FoR: A Dataset for Synthetic Speech Detection

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*—With the advancements in deep learning and other techniques, 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 even humans have difficulties 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.**

In this paper, we introduce the FoR Dataset, which contains more than 198,000 utterances from the latest deep-learning speech synthesizers as well as real speech. This dataset can be used as base for several studies in speech synthesis and synthetic speech detection. Due to its large amount of utterances, it is pertinent for machine learning studies, since it is able to train even complex deep learning models without overfitting. We present several experiments using this dataset, including a deep learning classifier that reached up to 99.96% accuracy in synthetic speech detection.

***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 advancements in deep learning and other techniques, synthetic speech is getting closer to a natural sounding voice. Some of the state-of-art text-to-speech (TTS) technologies achieve such a high level of naturalness that even 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 else'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. As a result, it is crucial to develop techniques that discriminate between real speech and synthetic speech. For such techniques that involve machine learning, it is important to have an appropriate dataset for training.

In this paper, we present the *Fake or Real (FoR) Dataset*, which contains more than 87,000 synthetic utterances as well as more than 111,000 real utterances. Such a dataset is fundamental for research in synthetic speech detection, since it contains enough data to train the most complex deep learning algorithms.

Although previous researchers also generated datasets containing real and synthetic utterances [1], [2], for this dataset we focus on the latest speech synthesis technologies using neural network architectures. We include not only open-source systems, but also commercial tools that can be used to generate synthetic speech.

To create this dataset, we conducted extensive research on the latest open source and commercial methodologies in speech synthesis. After these approaches were identified, we used a special set of phrases to generate utterances from each TTS system. That resulted in more than 87,000 synthetic utterances from a total of 33 synthesized voices.

We also collected real speech utterances for the FoR dataset. Collecting real utterances is a complex task: one needs to ensure a variety of recording methods, a variety of speaker genders, a variety of speaker ages, a variety of accents and even a variety of microphones used for recording. This variety is required to avoid unintentional bias in the training data that would result in classification methods not generalizing well to unseen TTS systems. We identified and collected utterances from a series of open source speech datasets as well as other sources of real speech, such as TED Talks and Youtube videos.

The dataset is published in four versions. The first version contains the files as collected from the speech sources, without any modification. The second version contains the same files but balanced in terms of gender and class and normalized in terms of sample rate, volume and number of channels. The third version is based on the second one, but with the files truncated at 2 seconds. The last version is a rerecorded version of the dataset, to simulate a scenario where an attacker sends an utterance through a voice channel (i.e. a phone call or a voice message).

We also present a series of experiments regarding synthetic speech detection and an analysis of the main differences between synthetic speech and real speech. Our first experiment consists in training machine learning architectures to detect synthetic speech. We achieve up to 99.96% validation accuracy and 92.00% testing accuracy.

We also analyze the impact of noise on the synthetic speech detection task and we identify that the classification accuracy stays high up to 35% noise ratio. After that, it starts to drastically decrease. After 45% noise/signal ratio, the machine learning models are not able to identify synthetic audio anymore.

Another experiment showed that synthetic speech and real speech present differences in terms of audio brightness, depth, roughness and hardness. Those differences can also be used as an additional input in the synthetic speech detection task.

The remainder of this paper is organized as follows: In the next section, we present an overview of the current research on synthetic speech datasets and synthetic speech detection. In Section III we formally introduce the FoR Dataset and its versions. In Section IV we present a series of experiments utilizing the FoR Dataset. In the last section, we present a conclusion for our work and discuss potential future work.

II. Background

A. Synthetic Speech Datasets

Automatic Speaker Verification (ASV) solutions are authentication solutions that use the human voice as a mean of authentication. A replay spoofing attack consists in recording someone's voice and replaying it in an attempt to fool the ASV system and gain access to a system. Motivated by this threat, researchers from across the globe created a dataset containing real voices and spoofed voices [3]. With this dataset in mind, the same researchers created a challenge, called ASVSpoof Challenge, so the research community could study and propose methodologies that solved the ASV replay spoofing attack. One of the most cited versions of this dataset is the ASVSpoof2015 dataset, which contains not only spoofed utterances, but also computed-generated speech. The synthetic utterances were generated using traditional text-to-speech systems and voice-conversion systems, which lack in naturalness of speech. The ASVSpoof2015 dataset does not include the latest deep-learning-based synthetic speech systems.

In a paper published in early 2016, researchers discuss methodologies for identifying spoofing attacks using automated solutions [4]. In their study, they analyze the effectiveness of 5 TTS systems as well as 8 voice-conversion (VC) systems against three ASV systems. Their conclusion is that the ASV systems are vulnerable to these spoofing attacks. However, adding their proposed spoofing detection system can lower the false-acceptance rates to less than 1%. The study brings several interesting findings to the research community. Apart from the key contribution of developing an anti-spoofing system, the authors also published the Spoof and Anti-Spoof dataset, that includes not only spoofed utterances, but also computer-generated speech. However, it is important to note that the TTS and VC systems utilized in this research are outdated compared to the current state-of-art speech-synthesis systems.

B. Synthetic Speech Detection

In mid 2017, Paul et al. proposed a set of short-term spectral features that can drastically improve the accuracy of synthetic speech detection [5]. The authors provide a thorough analysis of the differences between synthetic speech and real speech and identify interesting patterns, such as the fact that lower frequencies ($<$1kHz) and high frequencies ($>$7kHz) are the most useful frequencies for discrimination between synthetic and real speech.

A good portion of the synthetic speech detection studies focuses on extracting frequency information and using this information to train a classifier. This kind of approach usually assumes frame-by-frame independence and does not learn long-term temporal information. However, a study published in 2013 shows that having temporal data increases the performance of synthetic speech classifiers [6]. This is an indication that using a full audio representation with a deep learning approach may result in higher performance than using only frequency-based methodologies.

C. Neural Networks and Deep Learning

The idea behind neural networks goes back to 1943, when researchers created a computational model called threshold logic [7]. The model consists of a collection of connected units that perform logic tasks. Each unit, also called neuron, is composed by an input, an activation function and an output. The neurons can be interconnected using their inputs/outputs and a weight factor. This forms a computational network, also called, artificial neural network. The real potential of neural networks was only explored later in 1975, when researchers from Harvard University published their back-propagation algorithm [8]. This algorithm enabled neural networks to efficiently learn by adjusting the weights in each node, making it possible to train complex neural network models using supervised data. The back-propagation algorithm is still in use today, decades after its creation.

When first published in the 80's, the idea of *Convolutional Neural Networks (CNNs)* was a groundbreaking finding [9]. More than a decade later, researchers were able to use this architecture to ingest a multi-dimensional input (e.g. an image) and learn dimensional/positional relations between the pixels [10], which enable the neural networks to recognize shapes and patterns.

One of the main publications related to CNNs is an article from 2012 by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton regarding the use of convolutional neural networks for image classification [11]. In their publication, the authors discuss the use of an eight-layer deep convolutional neural network over the ImageNet dataset [12] to detect and classify objects in pictures.

D. Synthetic Speech Detection Using Deep Neural Networks

With the increase in the popularity of Deep Neural Networks (DNN) solutions, Yu et al. published a paper regarding the use of DNNs for speech spoofing detection [13]. The main idea is to use DNNs to extract dynamic acoustic features and classify an utterance as real or spoofed. The research shows that this proposed methodology overperforms the traditional static feature analysis with GMM classifiers. Previous studies [14], [15] show that dynamic acoustic features (such as dynamic filter banks, dynamic MFCCs, and dynamic linear prediction cepstral coefficients) are better candidates for spoofing detection than traditional static features (such as magnitude-based features and cosine normalized phase features). Based on that and the fact that DNNs are well known for their capabilities of extracting dynamic features, the researchers decided to implement a 5-layer deep neural network in conjunction with 5 different dynamic filter-bank-based scoring methods to perform classification on the AVSpeech2015 dataset. Although this dataset does not cover the latest deep-learning based TTS systems (such as DeepVoice3), it is a good starting point to train a DNN to detect spoofed speech. The results of the experiment show that DNNs with dynamic features present better performance than the previous methodologies using static features and GMM models.

In mid-2017, Zhang et al. published a paper regarding their investigation of deep-learning frameworks for speaker verification anti-spoofing [16]. In their research, the authors propose the use of CNNs in conjunction to RNNs to identify synthetic speech. Using as baseline the ASVSpoof2015 dataset, the proposed methodology presents the state-of-the-art performance for an end-to-end single system.

Xiaohai Tian and Xiong Xiao published a paper regarding their work on spoofed speech detection using temporal convolutional networks [17]. Their idea is to use a single convolutional neural network to classify an utterance instead of using handcrafted feature extractors with traditional machine learning approaches. The proposed architecture is tested against the ASVspoof2015 dataset and shows a relevant improvement, especially in unseen spoofing attacks and in temporal-based speech synthesizers.

Muckenhirn et al. explored the use of CNNs for end-to-end speech spoofing detection [18]. Although the proposed approach is not new and uses the outdated ASVspoof2015 dataset, the authors present an interesting analysis of what features are being learnt by the model. In this analysis, the authors show that the proposed architecture is mainly learning discriminative information from the lower and higher frequencies, which matches with previous studies that used manual feature extraction with traditional machine-learning algorithms. This shows that the DNN is able to extract frequency features right from the raw audio and that the DNN learns from the same spectrum regions than the traditional approach, with a similar or higher accuracy.

III. The FoR Dataset

Several datasets containing synthetic speech have been published in the past [2], [3], [13]. However, there are many reasons that necessitate the introduction of the dataset presented in this paper. To start with, the vast majority of the utterances in existing datasets have not been generated from 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 [19]. Finally, the majority of the published datasets focuses on the detection of spoofed utterances for automatic speaker verification systems.

In this paper, we introduce the *Fake or Real (FoR)* dataset, which is composed of more than 87,000 synthetic utterances as well as more than 111,000 real utterances (from a large variety of individuals). The main difference between the FoR dataset and previous works is that our dataset 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, it is enough to train complex models, such as InceptionV3 [20], without overfitting.

The FoR dataset is under GNU GPLv3 license and is publicly available to the community[1].

A. Synthetic Speech Collection

We begin by describing the part of the FoR dataset that contains synthetic utterances. As previously discussed, the use of deep learning for speech generation has increased in the past few years. With that in mind, we identified and collected utterances from the latest methodologies in speech synthesis, both open source and commercial. The chosen TTS systems and the number of voices and utterances can be seen in Table I.

TABLE I
TTS Systems and utterances

Source	Voices	Total Utterances
Deep Voice 3	1	2645
Amazon AWS Polly	8	21160
Baidu TTS	3	7935
Google Traditional TTS	1	2645
Google Cloud TTS	2	5290
Google Wavenet TTS	2	5290
Microsoft Azure TTS	16	42320
Total:	**33**	**87285**

With the scope of TTS systems defined, our next step was to identify a list of phrases that can be used to generate utterances. One of the main concerns when creating a dataset is to have a high variety of data points to ensure that the underlying distribution is well represented in the dataset. With that in mind, it was important to choose a high variety of phrases to be used as input to the TTS systems.

We utilized a phrase dataset[2] that is commonly used in natural language translation. This dataset is open to the public and contains over 150,000 English phrases and their French translation. Since our work focuses on the English language, the French part of the dataset was discarded, leaving us with a dataset of English phrases with a high variety of grammatical structures (passive/active phrases, simple/complex phrases, short/long phrases, affirmative/question phrases, etc.). The phrases were filtered to remove duplicates, as well as phrases surpassing 30 words.

The resulting phrase dataset was then randomly divided into 33 phrase buckets, containing 2645 phrases each. Each phrase bucket was used by only one TTS voice. This ensures that there are no repeated utterances in the dataset, minimizing

[1] http://bil.eecs.yorku.ca/datasets

[2] https://www.kaggle.com/percevalw/englishfrench-translations/kernels

the risk of models learning specific words/phrases instead of a generalized model to differentiate between real and synthetic speech. With the phrase buckets ready, we then generated utterances from each TTS system using their APIs, as described below.

The DeepVoice 3 [21] system is an end-to-end TTS solution developed by Baidu Labs. This model is capable of generating an audio representation (spectrogram) given a phrase as input. This representation can then be transformed into an utterance using an algorithm such as Griffin-Lim, or a Wavenet architecture. For this collection task, we obtained an implementation of the DeepVoice 3 system[3] which was then trained using the utterances from the LJSpeech speech dataset (see Section III-B). The reason for using LJSpeech for both generating synthetic utterances as well as for the real utterances part of the dataset is to increase the likelihood that models will learn real characteristics of synthetic and real speech, since using the same voice for both classes minimizes the chance of the model classifying based on voice properties (pitch, intensity, etc.). After the model was trained on the LJSpeech dataset, one phrase bucket was used to generate 2645 utterances.

The original Google TTS is one of the most known text-to-speech systems. It is present in several Google products, such as Google Translate[4] and Google Home. Although this system does not utilize the latest deep learning techniques, it is a popular TTS system thus it is included in our research. As Google TTS is a proprietary system, it is not possible to get access to its source code. However, it is possible to use API calls to extract audio from the system. By creating a script that sends an API request per phrase and saves the returning result, we were able to generate 2645 utterances from this system.

With the increase in the popularity of cloud TTS services, Google created its own cloud TTS service[5]. This service uses the latest deep learning techniques in conjunction with manual tuning to provide a cloud TTS service. The resulting utterances have a high naturalness, approaching human-like speech. Moreover, the system also accepts a SSML[6] (Speech Synthesis Markup Language) file as input, meaning that one can define several speech variables, such as emphasis and break times. To extract utterances from the cloud API, first it was required to create a Google Cloud account. Then, it was required to create a project key so the API can be accessed. After having the keys created, it was required to install the Google SDK and create a script that reads a file containing phrases and retrieves the utterances from the Google Cloud. With that done, 5290 utterances were extracted in two different voices (2645 utterances per voice).

Similarly to the Google Cloud TTS, Google released a premium version of its TTS system. This premium version, called Google Cloud Text-to-Speech with Wavenet, uses a mixed model using deep learning techniques in conjunction with a Wavenet model to generate the utterances. The system also accepts as input SSML files, meaning that precise speech can be generated. This improved model is able to generate utterances with very high naturalness, where the synthesized speech is almost indistinguishable from real speech. The process of utterance extraction on the Google Cloud TTS Wavenet is very similar to the non-premium Google Cloud TTS: First it is necessary to obtain API keys, then install the Google SDK and finally use a script to iterate over a text file and obtain the utterances through the API. With that done, 5290 utterances were extracted in two different voices (2645 utterances per voice).

Amazon AWS Polly[7] is one of the most known TTS systems. Similar to other commercial solutions, this service allows the synthesis of utterances using the latest deep learning techniques. At the moment the FoR dataset was generated, 8 English voices were available in a variety of accents (American English, British English, Australian English and Indian English). One of the main advantages of Amazon Polly is that it is able to synthesize natural speech with high pronunciation accuracy (including abbreviations, acronym expansions, date/time interpretations, and even homograph disambiguation). To synthesize utterances using Amazon Polly, it is first necessary to create an Amazon AWS account. With the account created, it is possible to generate an API key that enables the access to the Polly TTS system. A script was created to read phrases from a text file and interact with the Polly API to retrieve the utterances. A total of 21160 utterances were extracted from this system, 2645 utterances for each one of the 8 speakers.

Similar to Amazon and Google, Microsoft recently released its text-to-speech solution. Called Microsoft Azure Text-To-Speech[8], this service is capable of generating utterances from input phrases. The interesting part about Microsoft TTS is that it provides 16 voices just for the English language (in a variety of accents). Moreover, the Microsoft TTS system allows any person to upload samples of their voice so the model can learn and reproduce their voice. This allows a much more customized experience for customers, since one can have their own voice being spoken in a system. The process to extract utterances from the Microsoft TTS system is fairly simple: First, you create an API key to access the service. Then, using an HTTP request, one can send a phrase to the system that answers with an MP3 file. To automate the collection, a script was created to read an input file (containing phrases) and submit POST requests to the Microsoft TTS server. A total of 42,320 utterances were synthesized from the Microsoft TTS system, 2645 utterances for each one of the 16 voices.

Although Baidu published research papers detailing the use of machine learning for TTS (DeepVoice 1 [22], DeepVoice 2 [23] and DeepVoice 3 [21]), they also developed a commercial system (which is sold as a service) that has higher performance than the open source code released on the internet. This

[3] https://github.com/r9y9/deepvoice3_pytorch

[4] https://translate.google.com/

[5] https://cloud.google.com/text-to-speech/

[6] https://www.w3.org/TR/speech-synthesis11/

[7] https://aws.amazon.com/polly/

[8] https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/

service, called Baidu Cloud TTS[9] is able to generate utterances given text as input. Although the system is mainly trained for the Chinese language, the service is also offered in English (however, with lower naturalness compared to the Chinese voice). Baidu offers an interface through their speech synthesis app through which it is possible to submit phrases and obtain audio files. To automate the process, a script was created to generate HTTP requests to the app and receive the resulting MP3 file. As there is only one English voice available, 2645 utterances were extracted from this system.

B. Real Speech Collection

The FoR dataset also contains a large number of real utterances, i.e. speech recordings from humans. The process of collecting real utterances is a complex task since we need to ensure that the collection method does not introduce unintentional bias. For example, it is necessary to ensure that the utterances are recorded using a variety of microphones, otherwise the machine learning algorithm may learn to classify based on features specific to one recording device, instead of learning the real differences between a synthetic utterance and a real utterance. For the same reason, it is important to have a large variety of voices from all genders, as well as a good variety of accents.

The first step was to identify potential sources of real utterances. Two main source categories were identified: open source datasets, which provide a large amount of pre-processed speech; and internet recordings, such as speech extracted from Youtube videos.

Open source speech datasets can be a great source of real speech, since they provide a wide variety of clean recordings. The following open source datasets were selected and incorporated into the FoR Dataset:

- Arctic Dataset[10]: This dataset contains 1132 utterances spoken by 7 professional voice actors, resulting in a total of 7924 utterances. This dataset was chosen because it contains a good variety of accents as well as having utterances from all genders. Also, as we have the same utterances being spoken by 7 different speakers, it increases the chances of the classifier learning a more generalized model for real/synthetic classification.
- LJSpeech Dataset[11]: The LJSpeech dataset contains 13,100 utterances from one female speaker. This dataset is a well known real-speech dataset used in several TTS publications, such as DeepVoice 3 [21], and for this reason it was chosen for this research. Also, this dataset was used to train the DeepVoice 3 model, meaning that we have synthetic and real utterances from the same voice, increasing the chances of a classifier learning a more generalized model for real/synthetic classification.
- VoxForge Dataset[12]: VoxForge is an open source real speech dataset in which any person can record and submit utterances to the project. This creates a dataset with a large variety of voices, recording devices and even audio quality. At the time of the collection, this dataset contained more than 86,000 utterances, from more than 1,200 persons using a large variety of recording devices. This dataset was chosen due to its large amount of different voices as well as the large variety of recording devices, which increases the chances of the classifier learning a more generalized model.

Social media platforms, such as Youtube, can be an excellent source for speech data: they provide a high variability of voices as well as recording devices. However, audio from such sources usually contains a large amount of background noise and/or background music. To minimize the chances of background noise and poor recording quality, we selected a variety of educational videos as source of speech. Educational videos (such as TED talks, online courses and tutorials) are good candidates because they typically are recorded in a silent environment (using a high quality recording device) and typically contain only one speaker. For this research, 140 videos (speakers) were selected. From those videos, the full audio was extracted and the SoX[13] tool was utilized to segment the audio where a silence of 2 seconds or more was detected. The purpose of segmenting the audio based on silence is to avoid broken utterances, where audio is cut while someone is speaking. This process resulted in a total of 3720 utterances from 140 speakers.

C. Dataset Versions

As we expect the FoR Dataset to be used in machine learning experiments, it is important to pre-process the utterances in a way that eliminates bias. Based on the pre-processing applied, we identified and generated four different versions of the dataset: for-original, for-norm, for-2seconds and for-rerecorded.

The original dataset, named *for-original*, contains the files as collected from the speech sources, without any modification or class/gender balancing. A total of 195,541 utterances are present in this dataset version. The gender and class distributions can be found in Figure 1. Although the data in this dataset version is unbalanced in terms of gender and class distribution, it is being published so the research community can use the raw data with their own pre-processing techniques.

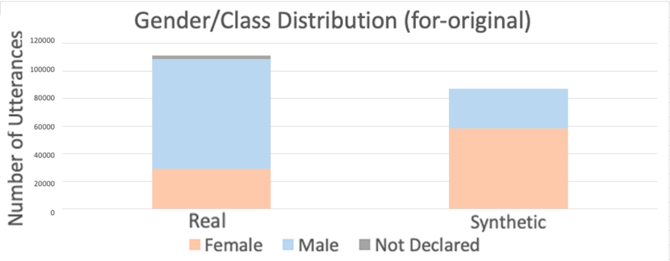

Fig. 1. Gender Distribution - FoR Original

[9] https://cloud.baidu.com/product/speech/tts

[10] http://festvox.org/cmu_arctic/

[11] https://keithito.com/LJ-Speech-Dataset/

[12] http://www.voxforge.org

[13] http://sox.sourceforge.net/

As the dataset is composed by utterances from several audio sources, it is essential to normalize the data to eliminate bias. The normalized dataset, named *for-norm* contains the same files as the original dataset but with the audio converted to WAV, normalized to 0dBFS, downsampled to 16kHz sample rate, converted to mono and with silences removed from the beginning and end of the utterances.

The filetype conversion consists in ensuring that every file in the dataset is in the same format. The original dataset contains audio in two formats: WAV and MP3. Since WAV is the preferred format for machine learning algorithms, all audio files were converted to WAV using the ffmpeg[14] tool. To automate the conversion of all the files, a script was created to convert the whole dataset keeping the same folder structure.

A possible concern for someone considering using the FoR dataset could be that the format in which the speech was originally recorded could impact on the classification results. To alleviate this concern, we performed an experiment in which we converted the whole dataset to MP3 and then converted everything back to WAV. This process did not affect our accuracy results, which suggests that MP3 compression does not introduce bias.

Normalizing the volume in an audio dataset is a common practice in machine learning research, since inconsistent volume levels can impact on learning and classification. As the files were collected from several data sources, each one with their own volume settings, it is important to normalize the volume of all utterances to eliminate the possibility of volume becoming a distinguishing factor. Using the SoX tool, all the audio files were normalized to 0dBFS.

The sample rate is an important factor when training a machine learning algorithm: all the input audio should be at the same sample rate to ensure the audio is processed correctly. The majority of the audio files collected had a sample rate of 16kHz, but there were also files recorded at 22kHz, 24kHz and 48kHz. Since the human voice frequency spectrum typically ranges from 300Hz up to 5000Hz, using 16kHz as the common sample rate provides enough room for the task, since it can accurately represent audio signals up to 8000Hz. Using the SoX tool in conjunction with a custom script, the whole dataset was downsampled to 16kHz.

All the synthetic speech solutions generate audio in a single channel (mono), while a good fraction of the real utterances were recorded in two channels (stereo). To avoid this becoming a distinguishing factor, all the audio files were converted to mono using the SoX tool. This tool uses a channel mixing technique, which combines two audio tracks into a mono track by scaling each track by 0.5 and adding the signals to result in a single track.

The SoX tool was also used to remove the silence in the beginning and end of an audio file. To automate the process, a script was created to automate the silence removal in the whole dataset. First, we remove the silence from the beginning of the file. Then, as SoX does not have a feature to remove the silence in the end of the file, we reverse the audio file, cut the silence from the beginning, and reverse the audio again. As a result of this processing step, all the files have no silence in the beginning nor in the end of the audio.

Finally, and perhaps most importantly, we balanced the data to achieve even distribution between genders and classes (synthetic/real). The resulting distributions can be seen in Figure 2, where it is possible to note a more even picture in terms of class and gender. Due to the balancing process, the for-norm version of our dataset contains a total of 69,400 utterances.

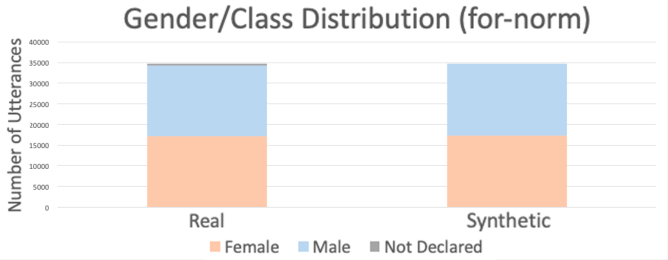

Fig. 2. Gender Distribution - FoR Normalized

Several audio classification methods require that all audio inputs are of the same length. As the for-norm version of our dataset contains full utterances, they are of varying length. In fact, it turned out that the synthetic audio was considerably shorter than the real audio. While the synthetic audio was on average 2.35 seconds long (with a standard deviation of 0.83 sec), the real utterances were on average 5.05 seconds long (with a standard deviation of 1.95 sec). Figure 3 shows the audio length distribution for both real and synthetic audio. This significant length difference may affect the classification results since the model may learn to distinguish between real and synthetic speech based on the length of the audio. To address this issue, we created a new version of the dataset as follows: we started by discarding all files shorter than 2 seconds. Then, we truncated the remaining files at the two-second mark. The last step was to re-balance the dataset to achieve even distribution across genders and classes. The resulting version of the dataset, named *for-2seconds*, contains a total of 17,870 utterances.

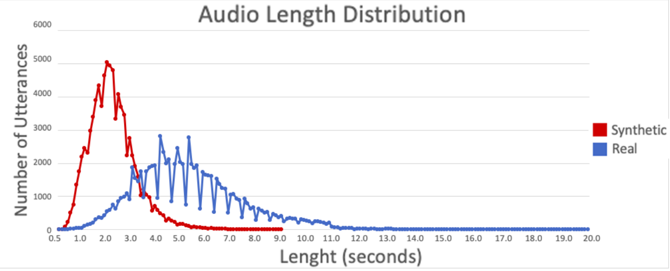

Fig. 3. Audio Length Distribution

To simulate a real-world synthetic speech attack, we decided to re-record the dataset. The idea is that in a real world

[14] https://www.ffmpeg.org/

scenario, a malicious person may generate/play the synthetic speech with one device (e.g. a computer) and record it using another device (e.g. a smartphone). This is an example where the attacker is trying to impersonate someone via a communication channel (e.g. a phone call or a voice message). To simulate this scenario, we played the utterances from the for-2seconds dataset using a regular computer speaker and recorded them using a non-professional microphone, simulating a casual attacker. The resulting version of the dataset, referred to as *for-rerecorded*, contains re-recorded utterances that simulate a real world attack. As the utterances were recorded at 16kHz and the volume was constant during recording, there was no need for downsampling or volume normalization.

To get a better understanding of which frequencies were most affected by the re-recording process, we used a chi-square test to identify the frequencies that differ the most between original and re-recorded audio. The representation used for this test was an 1024-bin STFT audio representation (due to its frequency bin linearity). To help with the visualization of the results, the frequency bins were ordered (from 0Hz to 8kHz) and a colour was attributed to each frequency bin: red meaning high difference score, green meaning low difference score.

The results of this analysis for synthetic speech can be seen in Figure 4, in which it is possible to observe that higher frequencies in synthetic speech were the most affected by the re-recording process. When the same analysis was applied to real speech, the results were as shown in Figure 5. Real speech was also affected when it comes to its high frequencies, but it is clear that the re-recording process has a larger effect on synthetic speech. This validates our decision to re-record the dataset, as it is quite possible that classification results may be quite different in this case.

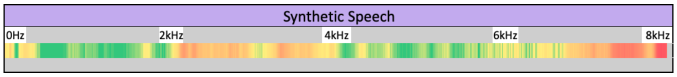

Fig. 4. Chi-Squared Frequency Change Map - Synthetic Speech

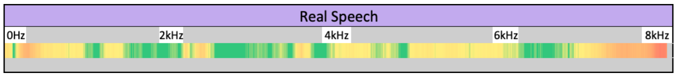

Fig. 5. Chi-Squared Frequency Change Map - Real Speech

D. Dataset Division

The original dataset (for-original) is organized in folders according to their source. All the preprocessed versions of the dataset (for-norm, for-2seconds and for-rerecorded) were divided into training, validation and testing, as is common practice in machine-learning research. The division is as follows:

- Training: Contains 77.73% of the dataset, utilized to train the machine learning models. Gender and class balanced.
- Validation: Contains 15.58% of the dataset, utilized to validate the accuracy of the machine learning models. Gender and class balanced. The validation utterances are unseen during the training phase.
- Generalization Testing: Contains 6.68% of the dataset. Contains only synthetic voices from one unseen algorithm (Google TTS Wavenet) and unseen real voices. Gender and class balanced. It is utilized to test if the trained model can generalize and detect unseen TTS algorithms and unseen real voices.

With the dataset versions created, processed and divided, they are ready to be used by the research community.

IV. Experiments

To illustrate the usefulness of the various versions of the FoR dataset, we present a series of experiments we conducted with it.

A. Experiment 1: Synthetic Speech Detection

The first experiment corresponds to the main motivation for the creation of this dataset: building a machine learning model that discriminates between real and synthetic speech. We built various traditional machine learning models, as well as several deep learning models. More details about these experiments can be found in [24].

The traditional machine learning models consists of extracting an audio representation (STFT, Mel-Spectrograms, MFCC and CQT) for each audio file, averaging the representation over time to obtain a frequency activation vector, and inputing this vector into Weka[15] with the appropriate classes (synthetic/real) to obtain accuracy results.

The deep learning models consist of extracting audio features (STFT, Mel-Spectrograms, MFCC and CQT) from each audio file and converting them to an image. The resulting images were then used to train selected pre-trained deep learning architectures, such as VGG16/VGG19 [25] and Inception v3 [20]. Since the deep learning models utilize spectrograms, we used the for-2second version of the dataset.

The traditional machine learning results indicate that the MFCC audio representation with the Random Forests method achieves up to 98.54% validation accuracy. This shows that it is possible to achieve high accuracy for some tasks without using the temporal aspect of the input audio (input representation was averaged over time). The results from the deep learning analysis show that the VGG16 and VGG19 models using the STFT audio representation presented the highest validation accuracy (99.96% and 99.94% respectively). This is to be expected as these models had access to temporal information as well.

To evaluate the generalization ability of the above models, we tested the performance of the trained models against a totally unseen TTS algorithm (Google TTS Wavenet, which was not included in the training/validation dataset). This experiment simulates how the models would react if an attacker

[15] https://www.cs.waikato.ac.nz/ml/weka/

creates a new TTS system. The traditional machine learning models achieved up to 86.94% accuracy, while the deep learning models achieved up to 92.00% accuracy.

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 repeated the above experiments with the for-rerecorded version of the dataset. When it comes to validation accuracy, the traditional machine learning models achieved up to 95.05%, while using deep learning it is possible to achieve 99.96%. This shows that the re-recording process had little-to-no impact on the performance of the deep learning methodologies.

However, when the models were applied to the unseen rerecorded TTS algorithm, accuracy dropped to 85.78% for the traditional machine learning models and 91.42% for deep learning. This indicates that deep learning techniques are quite resilient to the rerecording process even in the case of an unseen TTS algorithm.

B. *Experiment 2: Waveform Classification*

As seen in the previous experiments, the synthetic speech detection accuracy using spectrograms is quite high. This raises the question of whether simpler audio representations, such as waveform images, can be enough for the classification of real and synthetic utterances.

For this experiment, we used the for-2seconds version of the dataset, and converted it to waveforms using the ffmpeg tool[16]. The resulting images were then used to re-train the VGG19 model.

Using just waveform images with the VGG19 model, we achieved 89.79% validation accuracy. While this is significantly lower than the validation accuracy achieved with spectrograms, it is still rather surprisingly high. It does raise the question of whether the dataset contains volume bias. Even though all audio files are normalized to 0dBFS, it is definitely possible that the average loudness of synthetic speech is significantly different than that of real speech.

To ensure that the dataset does not contain volume bias, we employed the use of dynamic range compression (using the SoX tool) to reduce the dynamic range of every audio file in the dataset. All files were again normalized to 0dBFS to ensure that their respective loudness is similar. The compressed dataset was transformed into waveform images as before and were used to re-train the VGG19 model. This model achieved a validation accuracy of 89.15%, which is only slightly lower than the original waveform performance (89.79%). This result shows that volume discrepancies have little-to-no effect on the classification process, which suggests that the dataset does not contain volume bias.

C. *Experiment 3: Signal/Noise Ratio Analysis*

An important factor widely analyzed by previous research in synthetic speech detection is the relation between noise and accuracy. The idea is to investigate how noise impacts on the accuracy of the model by adding a variety of levels of pink noise to the utterances and observing the model performance.

For this experiment, we use the for-2seconds version of the dataset. Then, using the SoX tool, we apply pink noise in a variety of volume levels, from 2% to 50% of the resulting audio. This results in six sub-datasets, each one related to a particular level of noise. We then use each of the six sub-datasets to train an independent VGG19 model (with STFT audio representation).

Figure 6 presents our results. As expected, the higher the noise level, the lower the accuracy. It is also possible to note that the accuracy of the model remains high when the noise volume is up to 35%, showing that the architecture is fairly robust against noise. With a noise volume higher than 40% the accuracy starts to drastically decrease. When the noise level is equal or higher than 45%, the VGG19 is not able to distinguish between synthetic or real speech. It is important to note that 45% noise ratio results in an utterance of quite poor quality, making it hard to even distinguish what is being said.

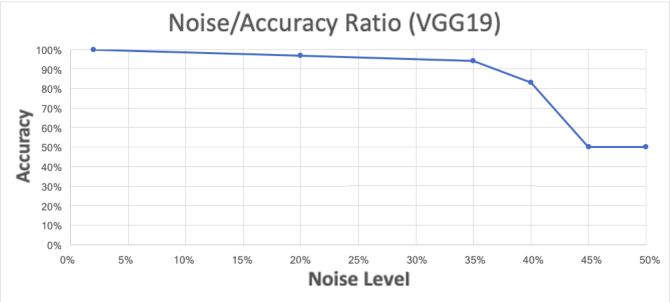

Fig. 6. Noise ratio and accuracy chart

D. *Experiment 4: Timbre Model Analysis*

Our last experiment does not involve the generation of images, so we use the for-norm version of the dataset. The purpose of the experiment is to investigate whether timbre models, such as brightness and roughness, can be used to discriminate between synthetic and real speech. For this reason, we extracted measurements for four main timbre models based on the AudioCommons standards [26]:

- Brightness: A bright sound is one that is clear/vibrant and/or contains significant high-pitched elements.
- Hardness: A hard sound is one that conveys the sense of having been made by something solid, firm or rigid; or with a great deal of force.
- Depth: A deep sound is one that conveys the sense of having been made far down below the surface of its source.
- Roughness: A rough sound is one that has an uneven or irregular sonic texture.

Values for these timbre models were obtained for each audio file using the Audio Commons[17] tool, which is able to generate

[16] https://www.ffmpeg.org/

[17] https://www.audiocommons.org/2018/07/15/audio-commons-audio-extractor.html

scores for each of the above mentioned features and much more. The data was then input into Weka for analysis. We evaluated how well would a classifier perform if only those four features were provided. Table II shows the results of this experiment. Using Random Forests we achieved 79.38% validation accuracy, and using SVM 73.46% testing accuracy. These numbers show that although timbre models are not sufficient classification attributes, they are statistically different in real utterances as opposed to synthetic utterances. Further research needs to be conducted to see how this information can be utilized.

TABLE II
TIMBRE MODEL ANALYSIS: RESULTS

Algorithm	**Validation Acc.**	**Testing Acc.**
Naive Bayes	69.71%	67.27%
SVM	69.91%	73.46%
Decision Tree (J48)	76.78%	70.26%
Random Forests	79.38%	71.47%

V. CONCLUSION

As speech synthesis improves, the need for an up-to-date synthetic speech dataset that can be used in the synthetic speech detection research also increases. In this paper, we introduce the FoR Dataset, which contains more than 198,000 utterances including the latest TTS algorithms and a large variety of real speech.

The dataset is published in four versions:

1) for-original: containing the utterances as collected from various sources
2) for-norm: containing the utterances after a set of preprocessing steps
3) for-2seconds: containing the utterances truncated at 2 seconds
4) for-rerecorded: containing all the utterances rerecorded using a speaker and a microphone

We hope that all four versions of the FoR dataset will provide to the community a solid source of data for synthetic speech detection experiments. This paper presented several such experiments that showcased the usefulness and versatility of the introduced dataset.

A. Research Contribution

With the dataset created and published, we hope that the research community can use it to improve the state of the art in two main areas. The first one is synthetic speech detection, which is a rising concern since TTS systems are achieving human naturalness and can be used for impersonation. As seen in our experiments, it is possible to use our dataset to train deep-learning-based classifiers and achieve high accuracy in this task. The second main area is speech synthesis, since our dataset can be used to improve the quality of neural-network-based text-to-speech systems by using adversarial networks.

B. Future Work

Although the dataset is already a good source of data for synthetic speech detection systems, it can be improved to provide a higher variety of data points.

In regards to synthetic speech, one could improve the dataset by including extra TTS algorithms and/or utterances from voice-conversion systems. Also, although our dataset includes the latest TTS systems, new speech synthesizers are constantly released and could be included in future versions of the dataset.

In regards to our rerecording process, our utterance rerecording was performed using only one type of speaker and one type of microphone. An interesting experiment would be to use a large variety of recording/playing devices in a large variety of recording rooms. This would create a more heterogeneous rerecorded dataset and would create a more generalized synthetic speech detection model.

REFERENCES

[1] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "Automatic speaker verification spoofing and countermeasures challenge (asvspoof 2015) database," 2015. [Online]. Available: http://dx.doi.org/10.7488/ds/298

[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] 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

[4] 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.

[5] D. Paul, M. Pal, and G. Saha, "Spectral Features for Synthetic Speech Detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 605–617, Jun. 2017.

[6] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7234–7238.

[7] W. S. M. Pitts, "Ia logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[8] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1975. [Online]. Available: https://books.google.ca/books?id=z81XmgEACAAJ

[9] K. Fukushima, *Neocognitron: A self-organizing neural network model for a mechanism of pattern recogni- tion unaffected by shift in position*, ser. 4. Springerr, 1980, vol. 36.

[10] Y. Lecun, L. D. Jackel, H. A. Eduard, N. Bottou, C. Cartes, J. S. Denker, H. Drucker, E. Sackinger, P. Simard, and V. Vapnik, "Learning algorithms for classification: A comparison on handwritten digit recognition," in *Neural Networks: The Statistical Mechanics Perspective*. World Scientific, 1995, pp. 261–276.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105, bibtex:alexnet. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 12 2015.

[13] 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.

[14] M. Sahidullah, T. Kinnunen, and C. Hanili, "A comparison of features for synthetic speech detection," 09 2015.

[15] H. Yu, A. Sarkar, D. Alexander Lehmann Thomsen, Z.-H. Tan, Z. Ma, and J. Guo, "Effect of multi-condition training and speech enhancement methods on spoofing detection," 07 2016.

[16] C. Zhang, C. Yu, and J. H. L. Hansen, "An Investigation of Deep-Learning Frameworks for Speaker Verification Antispoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 684–694, Jun. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7815339/

[17] X. Tian, X. Xiao, E. S. Chng, and H. Li, "Spoofing speech detection using temporal convolutional neural network," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. Jeju, South Korea: IEEE, Dec. 2016, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7820738/

[18] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-End convolutional neural network-based voice presentation attack detection," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. Denver, CO: IEEE, Oct. 2017, pp. 335–341. [Online]. Available: http://ieeexplore.ieee.org/document/8272715/

[19] 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/

[20] 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

[21] 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

[22] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time Neural Text-to-Speech," *arXiv:1702.07825 [cs]*, Feb. 2017, arXiv: 1702.07825. [Online]. Available: http://arxiv.org/abs/1702.07825

[23] S. Arik, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-Speaker Neural Text-to-Speech," *arXiv:1705.08947 [cs]*, May 2017, arXiv: 1705.08947. [Online]. Available: http://arxiv.org/abs/1705.08947

[24] R. Reimao, "Synthetic speech detection using deep neural networks," Master's thesis, York University, Toronto/Canada, 2019.

[25] 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

[26] R. M. Andy Pearce, Tim Brookes, "Audio commons: An ecosystem for creative reuse of audio content," 2019. [Online]. Available: https://bit.ly/2uJrnUZ