POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Combining Automatic Speaker Verification and Prosody Analysis for Synthetic Speech Detection

Laurea Magistrale in Music and Acoustic Engineering

Author: Luigi Attorresi

Advisor: Prof. Paolo Bestagini

Co-advisor: Davide Salvi, Clara Borrelli

Academic year: 2020-2021

1. Introduction

With the term deepfake (DF), we refer to a category of synthetic multimedia content generated through Deep Learning (DL) techniques that depict individuals in actions and behaviors that do not belong to them. In recent years, the fast development in this technology has made it increasingly realistic and accessible, enabling producing manipulated media that are almost impossible to distinguish from original ones. These improvements result in exciting and futuristic scenarios but also represent a potential tool for malicious purposes. When misused, these technologies generate non-consensual adult material and fake political news, access the victim's personal information, commit fraud, and provide support for voice phishing attacks.
Given this threat, there is an urgent need to develop systems able to discriminate between authentic and fake media. Several state-of-the-art methods have been proposed to face this problem for both videos and audio content [5]. These can be divided into two main groups. The first one includes methods that focus on low-level signal features, looking for artifacts introduced by the generators at the pixel or sample level. The second one relies on more semantically meaningful aspects and exploits high-level inconsistencies to discriminate DFs, assuming their weakness in emulating the finest aspects of the depicted content.
In this work, we propose a DF speech detector based on a semantic approach. We partially take inspiration from the system presented in [1], where face-swap DFs are identified by looking at the mismatch between facial recognition static cues and behavioral bio-metrics based on expression and head movement. Our scenario considers speaker identification aspects and speech prosody, defined as all the information present in a speech signal but not specified in the text (e.g., temporal variations in rhythm, intonation, stress, style, etc.). This constitutes a basis we can leverage to identify DF speech generated via different technologies that may be flawed in one semantic aspect or the other. We believe that combining two semantic representations as speaker-identity and prosody can model both the voice's physiological and behavioral characteristics.

2. Proposed System

In this work, we propose a method for synthetic speech detection named *ProsoSpeaker*. This an-

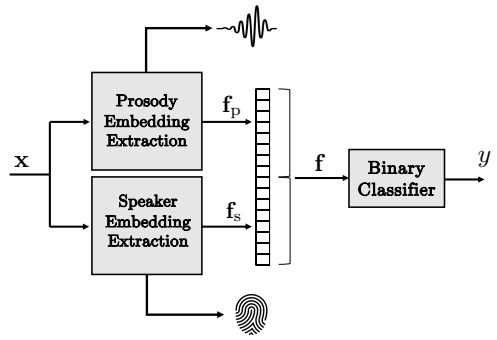


Figure 1: Pipeline of the proposed *ProsoSpeaker* system.

alyzes a given speech signal and determines if it is authentic or has been synthetically generated. Formally, given a discrete-time input speech signal $\mathbf{x}$ sampled with sampling frequency F_s, the goal is to predict the associated label y such that

$$y \in \{\text{REAL}, \text{DF}\}, \tag{1}$$

where REAL identifies authentic speech samples, while DF corresponds to speech that has been synthetically generated, either using Text-to-Speech (TTS) or Voice Conversion (VC) techniques. Figure 1 shows the pipeline of the proposed system. Starting from the input signal, we extract two different types of high-level features, which we will refer to as *speaker* ($\mathbf{f}_s$) and *prosody* ($\mathbf{f}_p$) embeddings. Then, we concatenate them and give as input to a binary supervised classifier, which predicts the label y for the signal $\mathbf{x}$. In the following, we provide additional details about each step of the pipeline depicted in Figure 1.

2.1. Speaker Embedding Extraction

The principle of VC algorithms is to operate on pristine speech signals and modify their frequency content to match a target identity. We believe that this kind of forgeries could leave traces in the speaker timbre quality that we can leverage to perform synthetic speech detection. We propose to do so through a feature set that describes each voice's unique fingerprint in a compact fashion, extracting the spectro-temporal characteristics of the analyzed spokesperson, i.e., timbre specific properties or pitch contour of the voice. This feature set, that we indicate with $\mathbf{f}_s$, is extracted exploiting a state-of-the-art network, called ECAPA-Time Delay Neural Network (TDNN) [2], originally proposed for a speaker recognition task. The proposed speaker embeddings can spot voice anomalies and allow us to discriminate between real and synthetic tracks generated through VC engines, as we will prove in the results section.

2.2. Prosody Embedding Extraction

Complementary to the aspects described above, we believe that high-level prosodic aspects, like speech signal variations in rhythm, intonation and style, constitute another aspect we can leverage to discriminate deepfake speech tracks. In particular, prosody measures an intrinsic human voice characteristic that we assume TTS synthesis algorithms struggle at recreating. In fact, despite the recent advances, synthetic prosody has different quality and intensity w.r.t. to human speech, and this difference can be captured using a set of prosody embeddings. This assumption is later proved by the presented results. The prosody embedding vector $\mathbf{f}_p$ we propose corresponds to the result of the reference encoder of the model presented in [3], which we will refer to as prosody encoder. This was originally introduced to improve the naturalness of the voices synthesized by Tacotron enhancing their prosody controls.

2.3. Binary Classifier

The final part of the *ProsoSpeaker* pipeline is a supervised binary classifier, as it is shown in Figure 1. Here, we concatenate the two embeddings $\mathbf{f}_s$ and $\mathbf{f}_p$ obtaining a final feature vector

$$\mathbf{f} = [\mathbf{f}_s, \mathbf{f}_p] \in \mathbb{R}^{N_s+N_p}, \tag{2}$$

which is fed to the classification stage. The supervised classifier is trained to predict the class y of the input speech $\mathbf{x}$. It is worth noting that any supervised classifier algorithm can be used at this stage, as our pipeline is classifier-independent.

3. Experimental Setup

In this section we provide the reader with some insights on the evaluation setup used to assess the performances of the *ProsoSpeaker* detector.

3.1. Dataset Description

We considered multiple datasets containing tracks of both REAL (i.e., authentic) and DF

(i.e., synthetic) classes, aiming to test the proposed method's generalization properties for almost 800000 tracks. We set the sampling frequency F_s to 16 kHz during all the experiments, hence if necessary, down-sampling the audio tracks.

ASVspoof 2019/2021 are speech audio datasets containing both real and synthetic tracks. They have been released for the ASVspoof challenges for the 2019 and 2021 editions. Here the participants compete to implement the best anti-spoofing system for Automatic Speaker Verification (ASV). Regarding the ASVspoof 2019 dataset, we consider the Logical Access (LA) partition, further divided in *train*, *dev* and *eval*, which includes spoofing attacks generated through TTS, VC and TTS/VC hybrid techniques. We consider *train* and *dev* partitions for training and fine-tuning the proposed method, while *eval* partition is used in test. About the ASVspoof 2021 dataset, we consider the DF partition for testing our method. It has been built by processing with different lossy codecs the data from ASVspoof 2019 LA *eval* set and additional sources.

LibriSpeech is a dataset containing about 1000 hours of authentic speech from different speakers. From this corpus we considered the subset *train-clean-100*. We include audio tracks from this dataset in the training set.

LJSpeech (LJS) is a dataset containing short audio tracks of REAL speech recorded from a single speaker. It is part of the test set.

Cloud2019 is a collection of TTS audio signals, synthesized by different speech generators available as cloud services: Amazon AWS Polly (PO), Google Cloud Standard (GS), Google Cloud WaveNet (GW), Microsoft Azure (AZ) and IBM Watson (WA). We include this dataset in the test set as DF signals.

Interactive Emotional Dyadic Motion Capture (IEMOCAP)(IEM) is a dataset originally designed for the Speech Emotion Recognition (SER) task. The data were recorded during scripted and improvised conversations by 10 actors. We include this dataset in the test set as authentic signals.

3.2. Training

Our system involves the training of three independent blocks: the ECAPA-TDNN network, the prosody encoder, the final binary classifier. Regarding the speaker embedding extractor, we use a version of ECAPA-TDNN trained with an Additive Margin Softmax Loss on VoxCeleb 1 and VoxCeleb 2 datasets. The final embedding vector $\mathbf{f}_s$ has dimension $N_s = 192$. For the prosody embedding extractor, we train the prosody encoder on Blizzard 2013 dataset, following the training procedure detailed in [3]. For computational issues, we modify only one parameter value, the mini-batch size, that in our training process is equal to 8. The resulting embedding vector $\mathbf{f}_p$ has length and $N_p = 128$. The final concatenated feature set is $\mathbf{f}$ of length $N = N_s + N_p = 320$. We standardize it using z-score, i.e., removing the mean and scaling to unit variance, and consider it as input to the binary classifier. As supervised classification algorithm we adopt a Support Vector Machine (SVM) classifier, following the training-development partition detailed in Section 3.1.

3.3. Baseline

To test the validity of our method, we compare its performances with those of RawNet2 [4], a state-of-the-art end-to-end neural network that operates on raw waveforms. It has been first proposed for the ASVspoof 2019 challenge and included as a baseline in the ASVspoof 2021 challenge both for LA and DF tasks. We trained it on the same training set we adopted for the proposed method.

4. Results

In this section we assess the performances of *ProsoSpeaker* detector, measuring the performances of the method in terms of Receiver Operating Characteristic (ROC) curves, Area Under the Curve (AUC), Equal Error Rate (EER), balanced accuracy and confusion matrices. All the models presented in the following have been trained on the same dataset, obtained by the union of ASVspoof 2019 LA and LibriSpeech, as described in Section 3.1.

4.1. Baseline comparison

As a first experiment, we compare the results obtained using the proposed method with those

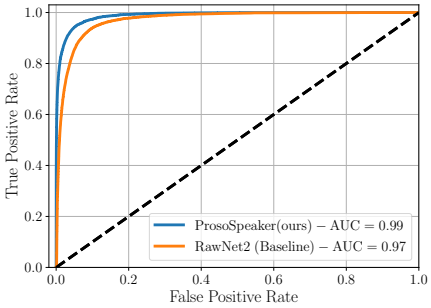


Figure 2: ROC curves for the proposed method and the considered baseline, evaluated on ASVspoof 2019 LA *eval* set.

of the considered baseline on the LA *eval* partition of the ASVspoof 2019 dataset. Figure 2 shows the ROC curves of the two detectors together with the corresponding AUC values. *ProsoSpeaker* detector outperforms the baseline in the considered metrics, improving by almost 2 over AUC. The EER and balanced accuracy improve too, going from 8.15% and 91.66% with RawNet2, to 5.39% and 94.43% with our method, respectively.

4.2. Embedding analysis and ablation study

In this second experiment we analyze in detail the characteristics and the importance of each embedding subset, namely the prosody embeddings $\mathbf{f}_\mathrm{p}$ and the speaker embeddings $\mathbf{f}_\mathrm{s}$, used in *ProsoSpeaker* method.

The first question may be how much speaker and prosody embeddings differ from each other to avoid the computation of redundant information. To do so, we measure the sample Pearson correlation coefficient $r_{f_i f_j}$ for each pair of elements (f_i, f_j) of the vector $\mathbf{f} = [f_0, f_1, ..., f_{N-1}]$ over the test dataset. The resulting matrix $\mathbf{R_{ff}}$ describes both cross-correlation between prosody and speaker embeddings $\mathbf{R}_{\mathbf{f}_\mathrm{s}\mathbf{f}_\mathrm{p}} = \mathbf{R}^T_{\mathbf{f}_\mathrm{p}\mathbf{f}_\mathrm{s}}$ both auto-correlations of each embedding vector $\mathbf{R}_{\mathbf{f}_\mathrm{p}\mathbf{f}_\mathrm{p}}$ and $\mathbf{R}_{\mathbf{f}_\mathrm{s}\mathbf{f}_\mathrm{s}}$. Figure 3 shows the results of this analysis computed in the ASVspoof 2019 *eval* partition. The diagonal has been set to 0 for visualization purposes. There, we can identify two rectangular regions, one at the top left, corresponding to $\mathbf{R}_{\mathbf{f}_\mathrm{s}\mathbf{f}_\mathrm{s}}$, and one at the bottom right, corresponding to $\mathbf{R}_{\mathbf{f}_\mathrm{p}\mathbf{f}_\mathrm{p}}$. Although the elements of $\mathbf{f}_\mathrm{p}$ have a higher degree of internal correlation than those of $\mathbf{f}_\mathrm{s}$, with mean value $\mu(\mathbf{R}_{\mathbf{f}_\mathrm{p}\mathbf{f}_\mathrm{p}}) = 0.21$ and standard deviation

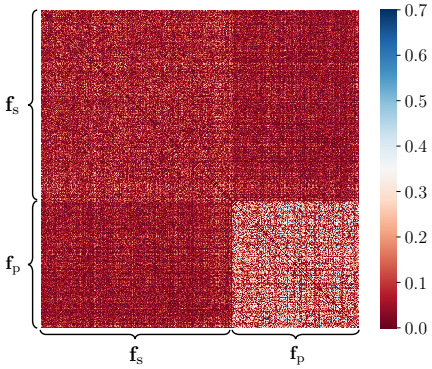
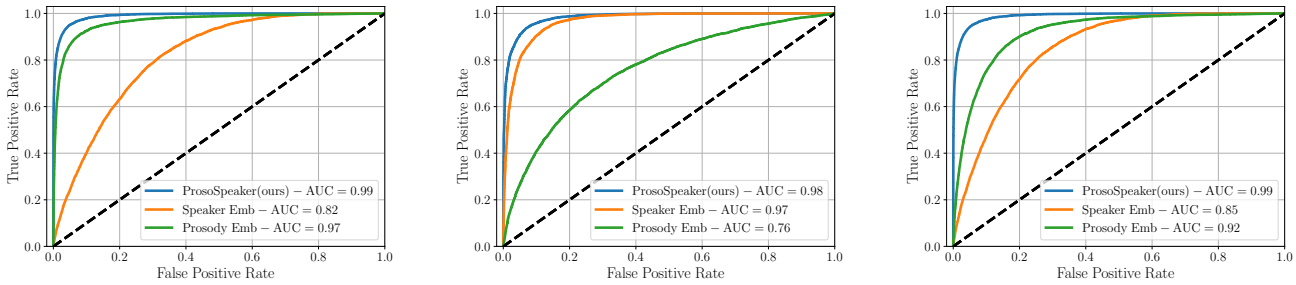


Figure 3: Cross-correlation matrix $\mathbf{R_{ff}}$ of feature vectors $\mathbf{f}$ realizations of ASVspoof 2019 *eval* set.

$\sigma(\mathbf{R}_{\mathbf{f}_\mathrm{p}\mathbf{f}_\mathrm{p}}) = 0.08$, respectively, the cross coefficients present low values, with an average value of $\mu(\mathbf{R}_{\mathbf{f}_\mathrm{s}\mathbf{f}_\mathrm{p}}) = 0.07$. This means that the two embedding vectors do not strongly correlate with each other and do not share much information. The spectro-temporal and prosodic characteristics we are considering have turned out to be orthogonal to each other, benefiting our detector.

Given these results, we test how the embedding types perform individually in different scenarios. In this analysis, we consider three distinct models, all based on the proposed architecture, differing only for the embeddings subset that the final SVM classifier receives as input. The first model, that we indicate with *Prosody Emb.*, is fully-prosodic and based on $\mathbf{f}_\mathrm{p}$ only. The second only considers the speaker information of $\mathbf{f}_\mathrm{s}$ and we indicate it as *Speaker Emb.* The third model is the complete one, i.e., *ProsoSpeaker*, and it performs classification using the concatenation of $\mathbf{f}_\mathrm{p}$ and $\mathbf{f}_\mathrm{s}$. All three models are trained on the same dataset, i.e., ASVspoof 2019 + LibriSpeech, with the same parameters. We then considered three test scenarios, depending on the synthesis techniques used to generate the synthetic speech signals of the test set. In the first scenario (a) we consider only speech tracks created with TTS techniques; in the second scenario (b) only speech tracks created with VC techniques; in the third scenario (c) both synthesis techniques are considered. All the tracks for the three scenarios are selected from ASVspoof 2019 dataset. Figure 4 shows the binary classification performances of this analysis in terms of ROC curves and associated AUC values obtained for the three models in the three test scenarios.



(a) TTS (b) VC (c) ALL

Figure 4: ROC curves obtained for the three models using different embeddings (*ProsoSpeaker*, *Speaker Emb.*, *Prosody Emb.*) and tested on the three scenarios (TTS, VC, ALL).

The predictions of the two partial models are orthogonal to each other and each performs better on a distinct scenario. In particular, prosodic embeddings $\mathbf{f}_p$ can discriminate speech signals generated with TTS algorithms well but are less effective with VC methods, while speaker embeddings $\mathbf{f}_s$ achieves better results in the VC case than TTS. From these results we can confirm our initial hypothesis, i.e., that each one of the two speech generation techniques fails in reproducing one of the semantic features encoded by $\mathbf{f}_s$ or $\mathbf{f}_p$. Nonetheless, the fusion of the two embeddings improves the predictions in all the considered scenarios, reaching an AUC = 0.99 in the case of the complete dataset. We can conclude that the concatenation of the two embeddings provides a more comprehensive and significant representation of the input speech signal, leading to higher binary classification performances.

4.3. Generalization

In this third set of experiments, we aim to analyze the consistency and generalization ability of the proposed method by augmenting the considered test set. First, we verify the performances of the proposed detector singularly on each algorithm present in ASVspoof 2019 *eval* to check the classification performances consistency over different synthesis strategies. Then, we want to assess *ProsoSpeaker*'s generalization capabilities across multiple datasets, unseen during training and external to the ASVspoof challenge corpora. Figure 5 shows the percentage of correct attribution values obtained for each synthesis algorithm included in ASVspoof 2019 *eval* set (A07, A08, ..., A13) and for LJSpeech, IEMOCAP and Cloud2019 (divided in PO, AZ, GS, GW, WA). The label AU corresponds to real speech samples

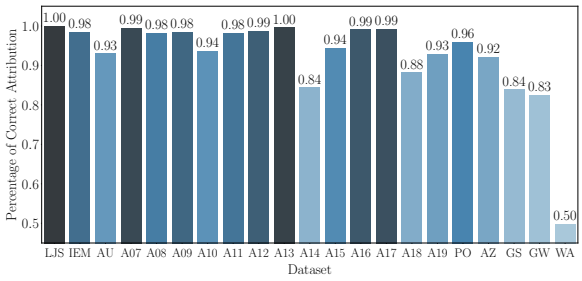


Figure 5: Bar plot of the percentage of correct attribution values of the proposed model on each partition of each considered dataset.

distributed in ASVspoof 2019. The proposed method is successful in almost all the considered cases, with a percentage of correct attribution value always higher than 0.80. This means that *ProsoSpeaker* has good generalization capabilities, and we can consider it a reliable method. The only exception is represented by the TTS generator IBM Watson, included in Cloud2019, where the accuracy is equal to 0.50. We believe this issue is due to the fact that the IBM TTS method is specifically trained considering a "prosodic-phonolog" approach for generating expressive speech, hence deceiving our detection method.

4.4. Robustness analysis

Some additional tests are finally necessary to verify the robustness of the proposed method to common signal manipulation, i.e., compression. In fact, in a real-world scenario, many operations can be performed to hide the artifacts introduced by deepfake generation algorithms, like, for instance, lossy compression. Some signal information is lost by compressing an audio track, including traces that may help deep-

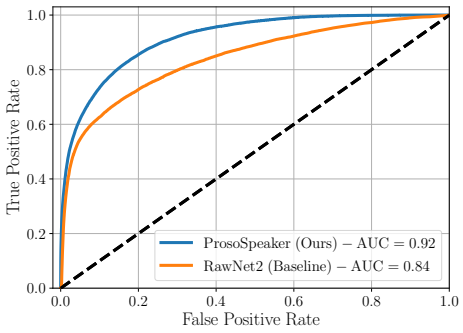


Figure 6: ROC curves of the proposed method and the considered baseline computed on ASVspoof 2021 DF *eval* set.

fake detectors determine the signal's authenticity. Since our method does not rely on low-level signal characteristics but analyzes semantic features, we hypothesize that compression should not affect its performance significantly. In practice, speaker and prosody embeddings should be only partially impacted by this type of data augmentation and keep their discriminative potential. We test such aspect on the DF partition of ASVspoof 2021, which comprises more than 600000 tracks from both REAL and DF classes, compressed with different codecs to simulate VoIP transmission. In this case, compared to the previous, we are not aware of which audio manipulation technique or parameters have been applied to the analyzed tracks. *ProsoSpeaker* method results are reported in Figure 6, together with those of the considered baseline. Our method performs significantly better with a difference of $\approx$ 7% on AUC compared to RawNet2. Similarly the EER improves by about 7%, going from 24.15% with RawNet2 to 17.16% with our method, while the balanced accuracy improves less, from 76.64% to 80.16%. The overall performance improvement on the baseline is even more significant than that obtained on ASVspoof 2019, proving great robustness of our method in a realistic and challenging scenario.

5. Conclusions

In this work we presented a novel method to perform DF speech detection based on high-level features. We have shown that the performance of the proposed system outperforms those of the state-of-the-art considered baseline. In addition to that, it presents good generalization properties and is robust to real-world audio manipulation, such as lossy compression. Moreover, through an ablation study, we observed how speaker and prosody embeddings perform individually in different scenarios and why their combination is the more effective strategy, achieving higher classification performances. The obtained results validate the idea of exploiting semantic features to discriminate deepfakes and highlight some of the aspects on which speech generators still fail.

Further studies may focus on improving the extraction of speaker and prosody embeddings to obtain a more discriminative and robust representation of the two semantic features or include additional features in the analysis, being them semantic or not.

References

[1] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020.

[2] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.

[3] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *International Conference on Machine Learning (ICML)*, 2018.

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

[5] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020.