the speech was generated by a computer or not. Ten synthetic utterances and ten real utterances were randomly selected from the FoR dataset for this task. A total of 29 participants were asked to listen to each audio just once using their own devices and select the options "Fake" or "Real" according to their guess. To ensure privacy, no personal information was collected.

After the period of three weeks, the results were gathered, compiled and analyzed. The average overall human accuracy was 64.83%, 60.34% for synthetic speech and 69.31% for real speech.

The numbers on this survey show that, on average, humans would miss 1 out of 3 synthetic utterances. If considering only high-performance algorithms (such as Microsoft TTS and Amazon Polly), humans mistake synthetic for real about half the time. This shows that the human perception of synthetic speech is vulnerable to the latest speech synthesizers and that an automated synthetic speech detector is needed.

B. Synthetic Speech Detection

With the human performance evaluated, the next step is to propose and analyze the performance of a variety of synthetic speech detection methodologies. For these experiments, we use the for-2seconds version of the dataset to understand the performance of both frequency analysis and deep learning approaches.

The frequency analysis experiments consist in using traditional machine learning techniques to classify utterances on the FoR dataset. This creates a baseline to compare the traditional machine learning methods with the deep learning approaches. These experiments are based on frequency analysis, i.e. the extraction of a frequency representation and classification using machine learning techniques.

The process for this experiment consists in extracting an audio representation (such as an STFT matrix) for each audio file, averaging the representation over time to obtain a frequency-activation vector, inputing this vector into Weka with the appropriate classes (synthetic/real) and comparing the results of the several machine learning algorithms. Even though during the averaging process the temporal information is lost, this technique is still valid for several audio classification problems. The following audio representations were chosen for this experiment: Fast Fourier Transform (FFT), Short-time Fourier Transform (STFT), Mel-Spectrograms, Mel-frequency Cepstral Coefficients (MFCC), and Constant-Q Transform (CQT).

The audio files were processed using the Librosa audio processing library, which was used to generate audio representations in the following formats: STFT (128 frequency bins), STFT (1024 frequency bins), FFT (1024 frequency bins), Mel-Spectrograms (128 frequency bins), Mel-Spectrograms (1024 frequency bins), MFCC (128 coefficients) and Constant-Q Transform (1008 frequency bins). The matrix-shaped audio representations (STFT, Mel-Spectrograms, MFCC and CQT) were then averaged on a horizontal axis, meaning, the frequency features were averaged over time. This results in one vector for each utterance.

FoR-2seconds (validation)							
Algorithm	**STFT 128**	**STFT 1024**	**FFT1024**	**Mel 128**	**Mel 1024**	**MFCC 128**	**CQT 1008**
Naïve Bayes	61.32%	65.95%	58.45%	62.84%	65.14%	79.22%	60.40%
SVM	56.15%	57.11%	57.25%	78.27%	80.53%	92.07%	84.32%
Decision Tree (J48)	77.14%	72.64%	70.98%	91.93%	93.17%	93.20%	86.37%
Random Forests	82.55%	80.96%	76.96%	96.81%	96.60%	98.54%	94.01%

Fig. 1. Frequency Analysis Accuracy - FoR 2 Seconds

Figure 1 shows the results of the frequency analysis experiments, in which it is possible to observe that using the MFCC audio representation with the Random Forests method achieves up to 98.54% accuracy on the validation dataset. This shows that even though frequency analysis may not be the best method, it is possible to achieve high accuracy using only frequency information.

To better understand the classification accuracy and the differences in the frequency spectrum, we decided to investigate which frequency ranges are more important for the classification task. We used two attribute ranking methods, Chi-Square [9] and Information Gain [10], to generate a *Frequency Classification Activation Map (FCAM)*.

Using the STFT (1024 bins) audio representation (due to its frequency bin linearity) we utilized the Weka tool to calculate the values of Chi-Square and Information gain for each of the frequency bins. To help with the visualization of the results, the frequencies were ordered (from 0Hz to 8kHz) and a colour was attributed to each frequency: red designating high importance for classification, green designating low importance for classification. The resulting frequency classification activation map can be seen in Figure 2.

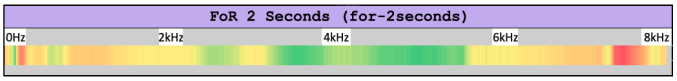

Fig. 2. Frequency Classification Activation Map - FoR 2 Seconds

The results of this experiment show that high frequencies (above 7.2kHz) are the most important frequencies to distinguish real speech from synthetic speech in its original form.

After performing the frequency analysis experiments, we used a series of deep learning techniques for synthetic speech detection. Following previous literature, we translate the audio classification problem into an image classification problem by using visual audio representations (i.e. spectrograms). This conversion is useful since the majority of the deep learning models available are designed for image classification.

The deep learning experiments consist of extracting audio features (STFT, Mel-Spectrograms, MFCC and CQT) from the FoR dataset (for-2seconds) and converting it to an image. This process was done using a custom script and the Librosa library. The resulting images were then used to train nine selected architectures for a maximum of 50 epochs (early stop if accuracy improvements were not seen in the last 10 iterations). The selected architectures are:

- 4-Layer fully connected neural network
- 2-Layer CNN with two extra fully connected layers
- 3-Layer CNN with two extra fully connected layers
- VGG16 [11] using the ImageNet weights and re-training only the last 5 layers
- VGG19 [11] using the ImageNet weights and re-training only the last 5 layers
- InceptionV3 [8] using the ImageNet weights and re-training only the last 2 inception blocks (249 layers)
- ResNet [12] using the ImageNet weights and re-training all the layers
- MobileNet [13] using the ImageNet weights and re-training all the layers
- XceptionNet [14] using the ImageNet weights and only re-training the last layer

The results can be seen in Figure 3, where it is possible to observe that the VGG16 and VGG19 models with STFT audio representations presented the highest validation accuracy (99.96% and 99.94% respectively). As a side finding, we noted that simpler models (such as 4-layer fully connected) were not able to learn at all.

One interesting technique in deep learning is the generation of Classification Activation Maps (CAMs). These maps show which areas of an image are more important for the classification task. To create a general activation map, we generated the individual CAM for each spectrogram in the dataset and averaged the results into one CAM. The *Average Classification Activation Map (ACAM)* for real and synthetic speech of the for-2seconds dataset (VGG19 model, STFT audio representation) can be seen in Figure 4.

From the average CAM it is possible to note that the higher frequencies are the most critical area for classification,

FoR-2seconds (validation)				
Algorithm	**STFT 1024**	**Mel 128**	**MFCC 128**	**CQT**
4-Layer Fully Connected	49.18%	46.14%	49.95%	49.98%
2-Layer CNN (+2FC)	47.56%	41.57%	90.48%	49.99%
3-Layer CNN (+2FC)	50.10%	38.29%	92.46%	50.01%
VGG16	99.96%	95.21%	98.88%	97.64%
VGG19	99.94%	96.42%	98.72%	97.58%
InceptionV3	99.88%	79.94%	91.97%	94.19%
ResNet	69.51%	78.98%	88.62%	75.68%
MobileNet	99.21%	94.36%	98.23%	96.89%
XceptionNet	98.64%	74.69%	80.80%	84.79%

Fig. 3. Deep Learning Accuracy - FoR 2 Seconds

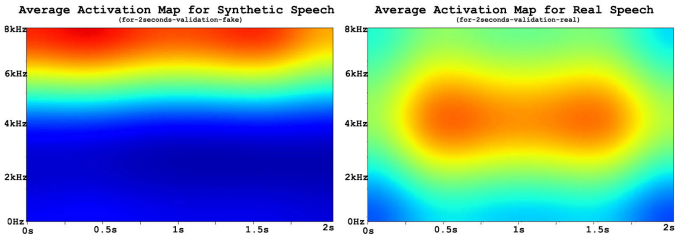

Fig. 4. Average CAM for Real and Synthetic audio

especially for synthetic utterances. This may indicate that synthetic speech generates audio mostly on the frequencies related to speech, while real audio may contain data in higher frequencies due to background noise or recording noises. Also, due to the fact that synthetic speech is mostly generated at 16kHz sample rate, the signal amplitude in frequencies close to 8kHz is low. This matches with the results of the frequency analysis experiments, which showed that high frequencies play an important role for synthetic speech detection.

Although the deep learning techniques presented the highest accuracy (99.96%), the frequency analysis methodologies also presented good results (98.54%). This indicates that there is a clear distinction between the frequency spectrum of real speech and synthesized speech.

C. Unseen Algorithm - Synthetic Speech Detection

To evaluate the generalization ability of the proposed models and to evaluate if frequency discrepancies are sufficient to detect synthetic speech, we use the same methodology to test the performance of the models trained in Section III-B against the testing part of the dataset, which is a totally unseen TTS algorithm (Google TTS Wavenet, not included in the training/validation dataset). This experiment also simulates how the models would react if an attacker creates a new TTS system.

The first step is to observe the performance of frequency analysis methodologies against the unseen algorithm. Following the same process as the previous experiments, we generate the audio representations for the testing dataset and use the previously trained Weka models to classify the utterances of the unseen algorithm.

Figure 5 shows the result of this analysis. One interesting point to note is that the top performer audio representation

FoR-2seconds (testing)							
Algorithm	STFT 128	STFT 1024	FFT1024	Mel 128	Mel 1024	MFCC 128	CQT 1008
Naïve Bayes	52.38%	53.21%	44.94%	54.77%	54.59%	48.25%	59.00%
SVM	48.89%	50.00%	47.79%	79.50%	79.59%	76.01%	83.91%
Decision Tree (J48)	55.14%	57.99%	55.51%	63.51%	65.62%	55.79%	73.71%
Random Forests	53.21%	59.46%	48.16%	83.36%	82.07%	56.98%	86.94%

Fig. 5. Frequency Analysis Accuracy - Unseen Algorithm

(MFCC) from the last experiment (Section III-B) had its accuracy drastically reduced. This may indicate that for unseen algorithms, MFCCs may not be the best representation. However, the CQT audio representation presented good performance (94.01%) in the first experiment (Section III-B) and now presents the best performance (86.94%).

The frequency classification activation map (using Chi-Square and STFT) was generated for the testing dataset. The frequency classification regions are similar to the ones in the original experiment (Section III-B), meaning that even though it is a new algorithm, the biggest differences between synthetic audio and real audio are still on the high frequencies.

FoR-2seconds (testing - mixed)							
Algorithm	STFT 128	STFT 1024	FFT1024	Mel 128	Mel 1024	MFCC 128	CQT 1008
Naïve Bayes	52.38%	53.67%	46.04%	56.25%	56.70%	51.65%	59.46%
SVM	50.27%	50.27%	50.45%	81.15%	81.70%	93.65%	87.59%
Decision Tree (J48)	68.38%	66.54%	62.68%	87.40%	88.87%	90.62%	80.14%
Random Forests	73.34%	69.02%	59.92%	91.81%	93.65%	99.17%	87.40%

Fig. 6. Frequency Analysis Accuracy - Learning a New Algorithm

To investigate if the models would be able to learn a new algorithm, we added 200 utterances (approximately 7 minutes) of the testing dataset into the training dataset and re-evaluated the testing dataset. The results can be seen in Figure 6, which shows that the models were able to adapt to the new TTS algorithm and deliver high accuracy (99.17%).

To verify if deep learning algorithms perform better than frequency analysis on unseen algorithms, we used the same deep learning analysis process as the previous experiments on the testing part of the dataset. We used the deep learning models trained in Section III-B to classify the Google Wavenet TTS utterances.

FoR-2seconds (testing)				
Algorithm	STFT 1024	Mel 128	MFCC 128	CQT
4-Layer Fully Connected	50.00%	50.00%	50.00%	50.00%
2-Layer CNN (+2FC)	50.00%	62.59%	72.33%	50.00%
3-Layer CNN (+2FC)	50.00%	71.14%	74.54%	52.25%
VGG16	50.00%	88.33%	66.18%	89.25%
VGG19	52.02%	87.78%	77.30%	90.72%
InceptionV3	52.76%	69.12%	70.68%	86.95%
ResNet	50.74%	83.55%	87.87%	75.83%
MobileNet	54.14%	83.73%	74.91%	92.00%
XceptionNet	68.57%	57.26%	59.10%	74.82%

Fig. 7. Deep Learning Accuracy - Unseen Algorithm

Figure 7 shows the results of this experiment. Similarly to the Frequency Analysis methodology, the accuracy also decreased but stayed significantly high (92.00% using CQT and MobileNet). One interesting observation is that STFT, which in the previous experiment (Section III-B) was the top performer, is now the audio representation with lowest accuracy. This may indicate that in unseen cases, CQT audio representation is the best option to be adopted.

To investigate if the models would be able to learn a new algorithm, we added 200 utterances (approximately 7 minutes) of the testing dataset into the training dataset. Then, using the audio representation that had the biggest impact by the unseen synthetic voice (STFT), we re-trained all the models to evaluate the testing accuracy. The results show a considerable improvement in the testing accuracy, achieving up to 99.82% (RESNET) and 98.99% (VGG19). This shows that although the proposed deep learning models had a significant decrease in performance with a fully unseen algorithm, only a small amount of data is required from the new TTS system to regain the original performance on the deep learning models.

In this experiment it is possible to observe that the behaviour of deep learning models was very similar to the frequency analysis methodologies. The accuracy drops with an unseen TTS algorithm, but it is regained when new utterances are added to the training data. However, throughout the whole experiment, the accuracy presented by the top performing deep learning methodology is higher than the one presented by the frequency analysis methodologies.

D. Rerecorded Synthetic Speech Detection

To evaluate the efficiency of the aforementioned detection approaches in a real-world scenario where an attacker plays a synthetic utterance through a voice channel, we apply them to the re-recorded dataset (for-rerecorded). This experiment is also important to test the accuracy of our models with a dataset where the differences in the high frequencies are reduced due to the rerecording process.

Using the same process as in previous experiments, we generated the frequency analysis results for the for-rerecorded dataset. The results from Weka are presented in Figure 8, which shows that the highest performance (95.05%) is with the MFCC audio representation and Random Forests, the same as the for-2second results (Section III-B). It is also possible to note a small drop in accuracy when compared to the for-2seconds result (from 98.54% to 95.05%). The presented numbers indicate that even though the frequencies were more uniform, the algorithms were still able to identify the reduced frequency differences.

FoR-rerecorded (validation)							
Algorithm	STFT 128	STFT 1024	FFT1024	Mel 128	Mel 1024	MFCC 128	CQT 1008
Naïve Bayes	61.98%	65.24%	62.03%	60.16%	60.20%	79.01%	69.11%
SVM	56.77%	53.23%	52.27%	77.18%	79.09%	81.14%	76.60%
Decision Tree (J48)	60.73%	63.81%	60.78%	86.09%	83.68%	87.03%	75.84%
Random Forests	66.97%	70.49%	66.97%	93.44%	92.11%	95.05%	90.55%

Fig. 8. Frequency Analysis Accuracy - FoR Rerecorded

To better understand the decline in the accuracy, a binary classification experiment was conducted with original synthetic utterances and re-recorded synthetic utterances. Then, we used traditional frequency analysis techniques, such as Naive Bayes and Random Forests, to perform the classification. The result was a classification accuracy of 99.86% (using

random forests), showing that there are significant differences between original and re-recorded audio.

The main hypothesis for the decline in the accuracy is related to the fact that the re-recording process reduces the frequency discrepancies between synthetic and real speech, especially in high frequencies.

Using the same process presented in Section III-B, we analyzed the impact of speech re-recording for deep learning models. The idea is to test if deep neural networks are able to distinguish real and synthetic utterances in re-recorded audio.

Using the re-recorded dataset presented in Section III-D (for-rerecorded), we trained and evaluated the deep learning models selected for this research. Figure 9 shows the result of this analysis, where it is possible to observe that the highest validation accuracy (VGG19 and STFT, 99.63%) is similar to the highest accuracy on the for-2second dataset (VGG16 and STFT, 99.96%). This shows that the re-recording process had almost no impact on the performance of the deep learning methodologies.

FoR-rerecorded (validation)				
Algorithm	**STFT 1024**	**Mel 128**	**MFCC 128**	**CQT**
4-Layer Fully Connected	48.87%	51.12%	49.08%	51.11%
2-Layer CNN (+2FC)	50.91%	49.11%	50.93%	51.10%
3-Layer CNN (+2FC)	50.94%	86.47%	88.98%	50.96%
VGG16	99.61%	91.60%	97.46%	92.70%
VGG19	99.63%	90.15%	96.75%	95.15%
InceptionV3	96.30%	70.64%	83.47%	87.11%
ResNet	65.49%	74.50%	75.43%	79.32%
MobileNet	98.90%	91.57%	93.93%	91.87%
XceptionNet	94.04%	69.81%	74.31%	75.89%

Fig. 9. Deep Learning Accuracy - FoR Rerecorded

To better understand what is being learned by the model, the ACAMs were generated for both synthetic and real audio (using STFT and VGG19), shown in Figure 10. The interesting point is that the model now presents a smoother classification area, showing that the discrepancies in the higher frequencies are not as drastic as in the for-2second dataset.

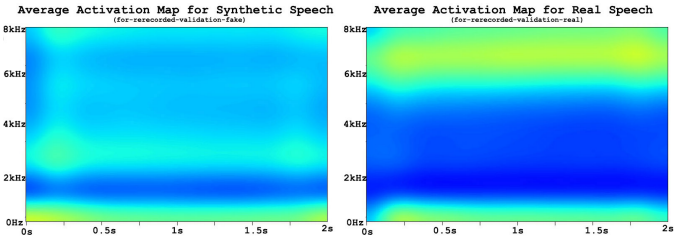

Fig. 10. Averaged Class Activation Maps (ACAMs) on Re-recorded Dataset

Since in the rerecorded dataset the frequency spectrum is more uniform between real and synthetic speech, the frequency based approaches were more impacted by the rerecording process, while the deep learning based methods were almost not impacted by the rerecording process and delivered an accuracy similar to the original for-2second dataset. This experiment shows the advantages of using deep learning techniques over frequency based ones.

E. Unseen Algorithm - Rerecorded Synthetic Speech Detection

To evaluate the generalization capabilities (through an unseen TTS algorithm) in a frequency-uniform dataset (through rerecording) we analyzed the performance of the proposed methodologies against the rerecorded version of the testing dataset. Following the same methodology as in the previous experiments, we generated the comparison between audio representations and classification models. The comparison can be seen in Figure 11.

FoR-rerecorded (testing)							
Algorithm	**STFT 128**	**STFT 1024**	**FFT1024**	**Mel 128**	**Mel 1024**	**MFCC 128**	**CQT 1008**
Naïve Bayes	56.25%	58.57%	56.98%	58.57%	57.35%	85.78%	82.35%
SVM	53.06%	53.06%	50.49%	75.61%	75.49%	68.62%	76.83%
Decision Tree (J48)	58.08%	55.02%	57.23%	70.95%	65.58%	65.44%	72.05%
Random Forests	60.66%	62.99%	61.88%	80.26%	84.43%	74.50%	85.17%

Fig. 11. Frequency Analysis Accuracy - Unseen FoR Rerecorded

As seen in Figure 11, the highest accuracy (85.78%, CQT and Random Forests) is considerably lower than the original for-2second dataset (98.54%, MFCC and Random Forests). This shows that the frequency analysis method is significantly impacted by the rerecording of an unseen TTS algorithm.

To better understand which frequencies are more relevant for the classification process, we generated the Frequency Classification Activation Map (FCAM) using STFT for the unseen utterances of the re-recorded dataset. The FCAM can be seen in Figure 12.

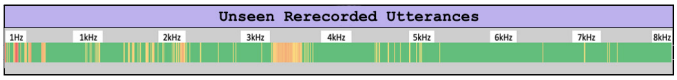

Fig. 12. Frequency Classification Activation Map - Unseen FoR Rerecorded

The FCAM shows a shift in the high-relevance classification area (red and orange areas of the FCAM) when compared to the FCAM presented for the for-2second dataset (in Section III-B). In this experiment, the high frequencies were not the main factor for classification, potentially due to the fact that the rerecording process reduces the discrepancies on high frequencies. Interestingly, the main classification area was shifted to low frequencies (around 140Hz), which may justify the decrease in the accuracy.

We performed similar experiments with the deep learning models. The highest accuracy across this experiment was 91.42% (CQT and VGG19), which is reduced when compared to the for-2second dataset but is still high (over 90%). This shows that although the deep learning algorithms were affected by the fact that the utterances were unseen and rerecorded, the best deep learning performer still performed well on the synthetic speech detection task. It is also interesting to note that the shift in the best accuracy for audio representation, from STFT to CQT, is the same shift observed in the non-rerecorded algorithm. This is one more piece of evidence that for an unseen TTS algorithm, CQT is the best performing audio representation.

To visualize what is being learned by the model, the ACAM for the VGG19 model (CQT audio representation) was

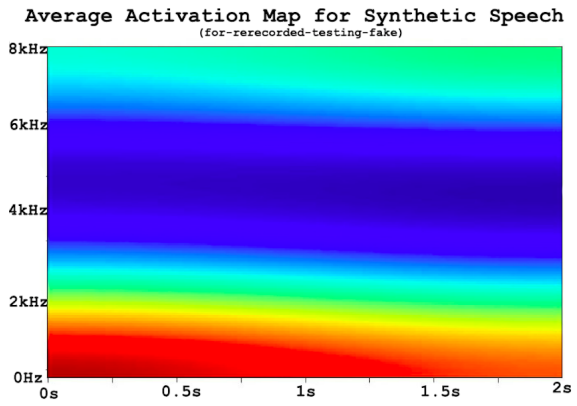

Fig. 13. Averaged Class Activation Maps (ACAMs) on Unseen Re-recorded Dataset

generated and can be seen in Figure 13. Similarly to what was observed in the frequency analysis, the main classification area is now the lower frequencies. This aligns with the previous observation that the rerecording process smooths out the high frequencies, which leads to a shift on the classification areas in the ACAM.

This experiment shows a clear distinction between the performance of frequency analysis and deep learning methodologies. While the best accuracy for frequency analysis had a decline of 12.76% in accuracy, the best accuracy for deep learning had a decline of only 8.54%. This means that deep learning models are 66.92% (8.54% over 12.76%) more effective than frequency analysis in the detection of unseen synthetic speech in a re-recorded environment.

It is possible to observe a significant drop in certain audio representations: MFCC for frequency analysis and STFT for deep learning. This confirms our theory that there is no generalized best audio representation for synthetic speech detection and each case should adopt its own appropriate audio representation. For deep learning approaches in seen data, the STFT audio representation is recommended, while for unseen data CQT is recommended. For frequency based approaches, MFCC is recommended for seen data while CQT is recommended for unseen data. It is also interesting to note that CQT presented the lowest averaged accuracy drop in all experiments, which may indicate that CQT is the most reliable audio representation if the type of the data (original/rerecorded, seen/unseen) is unknown.

Across all our frequency analysis experiments, it was possible to note that Random Forests presented the best performance in 3 out of 4 experiments (being only 0.61% behind in the unseen rerecorded experiment), meaning that it may be the best frequency based model for synthetic speech detection.

Across all deep learning experiments, the VGG19 model presented the best performance in 2 out of 4 experiments, being behind only 0.06% in the first experiment and 1.28% behind on the second experiment. This shows that VGG19 presents an overall good performance and may be the best classifier model for synthetic speech detection.

IV. Conclusion

As synthetic speech generation improves, the need for synthetic speech detection speech increases. With this work we hope to stimulate further research in synthetic speech detection. In our research, we were able to achieve a high level of accuracy for synthetic speech detection (90%+). Moreover, we demonstrate that deep-learning based techniques present higher accuracy across all our testing scenarios as well as being better at adapting to new TTS systems. This might indicates that such techniques are more suitable for the synthetic speech detection.

References

[1] R. Reimao and V. Tzerpos, "For: A dataset for synthetic speech detection," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Oct 2019, pp. 1–10.

[2] Z. Wu, P. L. D. Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, T. Toda, M. Wester, and J. Yamagishi, "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*, vol. 24, no. 4, pp. 768–783, Apr. 2016.

[3] H. Yu, Z. Tan, Z. Ma, R. Martin, and J. Guo, "Spoofing Detection in Automatic Speaker Verification Systems Using DNN Classifiers and Dynamic Acoustic Features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4633–4644, Oct. 2018.

[4] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," *arXiv:1710.07654 [cs, eess]*, Oct. 2017, arXiv: 1710.07654. [Online]. Available: http://arxiv.org/abs/1710.07654

[5] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499 [cs]*, Sep. 2016, arXiv: 1609.03499. [Online]. Available: http://arxiv.org/abs/1609.03499

[6] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 2–6. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1111.html

[7] H. Dinkel, Y. Qian, and K. Yu, "Investigating Raw Wave Deep Neural Networks for End-to-End Speaker Spoofing Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2002–2014, Nov. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8398462/

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv:1512.00567 [cs]*, Dec. 2015, arXiv: 1512.00567. [Online]. Available: http://arxiv.org/abs/1512.00567

[9] W. G. Cochran, "The χ^2 test of goodness of fit," *Ann. Math. Statist.*, vol. 23, no. 3, pp. 315–345, 09 1952. [Online]. Available: https://doi.org/10.1214/aoms/1177729380

[10] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar 1986. [Online]. Available: https://doi.org/10.1007/BF00116251

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," p. 7.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861 [cs]*, Apr. 2017, arXiv: 1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861

[14] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *arXiv:1610.02357 [cs]*, Oct. 2016, arXiv: 1610.02357. [Online]. Available: http://arxiv.org/abs/1610.02357