Synthetic Speech Detection through Audio Folding

Davide Salvi
davide.salvi@polimi.it
Politecnico di Milano
Milano, Italy

Paolo Bestagini
paolo.bestagini@polimi.it
Politecnico di Milano
Milano, Italy

Stefano Tubaro
stefano.tubaro@polimi.it
Politecnico di Milano
Milano, Italy

ABSTRACT

In the field of synthetic speech generation, recent advancements in deep learning and speech synthesis methods have enabled the possibility of creating highly realistic fake speech tracks that are difficult to distinguish from real ones. Since the malicious use of these data can lead to dangerous consequences, the audio forensics community has focused on developing synthetic speech detectors to determine the authenticity of speech tracks. In this work we focus on the wide class of detectors that analyze audio streams on a frame-by-frame basis. We propose a technique to reduce the inference time of these detectors by relying on the fact that it is possible to mix multiple audio frames in a single one (i.e., in the same way a mono track is obtained from a stereo one). We test the proposed audio folding technique on speech tracks obtained from the ASVspoof 2019 dataset. The technique proves effective with both entirely and partially fake speech tracks and shows remarkable results, reducing processing time down to 25%.

CCS CONCEPTS

• **Security and privacy**; • **Information systems** → *Multimedia information systems*; *Social networks*; • **Computing methodologies** → **Artificial intelligence**; *Machine learning*;

KEYWORDS

Audio Forensics, Synthetic Speech, Digital signal processing, Audio Folding

ACM Reference Format:
Davide Salvi, Paolo Bestagini, and Stefano Tubaro. 2023. Synthetic Speech Detection through Audio Folding. In *2nd ACM International Workshop on Multimedia AI against Disinformation (MAD '23), June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3592572.3592844

1 INTRODUCTION

In recent years, speech generation has seen significant improvements thanks to the advancements in deep learning and the increasing availability of computational power. This led to the creation of highly realistic synthetic speech tracks able to deceive human listeners into thinking they are genuine [14]. The possibility of using these technologies opens the door to a new set of exciting options and scenarios that were just imaginable a few years ago, such as seamless interaction with voice assistants and the control of digital devices solely through voice. However, when these systems are used with malicious intent, they can lead to unpleasant consequences, as evidenced by the numerous cases of fraud and blackmail that have recently occurred [8, 23]. The dangers related to the hostile use of synthetic speech generators highlight the need for robust security measures to prevent negative events from happening in the future.

To address this issue, the scientific community has proposed solutions that could benefit several aspects of the audio forensics field [7]. Among these, multiple techniques for synthetic speech detection [9, 19, 26] and attribution [5, 16] have been developed with the intent to guarantee the authenticity and integrity of audio tracks. Several methods have been proposed for this purpose, most of which are data-driven and trained on datasets released to push the research in the field [18, 25]. The developed systems are based on various approaches, ranging from end-to-end methods [11, 21], to the analysis of acoustic features [2, 3], and semantic aspects [4, 6].

In [12], fake speech detection is performed by using a Random Forest classifier that takes as input the errors in First Digit statistics computed on MFCC features with respect to the generalized Benford law. Similarly, [5] employs a set of features inspired by the speech-processing literature and uses them as input to a supervised classifier. The authors of [2] exploit the effectiveness of Residual Convolutional Neural Networks in many classification tasks and apply them to synthetic speech detection by building three ResNet variants based on three different sets of features (Mel-Frequency Cepstral Coefficient (MFCC), log-magnitude Short-Time Fourier Transform (STFT), and Constant-Q Cepstral Coefficient (CQCC)) and show that different acoustic feature sets can lead to different performances in the detection process. ResNet models are also used in [27], combined with a transformer encoder to perform the detection. Finally, [1] discriminates between real and fake speech by exploiting higher-order bispectral correlations that are typically absent in human speech and are introduced by synthesis algorithms.

Other proposed methods focus on working and being reliable in "in-the-wild" conditions. For instance, the authors of [17] present a classifier that estimates the reliability of a prediction made by a synthetic speech detector, increasing the final accuracy of the considered systems. Also, other works have tackled the problem of explainability in synthetic speech detection and explored the critical factors that drive the detection process to make the performed predictions more trustworthy [10, 20].

Even though the developed detectors have proven to be effective at providing accurate predictions in controlled scenarios, adapting them to real-world situations presents several challenges. For instance, if one wants to automatically analyze the authenticity of a large number of audio tracks from a multimedia platform, computational problems may arise due to the impossibility of processing

MAD '23, June 12–15, 2023, Thessaloniki, Greece

ACM ISBN 979-8-4007-0187-0/23/06…$15.00
https://doi.org/10.1145/3592572.3592844

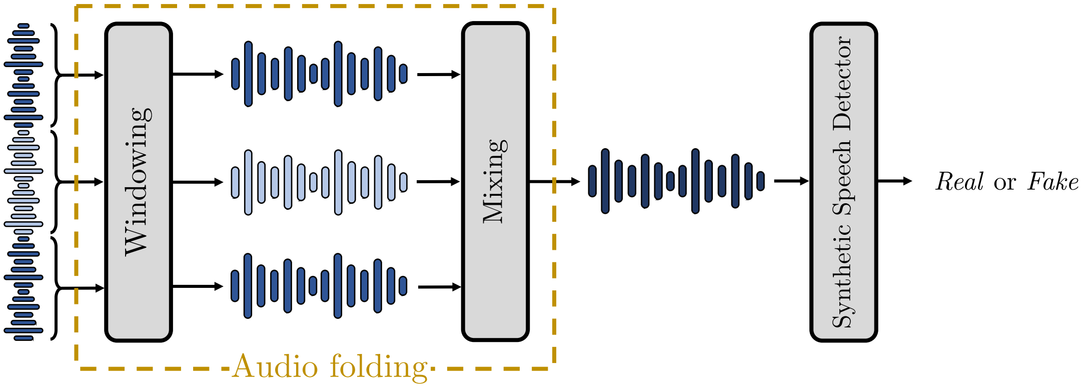

Figure 1: Proposed pipeline for synthetic speech detection on folded audio signals in the case where the number of folding $N_F = 2$.

such a large amount of content in a feasible time. The amount of data uploaded online daily is tremendous and it is exceptionally challenging to develop systems that can analyze all of them automatically on the fly.

Most of the current detection methods process long audio tracks by breaking them down into several time frames and examining each individually, resulting in a computation time that increases proportionately with the length of the signals being evaluated. This is an adequate processing pipeline when the number of tracks to investigate is limited, but it becomes unfeasible when the amount of data under analysis increases exponentially. To tackle this issue there is a need for more efficient systems that can make predictions quickly as well as techniques that can significantly reduce the time required for the analysis.

In this paper we propose a technique that reduces the inference time for a synthetic speech detector that works on a frame-by-frame basis. The proposed method leverages the possibility of mixing multiple audio tracks as commonly done to obtain a mono recording from a stereo one. The intuition is that it is possible to analyze a long track by folding it on itself multiple times. Given a long audio track that would require a significant amount of time to be analyzed window by window, the proposed system breaks down the track into frames and mixes many of them into folded frames. These are then used to perform synthetic speech detection on a shorter duration track, as described in Figure 1. The proposed approach allows us to consider a detector that works per windows and analyze $N_F + 1$ time windows at a time, where N_F is equal to the number of folds applied to the audio track, reducing the total computational time by a factor N_F.

The proposed method works with both speech tracks belonging to a single class (i.e., entirely real or entirely fake) and partially fake tracks (i.e., real and fake signals concatenated together). The difference between the two cases is that in the first one we only mix windows of the same class (i.e., real with real or fake with fake), while in the second one, the folded signal is the union of real and forged tracks together. We apply this technique in several scenarios, up to $N_F = 4$ folds, demonstrating that adequately trained classifiers can have remarkable results despite the difficulty of this task.

The rest of the paper is structured as follows. Section 2 presents the considered problem and illustrates the method we propose to solve it. Section 3 explains the experimental setup used to conduct the experiments. The obtained results are provided in Section 4, together with a brief discussion of them. Section 5 draws some conclusions and gives further ideas for future works.

2 PROPOSED METHOD

In this paper we consider the problem of synthetic speech detection and propose a technique to reduce the computational time needed to analyze extended audio tracks.

The proposed technique is based on the idea that our brain is able to interpret audio stimuli, even when we listen to many concurrent ones. For instance, when multiple individuals are talking together, we are usually able to understand how many speakers are there and follow part of the conversation. Therefore, it is possible to mix multiple single-speaker tracks into a single one containing multiple overlapping speakers (as one would do to convert a multi-channel recording to a mono audio signal) while retaining the information carried out by each speaker.

With this idea in mind, if we are given a single-speaker recording to analyze, we can think of folding the audio multiple times, thus obtaining a shorter track where multiple speakers (actually the same speaker from different time instants) are talking together.

Our intuition is that a synthetic speech detector capable of working with multiple overlapped speakers should also be able to pick up synthetic speech traces in this folded scenario. This is based on the hypothesis that distinctive artifacts that distinguish real and synthetic speech signals are persistent, and we can estimate the authenticity of an audio track even when it is mixed with other portions of the signal itself.

As shown in Figure 1, the folding is performed by mixing different time frames together to create a new signal. By doing so, each frame of the folded track incorporates information from multiple frames of the original input. The folded track is then fed to a synthetic speech detector for classification.

In the following, we provide the formal definition of the synthetic speech detection problem we consider, and we report all the details of our proposed solution.

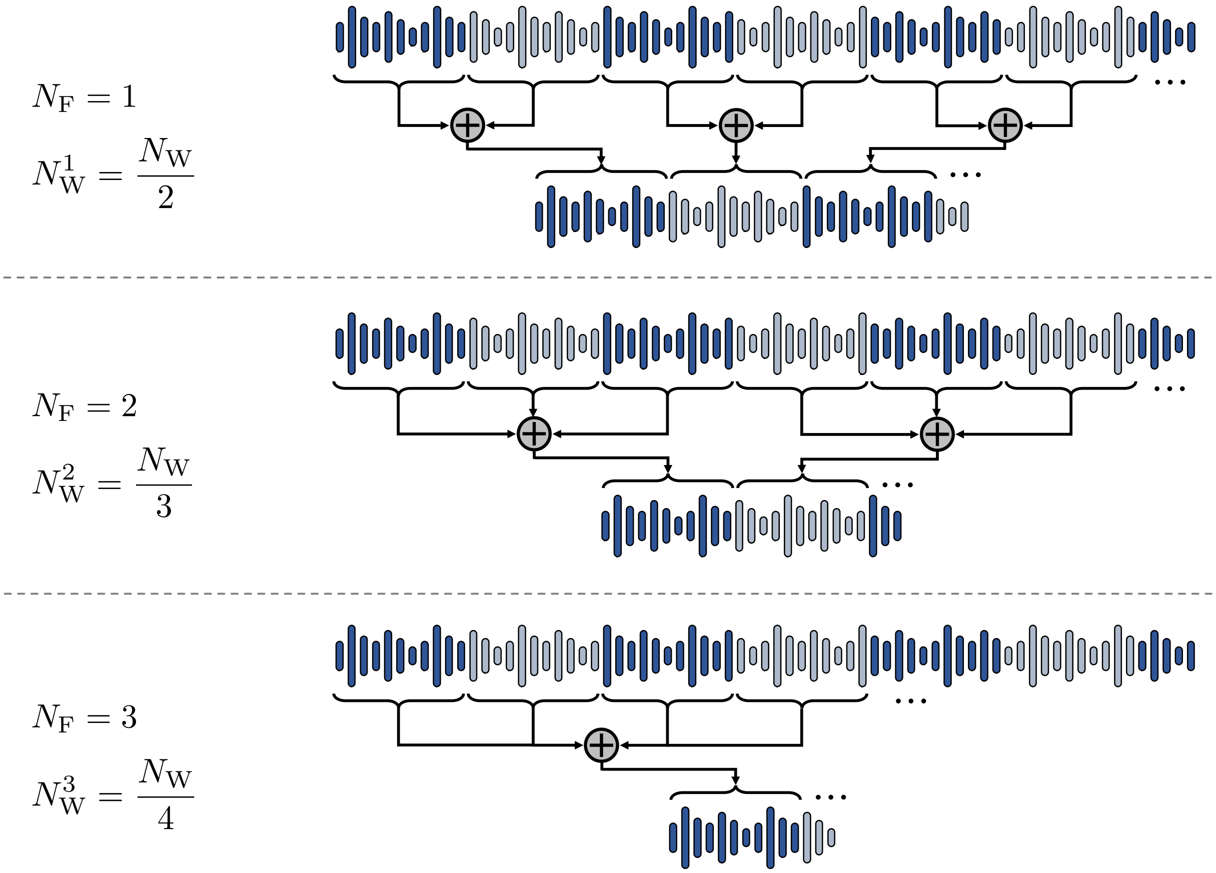
$N_\mathrm{F} = 1$
$N_\mathrm{W}^1 = \frac{N_\mathrm{W}}{2}$
$N_\mathrm{F} = 2$
$N_\mathrm{W}^2 = \frac{N_\mathrm{W}}{3}$
$N_\mathrm{F} = 3$
$N_\mathrm{W}^3 = \frac{N_\mathrm{W}}{4}$

Figure 2: Example of audio folding in the case of $N_\mathrm{F} = 1$, $N_\mathrm{F} = 2$ and $N_\mathrm{F} = 3$.

2.1 Problem formulation

The problem we address is formally defined as follows. Let us consider a discrete-time input speech signal $\mathbf{x}$ obtained by sampling a T second recording with a sampling frequency f_s. The signal is attributed a label $y \in \{0, 1\}$: $y = 0$ means that the recording contains only authentic speech; $y = 1$ means that the recording contains at least one fraction of synthetic speech.

Our goal is to develop a synthetic speech detector $\mathcal{D}$ trained to estimate the class of the signal $\mathbf{x}$ as

$$\hat{y} = \mathcal{D}(\mathbf{x}), \tag{1}$$

where $\hat{y} \in [0, 1]$ indicates the likelihood that $\mathbf{x}$ is fake.

2.2 Audio folding

The first step of the proposed solution consists in generating the folded signal from the input one. To do so, we divide the signal $\mathbf{x}$ into N_W time windows $\mathbf{x}_i$, $i \in [0, N_\mathrm{W} - 1]$ of fixed length T_W, where

$$N_\mathrm{W} = \lfloor T/T_\mathrm{W} \rfloor. \tag{2}$$

We then generate folded frames of order N_F by averaging groups of $N_\mathrm{F} + 1$ adjacent frames $\mathbf{x}_i$. Formally, the i-th folded frame is defined as

$$\mathbf{x}_i^{N_\mathrm{F}} = \frac{1}{N_\mathrm{F} + 1} \sum_{j=i\cdot(N_\mathrm{F}+1)}^{i\cdot(N_\mathrm{F}+1)+N_\mathrm{F}} \mathbf{x}_j, \tag{3}$$

where the sum is performed element-wise, and order $N_\mathrm{F} = 0$ corresponds to no folding (i.e., $\mathbf{x}_i^0 = \mathbf{x}_i$). The folded signal $\mathbf{s}^{N_\mathrm{F}}$ is obtained by concatenating folded frames $\mathbf{x}_i^{N_\mathrm{F}}$ as shown in Figure 2.

The total number of windows of $\mathbf{s}^{N_\mathrm{F}}$ is equal to

$$N_\mathrm{W}^{N_\mathrm{F}} = \lceil N_\mathrm{W}/(N_\mathrm{F} + 1) \rceil, \tag{4}$$

while the total duration of the new signal $\mathbf{s}^{N_\mathrm{F}}$ is equal to

$$T^{N_\mathrm{F}} = T_\mathrm{W} \cdot N_\mathrm{W}^{N_\mathrm{F}} = \frac{T}{N_\mathrm{F}}, \tag{5}$$

meaning that the signal length T^{N_F} gets shorter as much as we increase the number of folds N_F.

2.3 Synthetic speech detector

The second step of the proposed method consists of feeding the folded signal to a synthetic speech detector. To do so, we define as $\mathcal{D}_{N_\mathrm{F}}$ a synthetic speech detector trained on audio signals that have been folded with order N_F.

The synthetic speech detector that we consider in this study is RawNet2 [21]. This is an end-to-end neural network that operates directly on raw audio data, proposed to perform binary classification between real and fake speech tracks during the ASVspoof 2019 challenge [22] and included as a baseline in the ASVspoof 2021 challenge [25]. Its architecture comprises Sinc filters taken from SincNet [15], followed by two Residual Blocks with skip connections on top of a Gated Recurrent Unit (GRU) layer, to extract frame-level representations of the input signal. We use the exact implementation proposed in the original paper, so we refer the reader to that for more information.

We chose to use a detector that works on raw audio data as it has been proven that this type of system outperforms the feature-based models [13]. One of the advantages of using raw data-based models is their high-resolution capacity for making predictions. Compared to feature-based models, which rely on extracting relevant features from the audio signal, raw data-based models process the audio signals directly, capturing more intricate details and complexity that feature-based models may overlook. This greater level of detail can improve the accuracy and robustness of the detection system.

3 EXPERIMENTAL SETUP

In this section, we outline the evaluation setup employed in our experiments. We start by introducing the dataset that was used for training and testing the synthetic speech detectors. Then, we detail the training parameters that we utilized to train the considered detectors, together with the approach we considered to fold the audio tracks.

3.1 Dataset

During all the experiments, we considered the ASVspoof 2019 dataset [22]. This is a speech audio dataset that contains both real and synthetic tracks generated based on the VCTK corpus [24]. The dataset has been released for the homonymous challenge, where participants had to compete to implement the best detector for Automatic Speaker Verification (ASV). It has been proposed to address two different tasks and here we consider the Logical Access (LA) one, which relates to the synthetic speech detection problem. The LA dataset is further divided into three sub-partitions, called *train*, *dev* and *eval*, which all include authentic signals along with synthetic speech samples generated with various methods. The *train* and *dev* partitions have been created using the same set of 6 synthesis algorithms (named *A*01, *A*02, ..., *A*06), while the *eval* partition includes samples generated with 13 different techniques (*A*07, *A*08, ..., *A*19). In all the partitions, every speech generation technique replicates the entire corpus of real tracks. This results in a highly comprehensive dataset as it includes several synthesis algorithms but leads to a significantly unbalanced corpus as the real data are much less than the synthetic ones. To overcome this issue, we ensure to consider the same number of samples from the two classes in each batches, in the model training stage. We considered the *train* and *dev* partitions to train the considered detector and the *eval* set to test it. The distinction between the speech generation techniques included in the training and test partitions allows us to perform analyses in an open set scenario and evaluate the considered detector on data generated only by unseen synthesis algorithms, making our results more meaningful regarding the generalization capabilities of the considered detectors.

The average length of the tracks contained in this corpus is lower than 5 seconds, which does not allow us to apply the folding technique we propose. Nevertheless, we decided to consider this dataset as there are no synthetic speech detection datasets containing long tracks in the literature, and ASVspoof is the most representative corpus of state of the art. To create data for training and test, we employ a simulation method that combines multiple shorter tracks. This process mimics the folded pattern of long signals, which we propose in this work. This approach also allows us to have complete control over the mixed tracks, as we can decide the class of each of the signals considered, performing the experiments in a controlled scenario.

3.2 Training strategy

During the training phase, we trained the detector to discriminate between real and synthetic speech data. To read the audio signals, we considered a sampling frequency of $f_s = 16\,\mathrm{kHz}$ and assumed a time window $T_\mathrm{W} = 3.0\,\mathrm{s}$. We adopted this window length because, from preliminary experiments, it turned out to be a good compromise between the shortness of the window and the performance of the model, which is ideal in a real-world scenario. The audio tracks that are shorter than the assumed time window have been repeated to reach the length needed. As mentioned above, we ensure that each batch contains has the same number of real and fake samples. We have experienced in preliminary experiments that this approach can significantly increase the detection capabilities of the detector.

All the hyperparameters used to train the RawNet2 model have been fine-tuned to maximize its classification accuracy. In particular, we considered a number of epochs equal to 100 with an early stopping patience at 10 epochs, weighted cross-entropy as loss function and Adam optimization, and a batch size of 128. We adopted a learning rate value equal to 10^{-4}, a weight decay of 10^{-4}, and reduced the learning rate on plateau by a factor 0.1.

During the experiments, we train several detectors considering different folding numbers N_F of the input tracks. In particular, we train 4 models, called $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, which are trained with a number of folds from 0 to 3 respectively, using the same hyperparameter setup. When training the models, we approximate a folded signal by mixing several separate tracks. We do so by considering a number of tracks equal to $N_\mathrm{F} + 1$ and mix them together by performing a simple average of the signal.The resulting signal is then considered as the input of model.

Also in this case, we must pay attention to balance the classes considered to compensate for the imbalance of the dataset. We perform two different types of balancing. The first consists in balancing the two classes $y_i^\mathrm{s} \in \{0, 1\}$ among all the window signals $\mathbf{s}_i^{N_\mathrm{F}}$ considered in training. The second is an internal balancing of the class $y_i^\mathrm{s} = 1$, where we must consider all the possible combinations between the number of real and fake mixed tracks. For instance, in the case we have 3 mixed tracks, within the same class $y_i^\mathrm{s} = 1$ we can have 3 synthetic tracks, 2 synthetic and 1 real, 1 synthetic and 2 real. During the training, for each N_F, we balance all these cases for each batch considered by the detector.

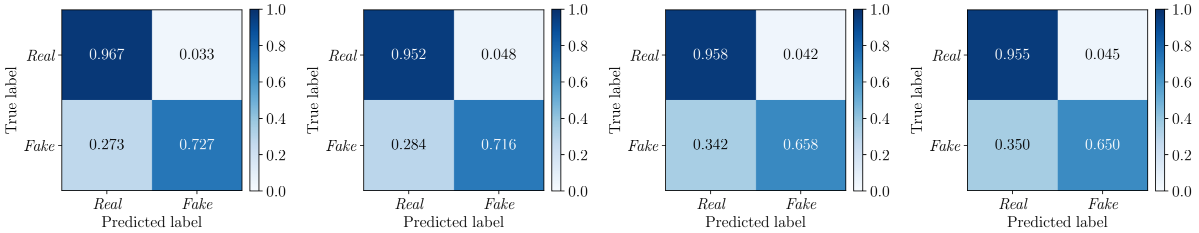

Figure 3: Confusion matrices obtained by testing the considered synthetic speech detectors $\mathcal{D}_{N_\mathrm{F}}$ on speech signals underwent different foldings N_F. From left to right $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$.

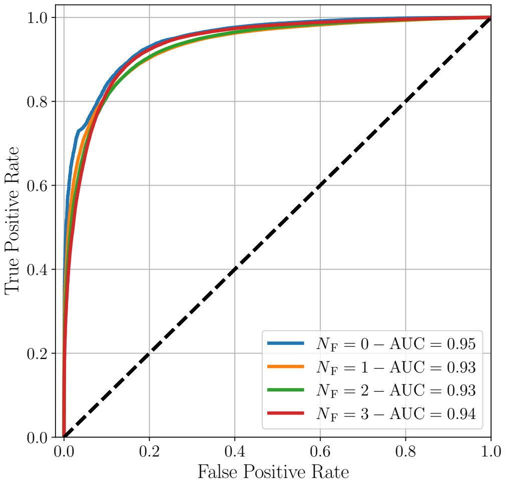

Figure 4: Receiver Operating Characteristic (ROC) curves obtained by testing the considered synthetic speech detectors $\mathcal{D}_{N_\mathrm{F}}$ on speech signals underwent different foldings N_F.

4 RESULTS

In this section we analyze and discuss the results of the proposed method of synthetic speech detection on folded audio signals. We evaluate the performances of the considered detectors in terms of ROC curves, Area Under the Curve (AUC), precision, recall and balanced accuracy. Optimal performances for all these metrics are reached when their value is equal to one.

As a first experiment we aim to test the ability of the proposed detectors in determining the authenticity of the audio tracks under analysis. Let us consider 4 different classifiers, all with the same architecture, but each one trained considering a different number of folds N_F from 0 to 3. We refer to these detectors as $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ respectively. As explained in Section 3, each detector is trained on signals obtained by mixing $N_\mathrm{F} + 1$ different tracks. We recall that we consider a signal obtained by more than one track as real ($y^\mathrm{s} = 0$) if all the tracks mixed in it are real, while it is classified as fake ($y^\mathrm{s} = 1$) if at least one of them is synthetic. In other words, after the mixing step, we can consider the output signal as an audio recording with multiple speakers speaking simultaneously. The signal will be classified as real only if all the speakers are authentic, fake otherwise.

Table 1: Evaluation of the considered synthetic speech detectors $\mathcal{D}_{N_\mathrm{F}}$ on speech signals underwent different foldings N_F.

	Bal. Accuracy	Precision	Recall	F1-Score
$N_\mathrm{F} = 0$	0.85	0.92	0.75	0.80
$N_\mathrm{F} = 1$	0.83	0.85	0.79	0.80
$N_\mathrm{F} = 2$	0.81	0.86	0.73	0.75
$N_\mathrm{F} = 3$	0.80	0.87	0.71	0.74

In this experiment, we test the detectors on data consistent with those seen in training, i.e., the number of folds N_F is the same for both training and test data. The results of the analysis are shown in Figure 3, Figure 4 and Table 1. In the first Figure, we compare the performance of the detectors by means of confusion matrices, while in the second using ROC curves. To produce the confusion matrices and compute the hard prediction scores we considered as threshold the optimal cut-off point calculated on the validation set. We did so as it mimics a real-world application, where the labels of the test data are not available to compute the optimal cut-off point on the eval set.

From the results we observe that the performances of the classifiers are comparable to each other in all the cases considered, with the AUC value that drops only by 0.02 between the case with 0 folds, corresponding to the original RawNet2 model, and the less performing one of the others. The same trend is also confirmed by the balanced accuracy value, which varies only by 0.05 between the considered cases, as shown in Table 1.

The experimental findings indicate that it is possible to perform synthetic speech detection on "folded" audio, and we can actually apply this technique. These results hold promising implications for enhancing the detection task in real-world scenarios, where the computational time required to process a given set of audio tracks often exceeds the available time.

To deepen the analysis, we perform an ablation study that encompasses all possible scenarios within the classification problem. For all the cases examined, where $N_\mathrm{F} > 0$, each window of the analyzed signal is produced by mixing $N_\mathrm{F} + 1$ distinct audio tracks. These tracks may not necessarily belong to the same class, and the number of synthetic tracks can vary from 0 to N_F. In this experiment, we use the same detectors considered in the previous case and evaluate their performance as the number of synthetic tracks in the mixture changes. We recall that we aim to classify the signal as fake when at least one of the mixed tracks is synthetic.

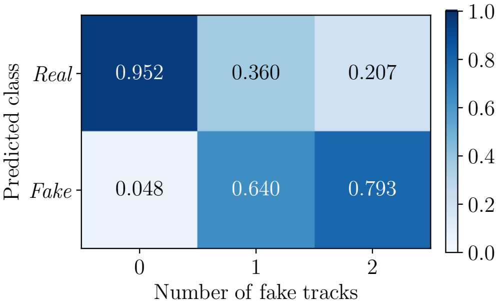

Figure 5: Confusion matrix representing the performance of the detector $\mathcal{D}_1$, broken down with respect to the number of fake tracks among the considered ones, when $N_F = 1$.

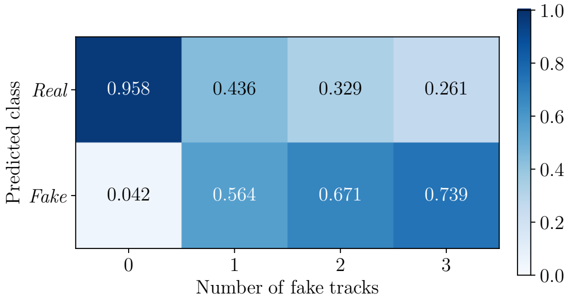

Figure 6: Confusion matrix representing the performance of the detector $\mathcal{D}_2$, broken down with respect to the number of fake tracks among the considered ones, when $N_F = 2$.

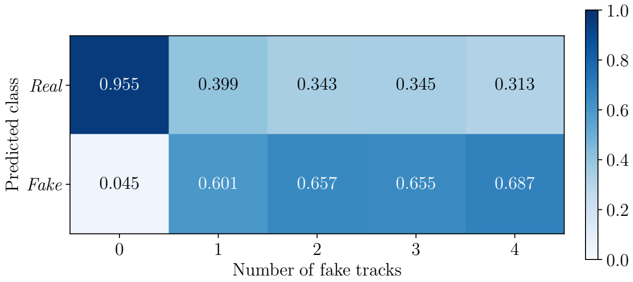

Figure 7: Confusion matrix representing the performance of the detector $\mathcal{D}_3$, broken down with respect to the number of fake tracks among the considered ones, when $N_F = 3$.

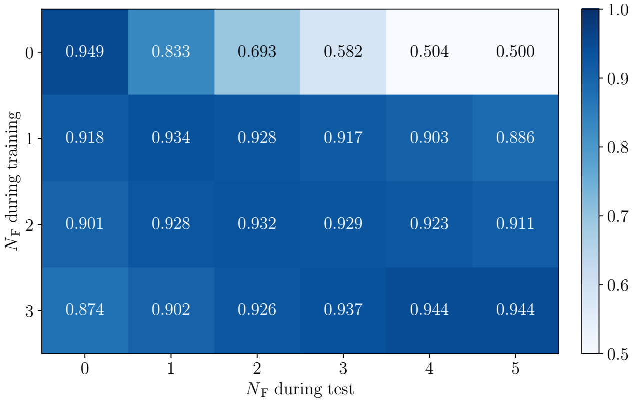

Figure 8: AUC values scored by the considered detectors $\mathcal{D}_{N_F}$ when tested on different number of mixed tracks N_F in test.

The results of this study are shown in Figure 5, Figure 6 and Figure 7, indicating the performance of $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ respectively. The confusion matrices indicate a consistent trend across all analyzed cases: as the number of fake tracks in the mix increases, the performances of the detectors improve. This phenomenon can be attributed to the growing number of artifacts introduced into the signal as the number of synthetic tracks rises, facilitating the identification process. Of particular interest is the observation that the false positive rate (i.e., cases where fake data is classified as real) remains consistently below 5%. This implies that when the detectors predict the presence of a fake track, it is highly probable that synthetic samples are actually present within the analyzed signal.

As a final experiment, we aim to explore the generalization capabilities of the proposed detectors. We do so by evaluating them considering different N_F values between training and test. This test aims to determine whether the proposed approach only works when the N_F value is the same between train and test, or we can apply a trained detector on different folded tracks. This is relevant as in a real-world scenario, we might have to reduce the computational resources required and we could not train a large number of detectors. To perform the analysis we consider the detectors trained in the previous experiments with N_F values ranging from 0 to 3. Then, we evaluate them assuming up to 6 tracks mixed together ($N_F = 5$).

Figure 8 shows the results of the analysis, evaluating the performance of the detectors by means of the AUC value. We observe a clear distinction between the performance of detector $\mathcal{D}_0$ and the other ones. The former has excellent performances in classifying tracks without any folding (AUC=0.949). However, when it is tested on mixed tracks, it sees a drop in performance, up to an AUC=0.5 when $N_F = 4$, which is equivalent to random guessing in a binary classification task as the one we are considering. In contrast, the other detectors exhibit robust generalization capabilities, as evidenced by their ability to perform well even when tested on different N_F values than those seen during training. In all the analyzed cases, the AUC value is never lower than 0.874. As expected, the best performances are obtained when N_F is the same between training and tests, except for detector $\mathcal{D}_3$, which performs better when tested on an $N_F > 3$. One possible explanation for this phenomenon is that detector $\mathcal{D}_3$ has learned to identify artifacts introduced by fake tracks. As the number of fake tracks increases, the number of artifacts also increases, thus allowing $\mathcal{D}_3$ to perform better in detecting them.

In general, the models trained on $N_F \geq 1$ show remarkable generalization capabilities, with $\mathcal{D}_2$ exhibiting an AUC value never lower than 0.9 in all the considered cases. This means that for a real-world application, it is possible to adopt a generic model trained on multiple foldings and apply it to a case with variable N_{F}, greatly simplifying the computational resources required for the detection process.

5 CONCLUSIONS

In this paper we proposed a novel approach to reduce the computational time required for analyzing extended audio tracks for the synthetic speech detection task. The proposed technique involves folding the audio track on itself, shortening its duration, and performing the detection process on the folded signal. This approach reduces the computational time by a factor proportional to the number of folds N_{F} applied to the audio track, enabling the analysis of multiple frames at once. The proposed method works for both single-class and partially fake audio tracks, as it can detect synthetic tracks even when mixed with real signals and has demonstrated remarkable results in several scenarios, up to $N_{\mathrm{F}} = 4$.

In future works, we will focus on developing more efficient systems that can further reduce the time needed to process extended audio tracks, and develop novel detectors tailored for this task to perform better in classifying mixed tracks. Additionally, we want to extend this approach to domains other than synthetic speech detection, such as Sound Event Detection (SED) and Environmental Sound Classification (ESC).

ACKNOWLEDGMENTS

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-20-2-0111, the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement numbers FA8750-20-2-1004 and HR001120C0126, and by the National Science Foundation under Grant No. 1553610. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and AFRL, the Army Research Office, the National Science Foundation, or the U.S. Government. This work was partly supported by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program.

REFERENCES

[1] Ehab A AlBadawy, Siwei Lyu, and Hany Farid. 2019. Detecting AI-Synthesized Speech Using Bispectral Analysis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Moustafa Alzantot, Ziqi Wang, and Mani Srivastava. 2019. Deep Residual Neural Networks for Audio Spoofing Detection. In *Conference of the International Speech Communication Association (INTERSPEECH)*.

[3] Tuba Arif, Ali Javed, Mohammed Alhameed, Fathe Jeribi, and Ali Tahir. 2021. Voice spoofing countermeasure for logical access attacks detection. *IEEE Access* 9 (2021), 162857–162868.

[4] Luigi Attorresi, Davide Salvi, Clara Borrelli, Paolo Bestagini, and Stefano Tubaro. 2022. Combining Automatic Speaker Verification and Prosody Analysis for Synthetic Speech Detection. In *International Conference on Pattern Recognition (ICPR)*.

[5] Clara Borrelli, Paolo Bestagini, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro. 2021. Synthetic speech detection through short-term and long-term prediction traces. *EURASIP Journal on Information Security* 2021, 1 (2021), 1–14.

[6] Emanuele Conti, Davide Salvi, Clara Borrelli, Brian Hosler, Paolo Bestagini, Fabio Antonacci, Augusto Sarti, Matthew C Stamm, and Stefano Tubaro. 2022. Deepfake Speech Detection Through Emotion Recognition: a Semantic Approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[7] Luca Cuccovillo, Christoforos Papastergiopoulos, Anastasios Vafeiadis, Artem Yaroshchuk, Patrick Aichroth, Konstantinos Votis, and Dimitrios Tzovaras. 2022. Open Challenges in Synthetic Speech Detection. In *IEEE International Workshop on Information Forensics and Security (WIFS)*.

[8] Digital Trends. 2022. AI-generated faces are taking over the internet. https://www.digitaltrends.com/social-media/ai-generated-faces-misinformation/

[9] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient Neural Audio Synthesis. *CoRR* abs/1802.08435 (2018). arXiv:1802.08435

[10] Suk-Young Lim, Dong-Kyu Chae, and Sang-Chul Lee. 2022. Detecting Deepfake Voice Using Explainable Deep Learning Techniques. *Applied Sciences* 12, 8 (2022), 3926.

[11] Youxuan Ma, Zongze Ren, and Shugong Xu. 2021. RW-Resnet: A novel speech anti-spoofing model using raw waveform. In *Conference of the International Speech Communication Association (INTERSPEECH)*.

[12] Daniele Mari, Federica Latora, and Simone Milani. 2022. The Sound of Silence: Efficiency of First Digit Features in Synthetic Audio Detection. In *IEEE International Workshop on Information Forensics and Security (WIFS)*.

[13] Nicolas M Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. 2022. Does audio deepfake detection generalize?. In *Conference of the International Speech Communication Association (INTERSPEECH)*.

[14] Sophie J Nightingale and Hany Farid. 2022. AI-synthesized faces are indistinguishable from real faces and more trustworthy. *Proceedings of the National Academy of Sciences* 119 (2022), 1–3.

[15] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker Recognition from Raw Waveform with SincNet. In *IEEE Spoken Language Technology Workshop (SLT)*.

[16] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. 2022. Exploring the Synthetic Speech Attribution Problem Through Data-Driven Detectors. In *IEEE International Workshop on Information Forensics and Security (WIFS)*.

[17] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. 2023. Reliability Estimation for Synthetic Speech Detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[18] Davide Salvi, Brian Hosler, Paolo Bestagini, Matthew C Stamm, and Stefano Tubaro. 2022. TIMIT-TTS: a Text-to-Speech Dataset for Multimodal Synthetic Media Detection. *arXiv preprint arXiv:2209.08000* (2022).

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. 2018. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[20] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. 2020. An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification. In *Speaker and Language Recognition Workshop (Odyssey)*.

[21] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. 2021. End-to-end anti-spoofing with RawNet2. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[22] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. 2019. ASVspoof 2019: Future horizons in spoofed and fake audio detection. In *Conference of the International Speech Communication Association (INTERSPEECH)*.

[23] The Verge. [n. d.]. 4chan users embrace AI voice clone tool to generate celebrity hatespeech. https://www.theverge.com/2023/1/31/23579289/ai-voice-clone-deepfake-abuse-4chan-elevenlabs.

[24] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. 2019. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit.

[25] 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. In *Automatic Speaker Verification and Spoofing Countermeasures Challenge*.

[26] Rui Yan, Cheng Wen, Shuran Zhou, Tingwei Guo, Wei Zou, and Xiangang Li. 2022. Audio Deepfake Detection System with Neural Stitching for ADD 2022. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[27] Zhenyu Zhang, Xiaowei Yi, and Xianfeng Zhao. 2021. Fake speech detection using residual network with transformer encoder. In *ACM Workshop on Information Hiding and Multimedia Security*.