IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 April 2023, accepted 28 April 2023, date of publication 15 May 2023, date of current version 31 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276480

RESEARCH ARTICLE

TIMIT-TTS: A Text-to-Speech Dataset for Multimodal Synthetic Media Detection

DAVIDE SALVI[1], (Student Member, IEEE), BRIAN HOSLER[2], (Student Member, IEEE), PAOLO BESTAGINI[1], (Member, IEEE), MATTHEW C. STAMM[2], (Member, IEEE), AND STEFANO TUBARO[1], (Senior Member, IEEE)

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy
[2]Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

Corresponding author: Davide Salvi (davide.salvi@polimi.it)

This work was supported in part by the Army Research Office and was accomplished under Cooperative Grant W911NF-20-2-0111; in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under Grant FA8750-20-2-1004 and Grant HR001120C0126; in part by the National Science Foundation under Grant 1553610; in part by the U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon; and in part by the PREMIER Project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 Program.

ABSTRACT With the rapid development of deep learning techniques, the generation and counterfeiting of multimedia material has become increasingly simple. Current technology enables the creation of videos where both the visual and audio contents are falsified. While the multimedia forensics community has begun to address this threat by developing fake media detectors. However, the vast majority existing forensic techniques only analyze one modality at a time. This is an important limitation when authenticating manipulated videos, because sophisticated forgeries may be difficult to detect without exploiting cross-modal inconsistencies (e.g., across the audio and visual tracks). One important reason for the lack of multimodal detectors is a similar lack of research datasets containing multimodal forgeries. Existing datasets typically contain only one falsified modality, such as deepfaked videos with authentic audio tracks, or synthetic audio with no associated video. Currently, datasets are needed that can be used to develop, train, and test these forensic algorithms. In this paper, we propose a new audio-visual deepfake dataset containing multimodal video forgeries. We present a general pipeline for synthesizing deepfake speech content from a given video, facilitating the creation of counterfeit multimodal material. The proposed method uses Text-to-Speech (TTS) and Dynamic Time Warping (DTW) techniques to achieve realistic speech tracks. We use this pipeline to generate and release TIMIT-TTS, a synthetic speech dataset containing the most cutting-edge methods in the TTS field. This can be used as a standalone audio dataset, or combined with DeepfakeTIMIT and VidTIMIT video datasets to perform multimodal research. Finally, we present numerous experiments to benchmark the proposed dataset in both monomodal (i.e., audio) and multimodal (i.e., audio and video) conditions. This highlights the need for multimodal forensic detectors and more multimodal deepfake data.

INDEX TERMS Audio, multimodal, deepfake, forensics, synthetic speech, text-to-speech, TIMIT.

I. INTRODUCTION

In recent years, deep learning technologies have rapidly advanced. This has enabled the creation of systems with humanlike performance that were unimaginable only a few years ago, such as virtual assistants powered by natural language processing and visual recognition algorithms. Similarly, these advances have also given rise to new forms of fake media, such as deepfake videos. These are videos produced through AI-driven technologies that can synthesize a person's identity or biometric aspects. While deepfake generation systems can create exciting new applications, they can also pose dangers and threats when misused. For example, deepfake techniques allow for the generation of video content that

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

depicts a victim engaging in actions or behaviors that they did not enage in in reality. This can lead to fraudulent activities, scams, and the dissemination of fake news [1], [2], [3]. This threat cannot be ignored, as we have reached a point where it is no longer always possible to distinguish real media from artificially generated ones [4], [5].

To combat the rise of malicious uses of deepfakes, the scientific community has begun working in several directions to mitigate this threat. Numerous deepfake detectors have been deeveloped to identify synthetic content in the both audio and video domains [6], [7]. The goal of these systems is to distinguish counterfeit media from pristine media. They do this by using several approaches which analyze different characteristics of the media, ranging from low-level artifacts left by the generators [8], [9] to more semantic aspects [10], [11]. To support the development of new detectors, international challenges have been organized to make people aware of the importance of fighting deepfake misuse. For instance, the DFDC challenge [12] focused on video deepfake detection, while ASVspoof [13], [14] and ADD [15] challenges have been proposed in the audio field. Additionally, part of the research community has focused on releasing deepfake datasets to help develop forensic detectors. This is the case of Faceforensics++ [16] and DeepfakeTIMIT [17] for videos, as well as WaveFake [18] for audio.

Despite the considerable effort put into fighting deepfakes, a common trait of the developed detectors is that they primarily focus on monomodal analysis: they consider either the audio or video deepfake detection problem separately. Videos, however, typically contain both audio and visual tracks, both of which are subject to editing and deepfaking. Focusing on only a single modality is an important limitation, because sophisticated forgeries may be difficult to detect without exploiting cross-modal inconsistencies (e.g. across the audio and visual tracks). Despite this, only a few approaches have been proposed to perform multimodal detection, i.e. leveraging inconsistencies or traces orthogonal to different modalities to identify counterfeit materials. For example, [19] exploits the inconsistencies between emotions conveyed by audio and visual modalities to perform a joint audio-visual deepfake detection. The authors of [20] incorporate temporal information from series of images, audio and video data to provide a multimodal deepfake detection approach. Alternatively, the authors of [21] show that combining audio and video baselines in an ensemble-based method provides better detection performance than a monomodal system.

The main reason for the lack of multimodal forensic systems for deepfake detection is the scarcity of data to train and test them. Most of these systems are data-driven and require a large amount of data to be trained. Still, most of the deepfake datasets proposed in the literature are monomodal. There is a dearth of challenging fake video datasets that also contain fake audio, making it difficult to develop, train, and evaluate the performance multimodal forensic systems.

In this paper we address the lack of multimodal deepfake datasets by first proposing a pipeline to generate forged multimodal data from deepfake videos, then by using this pipeline to create a new multimodal video forgery dataset.

The primary contributions of this work are the following:

- We propose a general pipeline to turn a monomodal video deepfake dataset from the literature into a multimodal audio-visual deepfake dataset.
- We use this pipeline to generate synthetic speech from 12 different Text-to-speech (TTS) systems, providing an overview of the most advanced techniques in state-of-the-art as well as standard tools that can be used even by non-expert attackers.
- We include a Dynamic Time Warping (DTW) step to increase the realism of the generated tracks when paired with videos (i.e., lip-sync must be guaranteed).
- We apply the proposed pipeline to the VidTIMIT [22] and DeepfakeTIMIT [17] datasets in order to build and release the novel multimodal TIMIT-TTS deepfake dataset containing almost 80 000 tracks.
- We benchmark the generated dataset by running a series of deepfake detection baselines that highlight the main challenges for future research.

The rationale behind our proposed pipeline is that while realistic deepfake video datasets have been proposed in the literature, these datasets do not contain accompanying deepfake audio. We present a technique that can be used to augment these datasets, or create new ones, by generating a synthetic speech track for a given input video. This approach allows us to generate fake audio content starting from any video containing speech, considering the most advanced state-of-the-art TTS systems. Once generated, the synthetic track can be paired with the input video and, depending on the authenticity of the latter, an audio-only or an audio-visual deepfake is generated. Our pipeline thus provides a viable solution for making counterfeit multimodal materials, which is in general complex to perform.

To showcase the actual feasibility of the proposed deepfake generation approach, we apply it to the VidTIMIT dataset [22] and DeepfakeTIMIT dataset [17]. The former contains audio-video recordings of 43 people speaking. The latter is a video deepfake version of the former. By generating synthetic speech for both video datasets, we end up with the proposed TIMIT-TTS, a synthetic speech dataset built using state-of-the-art TTS techniques. On the one hand, TIMIT-TTS can be used as a standalone audio dataset to test the developed speech deepfake detectors, as it contains the most cutting-edge methods in the synthetic speech synthesis field. On the other hand, TIMIT-TTS can also be combined with VidTIMIT and DeepfakeTIMIT to provide multimodal audio-video deepfake data, which is an overlooked aspect in the current literature.

Finally, we run a series of tests to provide some information on the challenges proposed by this new multimodal dataset. We adopt the video deepfake detector proposed in [23] and

the audio deepfake detector proposed in [24] to analyze videos and audio tracks in both monomodal and multimodal fashion. Results confirm that multimodal deepfake analysis should be preferred and show that audio deepfake attribution is an interesting topic for further research.

The rest of the paper is structured as follows. Section II recap the motivations behind our work and provides the reader with some helpful knowledge on generation and detection methods for speech deepfakes. Section III describes the proposed generation pipeline for the deepfake audio tracks and provides an overview of the considered TTS synthesis algorithms. Section IV explains the structure of the released TIMIT-TTS dataset. Section V presents the results of the analysis conducted on the released data. Finally, Section VI concludes the paper along with a brief discussion of possible future work.

II. BACKGROUND

This section provides the reader with some helpful background information needed to understand the primary rationale behind our proposal. First, we show the limitations of state-of-the-art multimodal datasets to highlight the need for a novel deepfake dataset as the one proposed in this paper. Then, we provide a quick overview of synthetic speech generation and detection techniques, which are at the base of our proposed dataset and benchmarking work.

A. EXISTING MULTIMODAL DATASETS

Numerous deepfake datasets have been proposed in recent years, both in audio and video domains, significantly pushing research toward the development of new methods for recognizing counterfeit material. The publication of these sets leads to designing more innovative and effective detectors since they provide new data on which to train and test them. However, most of the presented datasets focus only on a single modality at a time, resulting in valuable data for producing monomodal detectors but not relevant for multimodal methods. Indeed, to train and test multimodal detectors, there is a need for data that are altered in all the considered aspects (e.g., both video and audio). The lack of this data is one of the main reasons behind the lack of multimodal detectors investigations and is the primary motivation behind this work.

Recently, two multimodal deepfake datasets have been proposed, both containing counterfeit audio and video. These are DFDC [12] and FakeAVCeleb [25]. FakeAVCeleb contains 500 real videos extracted from the VoxCeleb2 corpus [26], used as a base set to generate around 20 000 deepfake videos using various deepfake generation methods. DFDC contains nearly 120 000 videos generated using eight different deepfake generation methods. Among these videos, 100 000 are labeled as Fake, and the rest as Real. The authentic videos in the DFDC dataset were captured in different environmental settings.

Although these propose a solution to the abovementioned problem, we cannot define either of these as complete, especially from an audio point of view. On one side, DFDC does not provide labels as to which of the audio or visual components are fake, but the content is labeled as fake when at least one of the two modalities is counterfeit. Therefore we do not have sufficient information to perform ablation studies on different scenarios (e.g., fake audio and real video or vice versa) and investigate which aspects the detector leverages to discriminate between real and altered data. On the other hand, the multimodal deepfakes contained in FakeAVCeleb are generated overlooking the audio modality. All the fake audio tracks are synthesized using the same TTS algorithm, and none of them is synchronized with the corresponding video. This results in a lack of both variety and realism in the released data.

These issues highlight the need for a novel multimodal deepfake dataset.

B. SPEECH DEEPFAKE GENERATION METHODS

Deepfake content generation techniques are becoming increasingly simple to use and the data they produce are getting more and more realistic. In some cases, the generated synthetic material is so lifelike that it is difficult to discern from an authentic one [4]. Although this is true for both audio and video data, here we focus on the generation methods of speech deepfakes, which are the main subject of study in this paper.

As far as synthetic speech data generation is concerned, techniques can be broadly split into two main families: TTS methods and Voice Conversion (VC) methods. The difference between these two kinds of techniques is mainly the input of the generation system. TTS algorithms produce speech signals starting from a given text. Conversely, VC methods take a speech signal as input and alter it by changing its style, intonation or prosody, trying to mimic a target voice.

Regarding TTS methods, a long history of classical techniques based on vocoders and waveform concatenation has been proposed in the literature [27]. However, the first modern breakthrough that significantly outperformed all the classical methods was introduced by WaveNet [28], a neural network for generating raw audio waveforms capable of emulating the characteristics of many different speakers. This network has been overtaken over the years by other systems [29], [30], which made the synthesis of highly realistic artificial voices within everyone's reach.

Most TTS systems follow a two-step approach. First, a model generates a spectrogram starting from a given text. Then, a vocoder synthesizes the final audio from the spectrogram. This approach allows combining different vocoders for the same spectrogram generator and vice versa. Alternatively, some end-to-end models have been proposed, which generate speech directly from the input text [31].

Considering VC algorithms, the earliest models were based on spectrum mapping using parallel training data [32], [33]. However, most of the current approaches are Generative Adversarial Network (GAN)-based [34], [35], allowing to learn a mapping from source to target speaker without relying on parallel data.

In this work we only consider TTS methods as they are more investigated in the literature and allow us to build a more varied dataset. Indeed, while VC systems can be effective in dealing with problems such as the one proposed, i.e., generating deepfake audio for a given video, we decided to use TTS methods for two main reasons. First, VC systems are complex to tune and require a considerable effort to generate a dataset with multiple speakers and several generation techniques like the one we present. Secondly, in the case of a real deepfake attack, a TTS system is more likely to be adopted. This is because it allows us to have greater freedom on the attack performed, generating speech from a simple text. On the other hand, with a VC system, we would have to record a track and edit it, resulting in a more unhandy pipeline. Nevertheless, also VC methods are worth further studies and will be the subject of future versions of this dataset.

C. SPEECH DEEPFAKE DETECTION METHODS

The speech deepfake detection task consists in determining whether a given speech track x is authentic from a real speaker or has been synthetically generated. Recently, this has become a hot topic in the forensic research community, trying to keep up with the rapid evolution of counterfeiting techniques [36].

In general, speech deepfake detection methods can be divided into two main groups based on the aspect they leverage to perform the detection task. The first focuses on low-level aspects, looking for artifacts introduced by the generators at the signal level. In contrast, the second focuses on higher-level features representing more complex aspects as the semantic ones.

As an example of artifacts-based approaches, [37] aims to secure Automatic Speaker Verification (ASV) systems against physical attacks through channel pattern noise analysis. In [38], the authors assume that a real recording has more significant non-linearity than a counterfeit one, and they use specific features, such as bicoherence, to discriminate between them. Bicoherence is also employed in [39] along with several features based on modeling speech as an auto-regressive process. The authors investigate whether these features complement and benefit each other. Alternatively, the authors of [24] propose an end-to-end network to spot synthetic speech, while those of [40] perform the detection task based on the use of MFCC features and an SVM.

On the other hand, detection approaches that rely on semantic features are based on the hypothesis that deepfake generators can synthesize low-level aspects of the signals but fail in reproducing more complex high-level features. For example, [41] exploits the deepfake detection task by relying on classic audio features inherited from the music information retrieval community. The authors of [42] exploit the lack of emotional content in synthetic voices generated via TTS techniques to recognize them. Finally, in [43] ASV and prosody features are combined to perform synthetic speech detection.

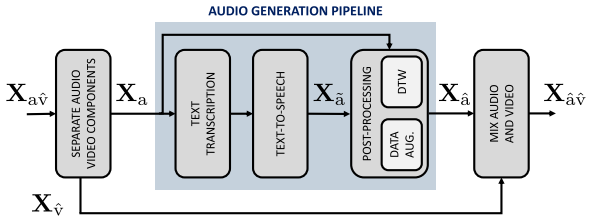


FIGURE 1. Pipeline of the proposed generation method.

III. DATASET CREATION METHODOLOGY

This section presents the methodology we propose to generate a deepfake speech track for a given input video, being the video real or fake. In doing so, we also detail all the implemented TTS systems used to synthesize the signals and the techniques applied to post-process them. This is the pipeline we follow to generate the proposed dataset. We generated synthetic speech faster than real-time using a server with an Intel Core i9 CPU with 36 cores and a single Nvidia Titan RTX GPU.

A. GENERATION PIPELINE

The proposed pipeline to generate a synthetic speech track for a given video comprises several steps, as it is shown in Figure 1. The input to the whole process consists of a video $\mathbf{X}_{av}$ that represents a speaking person. Here we consider a video $\mathbf{X}_{av}$ as a multimedia object composed of both an audio speech content $\mathbf{X}_{a}$ and a visual component $\mathbf{X}_{v}$ depicting a person's face, as in

$$\mathbf{X}_{av} = \mathbf{X}_{a} \oplus \mathbf{X}_{v}, \tag{1}$$

where $\oplus$ is the mixing operation between the audio and visual signals. The visual component of the input can be both real $\mathbf{X}_{v}$ or fake $\mathbf{X}_{\hat{v}}$. Depending on that, the output will be a monomodal ($\mathbf{X}_{\hat{a}v}$) or multimodal ($\mathbf{X}_{\hat{a}\hat{v}}$) deepfake. Here we consider as input a counterfeit video $\mathbf{X}_{a\hat{v}}$, since we aim at generating fake multimodal data. Our final goal is to produce a forged video $\mathbf{X}_{\hat{a}\hat{v}}$ containing the same visual subject as $\mathbf{X}_{\hat{v}}$ but where the speech track $\mathbf{X}_{\hat{a}}$ is a deepfake synthetically generated. To summarize, we can write

$$\mathbf{X}_{\hat{a}\hat{v}} = \Lambda(\mathbf{X}_{a\hat{v}}), \tag{2}$$

$$\mathbf{X}_{\hat{a}} \oplus \mathbf{X}_{\hat{v}} = \Lambda(\mathbf{X}_{a} \oplus \mathbf{X}_{\hat{v}}), \tag{3}$$

where $\Lambda(\cdot)$ indicates the complete pipeline we propose.

To achieve our goal, the first operation we perform is to split $\mathbf{X}_{a\hat{v}}$ into its components $\mathbf{X}_{a}$ and $\mathbf{X}_{\hat{v}}$. The speech track $\mathbf{X}_{a}$ becomes the input of the *audio generation pipeline*, which outputs its synthetic counterpart $\mathbf{X}_{\hat{a}}$. This segment is composed of three main blocks. The first is a speech-to-text algorithm, which transcribes the speech content of $\mathbf{X}_{a}$ into a text string. The second block is a TTS algorithm that produces a synthetic audio track $\mathbf{X}_{\tilde{a}}$ from a given string. Finally, the third block consists of a post-processing step, which takes the generated track $\mathbf{X}_{\tilde{a}}$ as input and outputs its processed version $\mathbf{X}_{\hat{a}}$, which is more realistic and challenging

to discriminate for deepfake detectors. $\mathbf{X}_{\hat{a}}$ is the deepfake version of the $\mathbf{X}_a$ input speech track.

Two different post-processing techniques are implemented in our pipeline, which can be applied individually or together. In case neither is applied, we output the clean signal $\mathbf{X}_{\hat{a}} = \mathbf{X}_{\tilde{a}}$. The first technique is speech-to-speech synchronization based on DTW. Since the goal of the proposed system is to generate a fake speech track for a given video, we need the synthesized audio to be synchronized with the video itself. Without performing the alignment, the synthetic track $\mathbf{X}_{\hat{a}}$ will have a different temporal trend from the input audio $\mathbf{X}_a$ and the corresponding video $\mathbf{X}_{a\hat{v}}$. This results in a deepfake that is very easy to detect for all the systems trained to analyze the discrepancies in time between the audio and video modalities. This pipeline step takes as input the two audio signals $\mathbf{X}_{\tilde{a}}$ and $\mathbf{X}_a$ and performs time warping on the former by mapping it to the latter. We do so through the alignment algorithm presented later in this section. The output track $\mathbf{X}_{\hat{a}}$, being synchronized with $\mathbf{X}_a$, is also synchronized with the input video $\mathbf{X}_{a\hat{v}}$.

The second block of the post-processing step consists of data augmentation. Here we apply several algorithms, including noise injection, pitch shifting and lossy compression, to make the generated data more challenging to discriminate for those deepfake detectors that are not robust to such operations. In fact, these processing operations hinder the traces that TTS algorithms could leave, making the generated data tougher to identify. Finally, once the audio track $\mathbf{X}_{\hat{a}}$ has been obtained, we mix it with the input video $\mathbf{X}_{\hat{v}}$ generating a new multimodal deepfake content $\mathbf{X}_{\hat{a}\hat{v}} = \mathbf{X}_{\hat{a}} \oplus \mathbf{X}_{\hat{v}}$. We remind that, depending on the authenticity or not of the input video, the output will be a mono or multimodal deepfake.

B. SPEECH SYNTHESIS

In the proposed pipeline, the TTS block can support multiple speech generation algorithms. We did so to add the possibility of generating data with different characteristics, not related to a single algorithm and more representative of the state-of-the-art. Most of the considered TTS algorithms follow a two-stage pipeline, while only a few methods have an end-to-end approach, generating speech signals directly from an input text. In the two-stage case, the first block takes a text as input and generates a spectrogram, while the second is a vocoder that sonifies the output of the first step. The two blocks are independent from each other and we can potentially use different vocoders for the same spectrogram generator. Here we consider a TTS method as a fixed pair of generator and vocoder. Even though this interchangeability allows us to potentially have a large number of methods, in this study we want to limit the number of vocoders considered. We do so since we want to keep the differences between the generated speech tracks primarily attributable to the spectrogram generators. Nevertheless, the artifacts introduced by the vocoders are noteworthy and will be the subject of subsequent versions of this dataset.

Here is a list of the considered spectrogram generators.

- **Tacotron** [30] is a seq2seq model, which includes an encoder, an attention-based decoder, and a post-processing net. Both the encoder and decoder are based on Bidirectional GRU-RNN. We consider the version implemented in [44].
- **Tacotron2** [45] has the same architecture as Tacotron but improves its performance by adding a Location Sensitive Attention module to connect the encoder to the decoder.
- **GlowTTS** [46] is a flow-based generative model. It searches for the most probable monotonic alignment between text and the latent representation of speech on its own, enabling robust and fast TTS synthesis.
- **FastSpeech2** [47] is composed of a Transformer-based encoder and decoder, together with a variance adaptor that predicts variance information of the output spectrogram, including the duration of each token in the final spectrogram and the pitch and energy per frame.
- **FastPitch** [48] is based on FastSpeech, conditioned on fundamental frequency contours. It predicts pitch contours during inference to make the generated speech more expressive.
- **TalkNet** [49] is consists of two feed-forward convolutional networks. The first predicts grapheme durations by expanding an input text, while the second generates a Mel-spectrogram from the expanded text.
- **MixerTTS** [50] is based on the MLP-Mixer architecture adapted for speech synthesis. The model contains pitch and duration predictors, with the latter being trained with an unsupervised TTS alignment framework.
- **MixerTTS-X** [50] has the same architecture as MixerTTS but additionally uses token embeddings from a pre-trained language model.
- **VITS** [51] is a parallel end-to-end TTS method that adopts variational inference augmented with normalizing flows and an adversarial training process to improve the expressive power of the generated speech.
- **SpeedySpeech** [52] is a student-teacher network capable of fast synthesis, with low computational requirements. It includes convolutional blocks with residual connections in both student and teacher networks and uses a single attention layer in the teacher model.
- **gTTS** [53] (*Google Text-to-Speech*) is a Python library and CLI tool to interface with Google Translate's text-to-speech API. It generates audio starting from an input text through an end-to-end process.
- **Silero** [54] pre-trained enterprise-grade TTS model that works faster than real-time following an end-to-end pipeline.

Regarding vocoders, we decided to stick with two of the most used and known systems in the literature, which are among the most realistic to find in real-case scenarios. In particular, we consider the two following vocoders.

- **MelGAN** [55] is a GAN model that generates audio from mel-spectrograms. It uses transposed convolutions

to upscale by the mel-spectrogram to audio. We considered this vocoder to generate speech from Tacotron2, GlowTTS, FastSpeech2, FastPitch, TalkNet, MixerTTS, MixerTTS-X, and SpeedySpeech.
- **WaveRNN** [56] is a single-layer recurrent neural network with a dual softmax layer, able to generate audio 4× faster than real-time. We considered this vocoder to generate audio from Tacotron.

Most of the models mentioned above follow a deep-learning approach and the data they generate is highly dependent on the one seen during the training phase. This also affects the speakers' number and identity that a model supports. In fact, if a system has been trained with numerous speakers, it will also be able to reproduce them at inference time, resulting in a multi-speaker generator. Conversely, if we train a system on one speaker only, it will be able to generate audio only with that tone of voice.

Here is a list of the datasets considered for training the used TTS methods in order to obtain different voice styles.
- **LJSpeech** [57] is a dataset containing short audio tracks of speech recorded from a single speaker reciting pieces from non-fiction books.
- **LibriSpeech** [58] is a dataset that contains about 1000 hours of authentic speech from more than 200 different speakers.
- **CSTR VCTK Corpus** [59] (Centre for Speech Technology Voice Cloning Toolkit) is a dataset that includes speech data uttered by 109 native speakers of English with various accents. Each speaker reads about 400 sentences from a newspaper and a passage intended to identify the speaker's accent.

Table 1 presents a summary of the datasets used to train each algorithm, together with the implemented number of speakers in TIMIT-TTS. The models trained on LibriSpeech and VCTK support multi-speaker synthesis, while those trained on LJSpeech only support a single speaker, which is an English female voice with an American accent. For gTTS, no dataset is indicated as it directly interfaces with Google Translate's TTS API and synthesizes speech using its pre-trained models. This model supports English in 4 different accents (United States, Canada, Australia and India). We highlight this aspect since data-driven vocoders are slightly influenced by the voices they have seen during training, and some artifacts could arise when dealing with the multi-speaker case. For this reason, it is relevant to remark on which datasets each model was trained on. Finally, it is worth noting that several methods have been trained on LJSpeech, resulting in diverse systems able to generate speech with the same voice. This allows the generation of speech data that are not biased by the speaker's identity and that are more difficult to discriminate by deepfake detectors, as shown in Section V.

TABLE 1. Datasets used to train each TTS method and considered number of speakers in TIMIT-TTS. The total number of speakers in the released corpus is 37.

Generator	Dataset	Num. Speakers
gTTS	//	4
Tacotron	LibriSpeech	8
GlowTTS	LJSpeech, VCTK	9
FastPitch	LJSpeech, VCTK	9
VITS	LJSpeech, VCTK	9
FastSpeech2	LJSpeech	1
MixerTTS	LJSpeech	1
MixerTTS-X	LJSpeech	1
SpeedySpeech	LJSpeech	1
Tacotron2	LJSpeech	1
TalkNet	LJSpeech	1
Silero	LJSpeech	1

C. AUDIO-VIDEO SYNCHRONIZATION

To generate a realistic audio-video deepfake, we need its audio and visual components to be synchronized with each other. This is crucial as diverse semantic deepfake detectors leverage the inconsistencies between the two modalities to discriminate among authentic and counterfeited media contents [60] and having the two components asynchronous would result in deepfake easy to spot. To avoid this, we synchronize the generated TTS track $\mathbf{X}_{\tilde{a}}$ with the original audio $\mathbf{X}_{a}$ of the input video. Since $\mathbf{X}_{a}$ is aligned with the original video component $\mathbf{X}_{v}$, the aligned TTS signal $\mathbf{X}_{\hat{a}}$ turns out to be synchronized with the video itself.

We address this point using the Dynamic Time Warping (DTW) implementation provided by Synctoolbox library [61]. This toolbox integrates and combines several techniques for the given task, such as multiscale DTW, memory-restricted DTW, and high-resolution music synchronization. The method used was initially proposed for synchronizing music, but we also tested its effectiveness in the case of speech. The DTW process computes the chroma features of the analyzed tracks and warps them by bringing them into temporal correspondence. The pipeline block inputs the original speech track $\mathbf{X}_{a}$, together with the TTS track $\mathbf{X}_{\tilde{a}}$, and outputs the processed signal $\mathbf{X}_{\hat{a}}$. In particular, $\mathbf{X}_{a}$ is the target signal and $\mathbf{X}_{\tilde{a}}$ is the one to be warped. In our pipeline, both $\mathbf{X}_{a}$ and $\mathbf{X}_{\tilde{a}}$ contain the exact text and the output $\mathbf{X}_{\hat{a}}$ has the same length as the target $\mathbf{X}_{a}$.

To improve the performance of this pipeline block we adopt a combined method of Voice Activity Detector (VAD) + DTW. In fact, in real cases, audio tracks often contain silences at their beginning or end, differently from TTS signals where silences are limited. These silences can ruin the synchronization performances of the two tracks, as they are not symmetric. To bypass this problem, we apply a VAD on both tracks before the alignment, removing the head and tail silences. Then, we perform the DTW only on the voiced segments. Finally, we add the silences removed from the target track to the warped one, obtaining a signal of the desired length. This approach allows us to achieve more effective alignments and more realistic results. Figure 2 shows the complete pipeline of the alignment block.

We underline that although the alignment performances are excellent in the dataset presented, this pipeline is not optimal

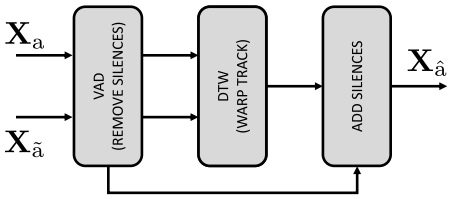


FIGURE 2. Pipeline of the speech-to-speech alignment block.

in all cases. The best results are obtained in a scenario with audio tracks of short sentences spoken regularly, which is the exact case we are working on. This is because, in this circumstance, the effort required for the alignment system is minimized. On the other hand, some artifacts may arise when dealing with more complex scenarios, and more elaborated interpolation techniques may be required. The analysis of these cases will be the subject of future versions of the dataset.

D. POST-PROCESSING

Deepfake audio detectors generally perform very well when dealing with clean data, but their performance drops as these are post-processed. When dealing with in-the-wild conditions, post-processing techniques are introduced to hide some artifacts present in the generated deepfake audio tracks. For example, applying MP3 compression reduces the audio quality and hides some defects, while adding reverberation simulates the environment in which the audio was captured. In our pipeline, we introduce a data augmentation block that allows us to generate more challenging data. Table 2 shows the techniques we implemented and the parameters we considered for each transform, as shall be better explained in the next section. We performed all the operations using the Python library audiomentations [62].

The augmented data we release has two different purposes, depending on how it is used. On one side, when included in the training process of a deepfake detection model, they help make it more reliable and robust, as shown in [63]. On the other hand, if the augmented data are included in the test set, we can use them to sample the performance of the proposed deepfake detectors in a scenario that is as broad as possible. When we test a detector in a real-world scenario, we do not know the exact post-processing pipeline used on the input data. For this reason, we have to make our systems as robust as possible to all potential attacks.

The second case is the one we consider in the experiments of the following sections, where we applied the same post-processing operations to both fake and real data. This is because these techniques aim to blur the differences between the two classes as much as possible. So, to make them effective, we have to apply them to both types of data.

IV. TIMIT-TTS DATASET GENERATION

This section provides all the details about the TIMIT-TTS dataset we release in this paper. After explaining its generation process, we illustrate its structure and possible applications.

A. REFERENCE DATASET

To generate a counterfeit speech dataset through the pipeline proposed in Section III, we need to define an audio-video set to use as a reference. Our goal is to produce a new version of the dataset where its audio component is replaced with a synthetic one. Here we consider the VidTIMIT dataset [22], [64]. This includes video and audio recordings of 43 people reciting 10 short sentences from the TIMIT Corpus [65], for a total of 430 videos. We chose to use VidTIMIT since it is state-of-the-art and highly regarded within the scientific community. Moreover, a counterfeited version of this dataset was released in 2018. This is called DeepfakeTIMIT [17] and includes 320 videos extracted from the VidTIMIT corpus modified using open-source software based on GANs to create video deepfakes. Being the released TIMIT-TTS an audio deepfake version of VidTIMIT, when it is used together with DeepfakeTIMIT, it provides audio-video content that is counterfeited in both modalities. This is extremely useful for the development of new multimodal deepfake detectors.

We extracted the text transcripts from the VidTIMIT tracks using the Speech-To-Text system Whisper from OpenAI [66], obtaining a Word Error Rate (WER) of 15.8%. This is an excellent result for this task, considering that the recordings under analysis were acquired in an office environment using a broadcast-quality digital video camera, resulting in noisy audio tracks that were difficult to transcribe. However, since all the considered sentences are extracted from the TIMIT Corpus, we are provided with all the transcripts of the video dialogues. This allows us to skip the text transcription step of the pipeline (see Figure 1), which could introduce even minimal errors within the generated tracks, undermining the reliability of the released dataset. Finally, the use of the official transcripts makes the generated speech perfectly synchronizable with the video, thus putting us in the most challenging forensic scenario where audio and video inconsistencies are minimal.

B. GENERATED DATASET

To develop the TIMIT-TTS dataset, we consider the whole VidTIMIT corpus. We generate a set of 430 synthetic speech tracks for each of the implemented generators, containing the same sentences as the reference videos. For the systems that support multispeaker synthesis, we synthesize a set of 430 tracks for each speaker. We created several versions of the dataset corresponding to the different post-processing operations we apply to the generated speech tracks. In particular, we consider two different processes: audio-video synchronization (DTW) and data augmentation. This results in the following four versions of the dataset:

- ***clean_data***: all the synthetic audio tracks are clean and no post-processing is performed after the TTS generation process.
- ***dtw_data***: DTW is applied to the generated data. Each speech is synchronized with the corresponding video track from VidTIMIT.

TABLE 2. List of the implemented data augmentation techniques.

Augm. technique	Parameter	Application range
Gaussian Noise	a - Amplitude	$e^{-3} < a < 1.5e^{-2}$
Time Stretching	r - Rate	$0.8 < r < 1.25$
Pitch Shifting	s - Semitones	$-8 < s < 8$
High-pass Filtering	f - Cutoff freq. [Hz]	$20 < f < 2400$
MP3 compression	a - Bitrate	$8 < b < 64$

- ***aug_data***: data augmentation is applied to each speech track.
- ***dtw_aug_data***: both DTW and data augmentation are applied to the generated data. First, we warp the tracks in time and then we augment them. We do so to prevent degradation from affecting the alignment process.

Considering the number of TTS methods and the number of speakers implemented, as shown in Table 1, each dataset partition is composed of 19 780 tracks, for a total of almost 80 000 speech signals on the entire dataset. All the tracks are released in *wav* format considering a sampling rate of 16 kHz. The complete dataset can be downloaded at this link,[1] while some examples of generated data, together with audio-video samples, can be found here.[2]

Each partition of the dataset contains two splits, named *single_speaker* and *multi_speaker*. The first one includes all the tracks generated using TTS algorithms that support LJSpeech's speaker. The second includes the signals generated from the generators that implement speakers from datasets other than LJSpeech. Each of the two splits contains a subfolder for each generator, where the audio tracks are stored. The name of each track is *dir_track.wav*, where *dir* and *track* are respectively the names of the directories in which VidTIMIT is structured and of the tracks it contains. We adopted this naming to make it easy to link each deepfake audio track with its corresponding video.

Regarding data augmentation, we applied all the implemented techniques to each speech track, with a probability $p = 0.3$ and a random value contained in a specific range for each method. Following this application approach, some generated tracks will be edited with more than one method at a time, while others will remain clean. At the same time, different augmentation levels will be considered for each track. This results in a dataset that is highly diverse and challenging to identify. Table 2 shows all the augmentation techniques implemented, together with their considered ranges, while a list of the augmentation techniques applied to each signal can be found in a *csv* file included in the partition folder.

The possible applications of TIMIT-TTS are numerous. As regards synthetic speech detection, it is possible to perform that both in closed and open set scenarios. The high number of TTS generators implemented within the dataset allows us to include some of them in the train set while introducing others only in the test partition, making the classification task more challenging. Furthermore, apart from binary classification, synthetic speech attribution can be performed. This consists of a multi-class classification problem, where for each of the proposed tracks, it is required to find the TTS generation algorithm used to synthesize it. Performing this study on TIMIT-TTS is fascinating since several of the proposed spectrogram generators only support the LJSpeech speaker. Indeed, synthetic speech attribution could be relatively easy to perform when each generator supports different speakers, but it becomes challenging when all the systems reproduce the same speaker. This type of analysis is presented in Section V.

[1] https://zenodo.org/record/6560159
[2] https://polimi-ispl.github.io/TIMIT-TTS/

V. RESULTS AND BENCHMARKING

In this section, we benchmark the released dataset using subjective and objective metrics and show some of its possible applications, presenting the results obtained by testing it with state-of-the-art deepfake detectors. We perform deepfake detection in both monomodal and multimodal scenarios, showing the effectiveness of considering multiple modalities at the same time.

A. TIMIT-TTS STATISTICS

When generating synthesized audio data, many aspects need to be addressed to ensure the forged material is reliable and realistic. These aspects include track length, silence duration, speech naturalness and number of supported speakers. Overlooking these aspects, we risk generating biased or easy-to-discriminate data.

The first aspect we analyze is the duration of the generated audio tracks. As the dataset will be mainly used to develop deepfake detectors, we need the length of the audio tracks to be compatible with the window sizes used by most of the systems. Furthermore, we want to avoid differences between the duration of the signals generated with distinct TTS algorithms to prevent tracks generated by different methods from being easily discriminated. The length of a signal generated through a TTS technique depends on the source text used as input. In our case, all the considered sentences are fixed and extracted from the TIMIT Corpus.

Table 3 shows the duration values for each TTS generation system. The average length over the entire dataset is equal to 3.10 s, while considering the single algorithms the durations range from 2.69 to 3.82 s. The standard deviation between the duration of the different methods is not noticeable, being equal to 0.33 s. This means the length of the tracks does not constitute a discriminating element between the different generation algorithms, resulting in a reliable dataset. When we apply DTW, the average length of the tracks rises to 4.25 s. In this case, the average duration is the same for all generation algorithms. This is because the generated tracks have the same duration as the target ones extracted from VidTIMIT, so their length is fixed.

Secondly, we examined the length of silences contained in each track. Although silence is a fundamental component of speech, this is often overlooked in data generation, leading

TABLE 3. Speech metrics for each TTS generator.

Generator	Track dur. [s]		Silences dur. [s]		MOS	
	Clean	DTW	Clean	DTW	Clean	DTW
gTTS	3.82	4.25	0.55	1.29	3.59	3.39
Tacotron	2.69	4.25	0.12	1.48	3.01	3.02
GlowTTS	3.57	4.25	0.78	1.39	3.54	3.51
FastPitch	2.74	4.25	0.41	1.47	3.48	3.35
VITS	2.85	4.25	0.59	1.51	3.69	3.43
FastSpeech2	3.03	4.25	0.05	1.32	3.03	3.00
MixerTTS	3.35	4.25	0.07	1.32	3.04	3.02
MixerTTS-X	3.34	4.25	0.11	1.35	3.02	3.01
SpeedySpeech	3.48	4.25	0.61	1.34	2.84	2.87
Tacotron2	3.21	4.25	0.09	1.34	3.09	3.04
TalkNet	3.02	4.25	0.05	1.33	3.00	2.99
Silero	3.04	4.25	0.09	1.39	2.97	2.97
Average	3.10	4.25	0.44	1.43	3.44	3.29

to biased tracks that are easy to discriminate [67]. This is a common problem, especially when dealing with TTS algorithms, where the prosodic component is less present [43] and the duration of the silences is shorter. Table 3 shows the silence durations of our tracks for both the original and the DTW cases. Here we observe a higher difference between the algorithms, with duration values ranging between 0.05 and 0.78 s. However, when we apply DTW, both the silence duration increase and the differences between the generation methods are reduced, homogenizing the synthesized data.

Next, as we are dealing with speech data, we assessed the naturalness of the generated tracks. We do so to avoid releasing audio signals that sound too unrealistic. The definition of metrics to compute the naturalness of synthetic speech tracks is a challenging task and is still an ongoing research topic [68]. We assume the Mean Opinion Score (MOS) as a metric to indicate the quality of the generated signals and compute it on the synthesized data through Mosnet [69]. MOS is a numerical measure of the human-judged overall quality of an event or experience, ranging from 1 (bad) to 5 (excellent). In our case, we use it to evaluate the naturalness of the generated speech tracks and compare it with the scores obtained on real audio data. The results for each generation algorithm are shown in Table 3. We score an average MOS value greater than 3, which is a value in line with that of other real datasets, such as LJSpeech (MOS=3.05) and VidTIMIT (MOS=2.43). The low value of the latter is likely due to the noisy environment in which it was acquired. These scores mean that, even if we are dealing with synthetic data, we are not neglecting the realism of the speech and its quality. The application of DTW has adverse effects on the MOS of the generated data, lowering the average computed on all the tracks by almost 0.2 points.

Finally, a crucial aspect to address in generated speech data is the number of supported speakers. As the primary goal of the TIMIT-TTS dataset is to perform binary detection of deepfakes, it is essential to provide several speakers. Training a deepfake detector on a few speakers may make the model learn how to discriminate tracks based on the tone of voice they contain instead of the traces left by the TTS generators, as we will highlight in the following experiments. Therefore, providing numerous speakers within the dataset helps avoid this bias and produce more effective models.

TIMIT-TTS implements a total of 37 different speakers in diverse numbers depending on the models used. In addition to the LJSpeech voice, supported by numerous TTS generators, each multi-speaker system implements 8 different voices, 4 male and 4 female, from the VCTK or LibriSpeech datasets. The only exception is gTTS, which only supports 4 English voices. In this case, we have included all 4 within the dataset. The number of speakers implemented for each generation method is shown in Table 1.

B. AUDIO CLASSIFICATION RESULTS

To benchmark the generated data on the deepfake classification task, we consider an audio baseline that performs deepfake detection. We adopt RawNet2 [24], a state-of-the-art end-to-end neural network that operates on raw waveforms. It has been introduced to perform binary classification between real and fake data during the ASVspoof 2019 challenge [13] and included as a baseline in the ASVspoof 2021 challenge [14]. We use the exact implementation proposed in the original paper, so we refer the reader to that for more information.

Here we use the baseline for two different classification tasks. The first one is what it was initially proposed for, namely real vs. fake binary classification. The second is multiclass synthetic speech attribution. Here, given an audio signal generated with any TTS technique, we train the network to discriminate which algorithm has been used to synthesize the audio itself. For this second task, we modified the output layer of the network so that it contains many neurons equal to the number of classes we are addressing. Although this is not the task for which the network was proposed, the problem is very close to that of deepfake classification and the considered model can address it without any issue [70]. Also, the synthetic speech attribution problem has not yet been explored extensively, so there are not many networks proposed explicitly for the task. We illustrate all the experiments performed with RawNet2 in the following sections.

1) AUDIO BINARY CLASSIFICATION: SYNTHETIC SPEECH DETECTION

In this experiment, we want to test how challenging the released dataset is in the deepfake detection task. We perform binary classification considering the audio tracks of the VidTIMIT dataset as real and those of TIMIT-TTS as fake. We use this dataset only in the test phase, following the approach presented in [71], which is helpful for testing the generalization capabilities of a detector. For this experiment, we train RawNet2 on ASVspoof 2019, considering balanced classes and data augmentation on the training data. ASVspoof 2019 is a speech audio dataset created to develop synthetic speech detection techniques. It contains both real and deep-

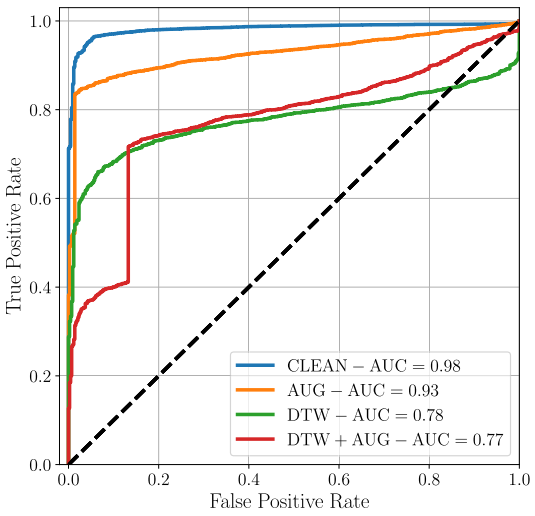


FIGURE 3. Audio binary classification - ROC Curves.

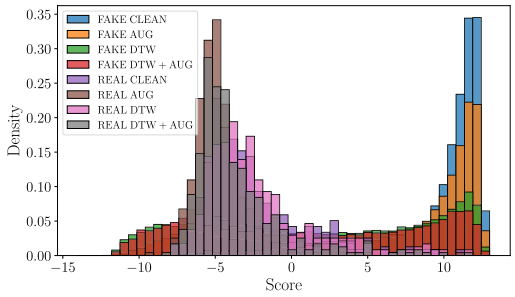


FIGURE 4. Audio binary classification - Scores distribution.

fake speech data generated using TTS, VC, or TTS/VC hybrid methods. We consider the Logical Access (LA) partition of the dataset, which is divided into *train*, *dev* and *eval* subsets. The total amount of speech tracks is around 90 000, split between Real or Fake ones. Fakes are generated with 19 different synthesis algorithms.

We test the detector on the individual partitions of TIMIT-TTS. When we consider augmented partitions, we also process real data from VidTIMIT following the same approach presented in the previous sections to make real and fake data as consistent as possible with each other. The metrics we consider to evaluate the detector on this task are Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values, which are the standard in the evaluation of multimedia forensics detectors.

Figure 3 shows the ROC curves of the results, while Figure 4 shows the distributions of the scores for all the considered classes. In this case, higher scores mean higher confidence in identifying a track as fake. We observe that the detection performance deteriorates as we increment the post-processing operations applied to the speech tracks. In particular, the operation that degrades the accuracy the most is the speech-to-speech alignment, with an AUC value that drops by 0.20 between the clean and the DTW cases. This means that, although these tracks present a lower MOS value in Table 3, deepfake detectors must be explicitly trained on this type of data to discriminate them correctly. Also, this shows that the detection problem of DTW tracks is not solved and our dataset could help in building new detectors that are more robust to in-the-wild conditions. Finally, the augmented tracks are more challenging to detect than the clean ones, with an AUC value that drops by 0.05 between the two cases. As we mentioned above, such post-processing techniques hide some of the traces left by the TTS generators, making it more challenging to identify the artifacts present in the synthesized tracks.

2) AUDIO MULTICLASS CLASSIFICATION: SYNTHETIC SPEECH ATTRIBUTION

In this experiment we want to test the TIMIT-TTS dataset on the synthetic speech attribution task. This consists in identifying, given an input TTS track y, which algorithm has been used to synthesize it. Formally, we have to determine $c_y \in \{c_1, c_2, \ldots, c_i\}$ where i is the number of implemented TTS generation methods. We consider all the 12 generation methods available in the TIMIT-TTS dataset, including all implemented speakers. We split the corpus into train and test sets following a 66% - 33% policy. We ensure a coherent number of tracks for each generation algorithm in both the partitions. We train the RawNet2 model for 100 epochs, using Cross Entropy as loss function and a learning rate equal to 10^{-4}.

Figure 5 shows the results of the analysis through a confusion matrix. We observe different performances for the considered algorithms. In particular, the systems trained to produce speech from multiple speakers are relatively easily identified, while those considering only one speaker are more challenging to distinguish. This is due to the fact that the detection algorithm seems to leverage the different speakers to perform classification rather than focusing on the traces left by each TTS algorithm itself. On the other hand, the methods that implement the same speaker force the model to learn how to discriminate tracks adequately, and the deterioration in performance is due to the difficulty of the required task. To verify this hypothesis, we repeat the same experiment by independently considering the speech tracks generated by models trained on LJSpeech and those trained on other speakers. The results of this analysis are shown in Figure 6 and Figure 7 and confirm the same trend as before, with the initial balanced accuracy value of 0.77 that drops from to 0.67 when we consider LJSpeech models and rises to 0.92 when considering the other models. We believe this aspect is paramount when dealing with both deepfake detection and attribution tasks, as we do not want the results obtained by the algorithm to be biased by the considered speakers. Indeed, having multiple TTS methods trained to reproduce the same voice constitutes a more challenging scenario as it forces the detector to learn the traces left by the generators.

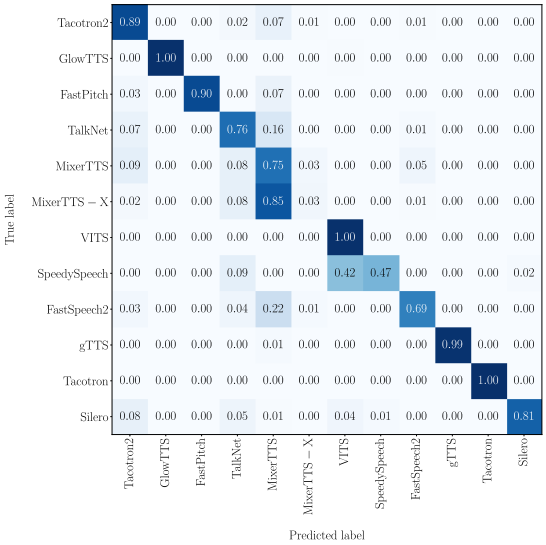


FIGURE 5. Confusion matrix showing the baseline performance on the synthetic speech attribution task, considering all the implemented TTS methods.

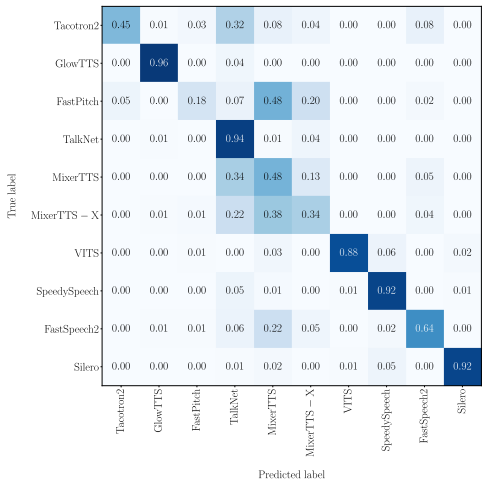


FIGURE 6. Confusion matrix showing the baseline performance on the synthetic speech attribution task, considering only the TTS methods trained on a single speaker.

TIMIT-TTS, providing numerous generation methods trained on LJSpeech, can help develop new attribution algorithms.

C. VIDEO CLASSIFICATION RESULTS

As the final goal of our work is to use the proposed dataset to perform multimodal deepfake detection, we need to compare the final detection performance with those of the single modalities. For this reason, after analyzing the audio component, we operate the detection on the video one. In this case we consider as baseline an EfficientNetB4 [72] network modified following the implementation proposed in [23], which studies the ensembling of different trained CNNs making use of two different concepts such as attention layers and siamese training. Since we use the exact implementation proposed

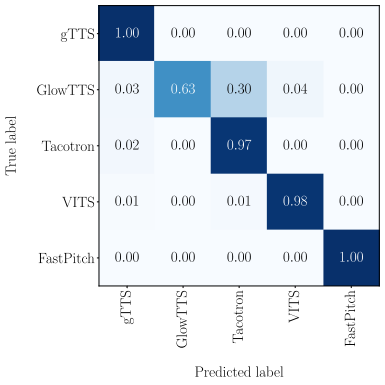


FIGURE 7. Confusion matrix showing the baseline performance on the synthetic speech attribution task, considering only the TTS methods that produce speech with multiple speakers.

in the original paper, we refer the reader to that for more information. As we did in the audio case, we consider a model trained on an external dataset to test its generalization capabilities. We consider the model provided by the authors pre-trained on FaceForensics++ [16] and test it on VidTIMIT and DeepfakeTIMIT datasets, considering them as real and fake data, respectively. The FaceForensics++ dataset contains 5000 videos which were generated using four different deepfake generation methods using a base set of 1000 real YouTube videos.

We build two different versions of the test set, corresponding to two different compression stages of the videos. In particular, we generate a high and low-quality version of the data obtained by considering two different values of quantization parameters (QP=23 and QP=40), where higher QP means lower quality. This has been done for two main reasons. First, this is the same compression approach considered in the FaceForensics++ dataset, so we used it to make our data comparable to those the model has been trained on. Second, we want to study the robustness of the model to compression and analyze how much this influences the detection performance. Robustness is a crucial aspect when dealing with deepfake detectors. The reason is that most of the multimedia material we deal with comes from social media, where they undergo several post-processing and compression steps. Developing a robust algorithm means being able to correctly analyze the multimedia material despite these operations.

Figure 8 shows the results of the detection task in terms of ROC curves and AUC, while Figure 9 shows the distributions of the scores in the considered cases. As in the previous experiment, higher score values mean a higher likelihood that the video is fake. The detection task is accomplished very well when considering "high quality" videos, with an AUC value that is equal to 0.99. This is a significant result, but we will unlikely find data with such high quality in in-the-wild conditions. On the other hand, the performance significantly deteriorates when considering the "low quality" data, with

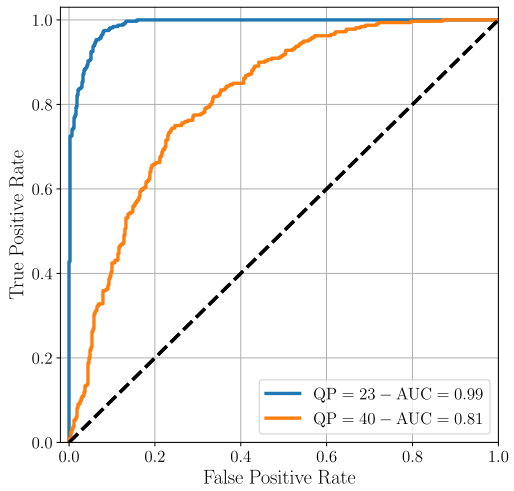


FIGURE 8. Video binary classification - ROC Curves.

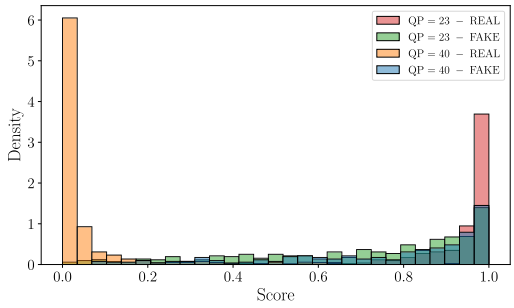


FIGURE 9. Video binary classification - Scores distribution.

an AUC value that drops by almost 0.2. This leaves room for improvement in case of multimodal analysis.

D. MULTIMODAL CLASSIFICATION RESULTS

In this experiment we test the deepfake detection performance of the implemented baselines when considering a multimodal approach. We want to sample if simultaneously examining multiple aspects of a multimedia material can improve the detection capabilities or not. To do so, we combine the VidTIMIT, DeepfakeTIMIT and TIMIT-TTS datasets and associate each audio track with its corresponding video. In this way, we obtain a set of data that is falsified in both audio and video modalities. During this study we analyze the two following scenarios:

- **Scenario 1** - We only consider videos where both their modalities belong to the same class, e.g., audio and video are both real or both fake.
- **Scenario 2** - We consider videos where all the combinations between classes are possible, including data that are counterfeited in only one modality at a time. In this case, we label a video as fake when at least one between its audio and video components is falsified.

We do so to consider two different application cases for a multimodal approach.

In the first scenario, since the classes of the two components are the same, it would also be possible to use a monomodal approach. Nonetheless, we show that analyzing different aspects of the given material can help improve the detection performance.

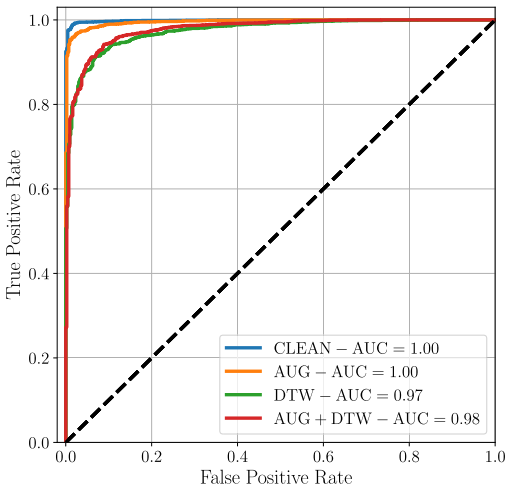


FIGURE 10. Multimodal binary classification - Scenario 1 (RR vs. FF) - QP=23.

The second scenario, on the other hand, is more similar to a real-world case. Here, using a multimodal approach is fundamental since analyzing only one aspect at a time we would lose information and have partial results. For example, we would be unable to detect videos that are counterfeited in just one modality if that is different from the one we are analyzing.

For both scenarios, we consider the baselines introduced above for the single modalities, and we fuse their score in two different ways. In the first case, we compute the average between the two scores, while in the second we consider the higher of the two, which identifies the analyzed element as more likely to be false. Formally, the scores of the two modalities are fused respectively with avg(·) and max(·) functions in the two scenarios.

The results conducted on the first scenario are shown in Figures 10 and 11, divided according to the compression applied to the video modality. The detection performance improves significantly, especially when dealing with post-processed data. In particular, the AUC values improve in all the cases compared to the corresponding experiments on the audio modality. In the second scenario, likewise, the multimodal approach performs considerably better than the monomodal ones. However, comparing Figure 8 and Figure 10, we notice cases where the multimodal performances lightly worsen the monomodal ones. For example, the video-only baseline tested on QP=23 data (AUC=0.99) performs slightly better than the QP=23 + DTW multimodal case (AUC=0.97). This shows that if, on the one hand, a multimodal analysis allows us to obtain reliable and robust results, it is also a technique that must be used consciously. In fact, combining a highly performing monomodal method with another one that is less efficient could worsen the results of the better of the two. In any case, in light of these

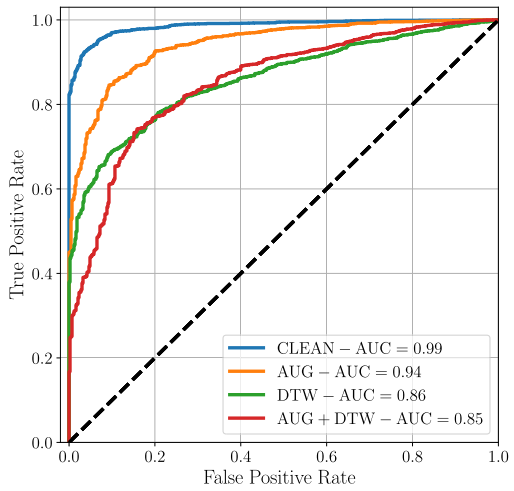


FIGURE 11. Multimodal binary classification - Scenario 1 (RR vs. FF) - QP=40.

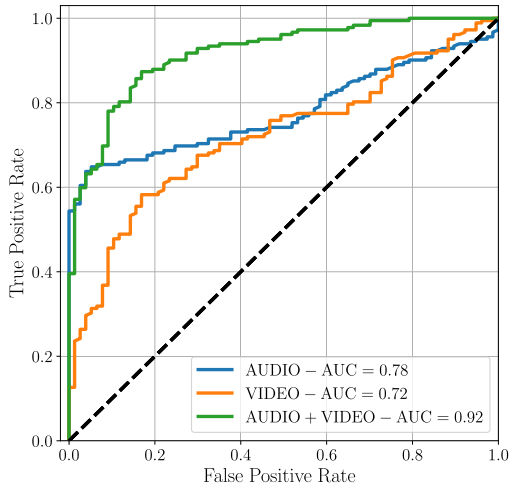


FIGURE 12. Multimodal binary classification - Scenario 2.

experiments, we believe that a multimodal analysis is generally more reliable and robust than a monomodal one.

Figure 12 shows the obtained results in the case we consider clean audio data and a QP=23 for the video, where an AUC improvement of 0.15 is achieved over both the single modalities. This is very interesting since it allows us to detect fake videos that we could not find otherwise. The same experiment was also performed considering post-processed audio data and showed similar results. For this reason, here we only reported the most straightforward case. We highlight that such positive results have been achieved by fusing the scores of the monomodal detectors in a very simple way. We are confident that combining them more smartly could further improve the performance, demonstrating the effectiveness of multimodal deepfake detectors.

VI. CONCLUSION

In this work we presented a pipeline to forge synthetic audio content starting from an input video in order to create a multimodal deepfake dataset. We used this pipeline to generate and release TIMIT-TTS, a synthetic speech dataset that includes audio tracks generated using 12 different TTS systems, among the most advanced in the literature, for a total amount of almost 80 000 tracks. The released dataset has several applications in the forensics field, such as synthetic speech detection and attribution. Moreover, it can be used in conjunction with other well-established deepfake video datasets to perform multimodal studies, bridging an overlooked aspect in the current state-of-the-art. From the presented results, it emerges that multimodal analyzes improve the performance of the detectors, producing more capable and robust systems. At the same time, however, the performances are not entirely satisfactory, so we need more multimodal deepfake datasets, like the one we release, to train and test the developed networks.

To summarize, these are the following contributions of the paper:

- We present a general pipeline for synthesizing speech deepfake content from a given real or fake video, facilitating the creation of counterfeit multimodal material.
- We released TIMIT-TTS, a synthetic speech dataset containing the most cutting-edge methods in the TTS field that can be used as a standalone audio dataset or combined with other sets to perform multimodal research.
- We have shown the effectiveness of performing multimodal analyses, helping develop a new class of detectors that are intrinsically more robust to adversarial and anti-forensic attacks.

This is the dataset's first version, and future developments will be released. There are several aspects worth investigating and synthesis algorithms that have not been included in this set. Regarding TTS systems, we want to examine the effects of using different vocoders on the performance of deepfake detectors and implement a higher number of speakers for all the systems. Also, we want to improve the synchronization method that we are using, which is currently based on the joint use of VAD and DTW. Finally, we also want to include VC algorithms in the study since they have not been involved in this work, even if they could play a key role in this task.

We hope this work will help the development of new multimodal deepfake detectors and provide new data to train and test existing systems to make them able to address in-the-wild conditions.

ACKNOWLEDGMENT

The views and conclusion 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.

REFERENCES

[1] "Pennsylvania woman accused of using deepfake technology to harass cheerleaders," The New York Times, Mar. 14, 2021. [Online]. Available: https://www.nytimes.com/2021/03/14/us/raffaela-spone-victory-vipers-deepfake.html

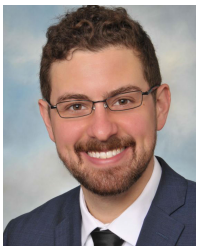
[2] Wired. *A Zelensky Deepfake was Quickly Defeated. The Next One Might Not Be*. Accessed: Mar. 17, 2022. [Online]. Available: https://www.wired.com/story/zelensky-deepfake-facebook-twitter-playbook/
[3] CNN Business. *Deepfakes are Now Trying to Change the Course of War*. Accessed: Mar. 25, 2022. [Online]. Available: https://edition.cnn.com/2022/03/25/tech/deepfakes-disinformation-war/index.html
[4] S. J. Nightingale and H. Farid, "AI-synthesized faces are indistinguishable from real faces and more trustworthy," *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 8, Feb. 2022, Art. no. e2120481119.
[5] NPR. *That Smiling LinkedIn Profile Face Might be a Computer-Generated Fake*. Accessed: Mar. 27, 2022. [Online]. Available: https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles?t=1649099040349
[6] L. Verdoliva, "Media forensics and deepfakes: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 910–932, Aug. 2020.
[7] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake detection: A systematic literature review," *IEEE Access*, vol. 10, pp. 25494–25513, 2022.
[8] H. R. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *IEEE Access*, vol. 7, pp. 41596–41606, 2019.
[9] L. Guarnera, O. Giudice, and S. Battiato, "Deepfake detection by analyzing convolutional traces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 2841–2850.
[10] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes detection using human eye blinking pattern," *IEEE Access*, vol. 8, pp. 83144–83154, 2020.
[11] S. Agarwal, H. Farid, T. El-Gaaly, and S. Lim, "Detecting deep-fake videos from appearance and behavior," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.
[12] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (DFDC) dataset," 2020, *arXiv:2006.07397*.
[13] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019.
[14] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," in *Proc. Ed. Autom. Speaker Verification Spoofing Countermeasures Challenge*, Sep. 2021.
[15] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, C. Fan, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, Z. Wen, and H. Li, "ADD 2022: The first audio deep synthesis detection challenge," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 9216–9220.
[16] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.
[17] P. Korshunov and S. Marcel, "DeepFakes: A new threat to face recognition? Assessment and detection," 2018, *arXiv:1812.08685*.
[18] J. Frank and L. Schönherr, "WaveFake: A data set to facilitate audio deepfake detection," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2021.
[19] B. Hosler, D. Salvi, A. Murray, F. Antonacci, P. Bestagini, S. Tubaro, and M. C. Stamm, "Do deepfakes feel emotions? A semantic approach to detecting deepfakes via emotional inconsistencies," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1013–1022.
[20] M. Lomnitz, Z. Hampel-Arias, V. Sandesara, and S. Hu, "Multimodal approach for deepfake detection," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2020, pp. 1–9.
[21] H. Khalid, M. Kim, S. Tariq, and S. S. Woo, "Evaluation of an audio-video multimodal deepfake dataset using unimodal and multimodal detectors," in *Proc. 1st Workshop Synth. Multimedia-Audiovisual Deepfake Gener. Detection*, Oct. 2021, pp. 7–15.
[22] C. Sanderson and B. C. Lovell, "Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference," in *Advances in Biometrics* (Lecture Notes in Computer Science), vol. 5558. 2009, pp. 199–208.
[23] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of CNNs," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 5012–5019.
[24] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6369–6373.
[25] H. Khalid, S. Tariq, M. Kim, and S. Simon Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," in *Proc. Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round)*, 2021.
[26] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2018, pp. 1086–1090.
[27] D. H. Klatt, "Review of text-to-speech conversion for English," *J. Acoust. Soc. Amer.*, vol. 82, no. 3, pp. 737–793, 1987.
[28] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. SSW*, vol. 125, 2016, p. 2.
[29] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "NeMo: A toolkit for building AI applications using neural modules," 2019, *arXiv:1909.09577*.
[30] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Aug. 2017, pp. 4006–4010.
[31] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7654–7658.
[32] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech Language Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.
[33] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 5, pp. 954–964, Jul. 2010.
[34] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 266–273.
[35] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-VC2: Improved cyclegan-based non-parallel voice conversion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6820–6824.
[36] S. Lyu, "Deepfake detection: Current challenges and next steps," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.
[37] Z.-F. Wang, G. Wei, and Q.-H. He, "Channel pattern noise based playback attack detection algorithm for speaker recognition," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2011, pp. 1708–1713.
[38] H. Malik, "Securing voice-driven interfaces against fake (cloned) audio attacks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 512–517.
[39] C. Borrelli, P. Bestagini, F. Antonacci, A. Sarti, and S. Tubaro, "Synthetic speech detection through short-term and long-term prediction traces," *EURASIP J. Inf. Secur.*, vol. 2021, no. 1, pp. 1–14, Dec. 2021.
[40] A. Hamza, A. R. R. Javed, F. Iqbal, N. Kryvinska, A. S. Almadhor, Z. Jalil, and R. Borghol, "Deepfake audio detection via MFCC features using machine learning," *IEEE Access*, vol. 10, pp. 134018–134028, 2022.
[41] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2015.
[42] E. Conti, D. Salvi, C. Borrelli, B. Hosler, P. Bestagini, F. Antonacci, A. Sarti, M. C. Stamm, and S. Tubaro, "Deepfake speech detection through emotion recognition: A semantic approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8962–8966.
[43] L. Attorresi, D. Salvi, C. Borrelli, P. Bestagini, and S. Tubaro, "Combining automatic speaker verification and prosody analysis for synthetic speech detection," in *Proc. Int. Conf. Pattern Recognit.*, 2022.
[44] C. Jemine. (2022). *Real-Time-Voice-Cloning*. [Online]. Available: https://github.com/CorentinJ/Real-Time-Voice-Cloning
[45] 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, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4779–4783.

[46] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8067–8077.
[47] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," 2020, *arXiv:2006.04558*.
[48] A. Lancucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6588–6592.
[49] S. Beliaev, Y. Rebryk, and B. Ginsburg, "TalkNet: Fully-convolutional non-autoregressive speech synthesis model," 2020, *arXiv:2005.05514*.
[50] O. Tatanov, S. Beliaev, and B. Ginsburg, "Mixer-TTS: Non-autoregressive, fast and compact text-to-speech model conditioned on language model embeddings," 2021, *arXiv:2110.03584*.
[51] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5530–5540.
[52] J. Vainer and O. Dušek, "SpeedySpeech: Efficient neural speech synthesis," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Oct. 2020, pp. 3575–3579.
[53] P. N. Durette. (2022). *gTTS*. [Online]. Available: https://github.com/pndurette/gTTS
[54] Silero Team. (2021). *Silero Models: Pre-Trained Enterprise-Grade STT/TTS Models and Benchmarks*. [Online]. Available: https://github.com/snakers4/silero-models
[55] K. Kumar, R. Kumar, T. D. Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. D. Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[56] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2410–2419.
[57] K. Ito and L. Johnson. (2017). *The LJ Speech Dataset*. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/
[58] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.
[59] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," Centre Speech Technol. Res., Univ. Edinburgh, Edinburgh, U.K., 2019, doi: 10.7488/ds/2645.
[60] K. Chugh, P. Gupta, A. Dhall, and R. Subramanian, "Not made for each other-audio-visual dissonance-based deepfake detection and localization," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 439–447.
[61] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync toolbox: A Python package for efficient, robust, and accurate music synchronization," *J. Open Source Softw.*, vol. 6, no. 64, p. 3434, Aug. 2021.
[62] I. Jordal. (2022). *Audiomentations*. [Online]. Available: https://github.com/iver56/audiomentations
[63] A. Cohen, I. Rimon, E. Aflalo, and H. H. Permuter, "A study on data augmentation in voice anti-spoofing," *Speech Commun.*, vol. 141, pp. 56–67, Jun. 2022.
[64] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Proc. Int. Conf. Biometrics*, 2009, pp. 199–208.
[65] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," Feb. 1993, p. 27403. [Online]. Available: https://ui.adsabs.harvard.edu/abs/1993STIN...9327403G
[66] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022, *arXiv:2212.04356*.
[67] N. M. Müller, F. Dieckmann, P. Czempin, R. Canals, K. Böttinger, and J. Williams, "Speech is silver, silence is golden: What do ASVspoof-trained models really learn?" in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2021.
[68] G. Mittag and S. Möller, "Deep learning based assessment of synthetic speech naturalness," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Oct. 2020, pp. 1748–1752.
[69] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep learning-based objective assessment for voice conversion," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 1541–1545.
[70] D. Salvi, P. Bestagini, and S. Tubaro, "Exploring the synthetic speech attribution problem through data-driven detectors," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2022, pp. 1–6.
[71] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" 2022, *arXiv:2203.16263*.
[72] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

DAVIDE SALVI (Student Member, IEEE) was born in Bergamo, Italy, in 1995. He received the M.Sc. degree in music and acoustic engineering from Politecnico di Milano, Italy, in 2020, where he is currently pursuing the Ph.D. degree with the Image and Sound Processing Laboratory (ISPL), Department of Electronics, Information and Bioengineering (DEIB). His research interest includes signal processing for multimedia forensics applications.

BRIAN HOSLER (Student Member, IEEE) received the B.S. degree in electrical engineering from Drexel University, Philadelphia, PA, USA, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. From 2014 to 2016, he was involved in the fields of 2D nanomaterials with the Drexel Nanomaterials Institute. From 2017 to 2018, he was an Engineer with BMW Manufacturing, Greenville, SC, USA. He is currently a Research Assistant with the Multimedia and Information Security Laboratory (MISL), Drexel University, where he is conducting research on video and multimedia forensics. His current research interests include signal processing and machine learning.

PAOLO BESTAGINI (Member, IEEE) was born in Novara, Italy, in 1986. He received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in information technology from Politecnico di Milano, Italy, in 2010 and 2014, respectively. He is currently an Assistant Professor with the Image and Sound Processing Laboratory (ISPL), Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano. He has been a Scientific Investigator for the European projects SCENIC and REWIND coordinated by Politecnico di Milano and a Co-Principal Investigator for the DARPA-funded MediFor Project. He is a Co-Principal Investigator for the DARPA-funded SemaFor Project. His research interests include multimedia forensics and acoustic signal processing for microphone arrays. He is elected as a member of the IEEE Information Forensics and Security Technical Committee for the second time. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) and the *Journal of Visual Communication and Image Representation* (JVCI) (Elsevier). He is Co-Organizer of the IEEE Signal Processing Cup, in 2018 and 2022.



MATTHEW C. STAMM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2004, 2011, and 2012, respectively. Prior to beginning the bachelor's degree, he was an Engineer with the Applied Physics Laboratory, Johns Hopkins University. Since 2013, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA. He leads the Multimedia and Information Security Laboratory (MISL), where he and his team conduct research on signal processing, machine learning, and information security with a focus on multimedia forensics and anti-forensics. He was a recipient of the 2016 NSF CAREER Award and the 2017 Drexel University College of Engineering's Outstanding Early-Career Research Achievement Award. For his Ph.D. dissertation research, he was named the winner of the Dean's Doctoral Research Award from the A. James Clark School of Engineering. While at the University of Maryland at College Park, he was also a recipient of the Ann G. Wylie Dissertation Fellowship and the Future Faculty Fellowship. He was the General Chair of the 2017 ACM Workshop on Information Hiding and Multimedia Security. He is the Lead Organizer of the 2018 IEEE Signal Processing Cup Competition. He currently serves as a member of the IEEE SPS Technical Committee on Information Forensics and Security and a member of the editorial board of the IEEE SigPort.

STEFANO TUBARO (Senior Member, IEEE) was born in Novara, Italy, in 1957. He received the M.Sc. degree in electronic engineering from Politecnico di Milano, Milan, Italy, in 1982. He then joined the Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, first as a Researcher of the National Research Council, and then as an Associate Professor, in 1991. Since 2004, he has been appointed as a Full Professor in telecommunication with Politecnico di Milano. He coordinates the research activities of the Image and Sound Processing Laboratory (ISPL), Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano. He had the role of a Project Coordinator of the European Project ORIGAMI: A new paradigm for high-quality mixing of real and virtual and of the research project ICT-FET-OPEN REWIND: REVerse engineering of audio-VIsual coNtent Data. This last project was aimed at synergistically combining principles of signal processing, machine learning, and information theory to answer relevant questions on the past history of such objects. He has authored more than 180 scientific publications on international journals and congresses and has coauthored more than 15 patents. In the past few years, he has focused his interest on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the processing history of multimedia objects. His current research interest includes advanced algorithms for video and sound processing. He is a member the IEEE Multimedia Signal Processing Technical Committee and of the IEEE SPS Image Video and Multidimensional Signal Technical Committee. He was on the Organization Committee of a number of international conferences, including the IEEE International Workshop on Multimedia Signal Processing (MMSP) in 2004 and 2013, the IEEE International Conference on Image Processing (ICIP) in 2005, the IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS) in 2005 and 2009, the IEEE International Conference on Distributed Smart Cameras (ICDSC) in 2009, the IEEE MMSP in 2013, and the IEEE International Conference on Multimedia and Expo (ICME) in 2015. From 2012 to 2015, he was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP). He is currently an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS).

Open Access funding provided by 'Politecnico di Milano' within the CRUI CARE Agreement