Is synthetic voice detection research going into the right direction?

Stefano Borzì [1], Oliver Giudice [1,2], Filippo Stanco [1], Dario Allegra [1]

[1] Department of Mathematics and Computer Science, University of Catania, Italy

[2] Banca d'Italia, Applied Research Team, IT dept., Rome, Italy

stefano.borzi@phd.unict.it, giudice@dmi.unict.it

filippo.stanco@unict.it, dario.allegra@unict.it

Abstract

Machine Learning, and in general Artificial Intelligence approaches, brought a great advance in each and every field of Computer Science increasing accuracy levels of predictors in any known problem. Indeed, this evolution enabled the construction of effective frameworks and solutions able to be used in investigative and forensics scenarios for detection of fakes and, in general, manipulations in multimedia contents. On the other hand, can we trust these systems? Is research activity going in the right direction? Are we just taking the low-hanging fruit without taking into account many real-case-in-the-wild situations? The purpose of this paper is to raise an alert to the research community in the specific context of synthetic voice detection, where data available for training is not big enough to give sufficient trust in the techniques available in the literature. To this aim, an exploratory investigation of the most common voice spoofing dataset was carried out and it was surprisingly easy to build simple classifiers without any Deep Learning techniques. Simple considerations on bitrate were sufficient to achieve an effective detection performance.

1. Introduction

During the pandemic, there was an increase in the use of technology for various purposes, such as opening a bank account via webcam and using voice recognition as access authentication. Together, the deepfakes have gained relevant widespread on the web and today, thanks to several tools, it is easy for everyone to clone voice or create an entire video of famous people saying anything, by creating fake news. In the report "Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case" [23], criminals used AI software to impersonate a CEO's voice and successfully stole more than $243,000 by speaking on the phone. In order to improve the research about spoofing countermeasure, the ASV community has released standard spoofing datasets [29], [15], [27]. The ASVspoof 2019 challenge [27] combines both Logical Access (LA) and Physical Access (PA) attacks using the latest state-of-the-art Voice Conversion (VC) and Text-to-Speech (TTS) methods, in particular PA includes replay attacks and the LA includes the VC and TTS attacks. In this paper, we analyze ASVspoof 2019 LA dataset, its characteristics and related drawbacks.

Multimedia forensics has been exploiting Machine Learning solutions to build alteration and/or fake detectors for years specifically in the image/video context [4, 25]. Indeed, the evolution of Machine Learning (ML) techniques has increasingly led researchers to solve any problem with these kinds of techniques. The approach is always the same: to encode some feature and to build an ensemble that with enough data always proves to be able to find an effective solution. With the introduction of Deep Learning techniques this phenomenon has become more and more sophisticated. But is research moving into the right direction? It is true that when we talk about images, we talk about giant large-scale datasets. Many solutions for Deepfake Detection, of faces, for example, were already presented with impressive results [10, 13]. But when it comes to audio samples, the datasets are not that large, often there is not enough variability in terms of language, gender and number of speakers or characteristics and types of microphones. In a context in which the datasets are much smaller than in other contexts and in which their variability is not guaranteed, is it correct to apply ML intensively? Do ML solutions just focus on trivial features? In this paper, in order to answer this question, audio files of the ASVspoof dataset will be employed, which is commonly used by state-of-the-art papers as a benchmark. The dataset will be analyzed on the basis of extremely simple features. To this aim, an exploratory investigation was carried out and it was surprisingly easy to build simple classifiers without any Deep Learning techniques. Simple considerations on file bitrates were sufficient to achieve a high level of detection performance.

The remainder of this paper is organized as follows: Section 2 presents the state of the art solutions dividing them into those based on hand-crafted features and those based

on Deep Learning approaches; Section 3 presents the employed dataset and Section 4 will present the feature that will be investigated on it. Section 5 will present results obtained by means of simple features for the synthetic audio detection problem thus proposing an extremely simple but effective pipeline. Obtained results will be discussed in Section 6. Section 7 concludes the paper.

Given the high number of features considered throughout this paper, we report all the experiment results in our web viewer (https://unict-fake-audio.github.io/ASVspoof2019-feature-webview/dataset-webview).

2. Related works

Most of the publications related to spoofing attacks detection that uses the dataset ASVspoof 2019 are focused on the Logical Access (LA) partition because instead of the other part Physical Access (PA) which is based only on replay attacks, the LA partition seems to be an imminent threat for unknown nature of attacks [6].

Several researchers based their approaches of spoofing detection on specific and very engineered pipelines, a list of recent study cases that can represent the state of the art are present in this survey [18]. A good example of well designed pipeline is in the paper of Wang et al. [26], where has been used the DeepSonar framework for the feature extraction monitoring the behavior of a speaker recognition model (thin-ResNet), selecting the feature from a significant layer of the chosen deep neural network model. In the Wang et al. work, it is considered the robustness of applying their approach through multiple datasets, indeed they have created and edited different datasets to test the robustness considering that the voice manipulations like voices resampling, adding noises are really common in real applications, thus, for detectors is important to avoid and be robust on possible manipulation attacks. Similarly. in this paper we will show that we have applied different changes into the dataset applying a loud normalization [22] and changing specifically the bitrate values to verify the robustness of our approach. Another recent work with a very engineered pipeline is the Zhang et al. [32] one which consists of data augmentation, acoustic feature extraction, like the spectrogram features, applying a transformer encoder and ResNet for further feature extractions (TE-ResNet), average pooling and data drop out. Moreover, in the data augmentation step, five speech data augmentation techniques have been used [7]. The Zhang et al. approach reached high performance (EER 5.89% using ASVspoof2019 LA) but it requires extensive training data. As a result of the survey [18], most of the approaches used to reach high performances but at the same time, they are really computationally expensive and complex, in this paper, we propose a simple and not computationally expensive approach with modest results showing that there can be a different approach compared the one used so far. Furthermore, we partially agree that specific spoofing techniques included in ASVspoof2019 LA like A14, A17, A18 are not immediate to identify and using simple approaches based on Softmax are not efficient and close-to-zero results [31] but at the same time we propose some other simple approaches that reach with low hardware resources the accuracy of 88.6% - 99.6%.

3. Dataset ASVspoof2019 Logical Access (LA)

The Automatic Speaker Verification (ASV) spoof 2019 dataset [27] is partitioned in two-part, Logical Access (LA) and Physical Access (PA), both parts are based upon the Voice Cloning Toolkit (VCTK) corpus [24].

The ASVspoof 2019 dataset was created using utterances from 107 speakers (46 male, 61 female), all the utterances represent the genuine voice and spoofing attacks based on replay, speech synthesis, and voice conversion attacks. In this paper, we will focus on ASVspoof 2019 Logical Access dataset where during the analysis have been found different significant features that help to detect the spoofing utterances samples.

All the utterances present in ASVspoof 2019 LA dataset have been created using text-to-speech synthesis (TTS) and voice conversion (VC) attacks, in particular, some of the algorithms involved are illustrated in table 1 with the quantity of detectable utterances by the bitrate feature and the total utterances of the related spoofing techniques.

4. Feature extraction and analysis

Initially, we started to analyze the dataset extracting the main spectrum features and all generic technical features related to audio using the framework Simplified Python Audio-Features Extraction (SPAFE) and *pydub*. The feature extracted are the following:

- *mean of MFCC, IMFCC, BFCC, LFCC, LPC, LPCC, MSRCC, NGCC, PSRCC, PLP, RPLP, GFCC*
- *spectrum*
- *mean_frequency*: mean frequency (in kHz)
- *peak_frequency*: peak frequency (frequency with highest energy)
- *frequencies_std*: frequency standard deviation
- *amplitudes_cum_sum*: cumulative sum of the amplitude
- *mode_frequency*
- *median_frequency*: median frequency (in kHz)

ID	Technique	Detectable utterances (by bitrate)	Total utterances
A07	TTS vocoder+GAN	4803	4914
A10	TTS neural waveform	4810	4914
A11	TTS griffin lim	4621	4914
A13	TTS_VC waveform concatenation+filtering	4883	4914
A14	TTS_VC vocoder	4883	4914
A15	TTS_VC neural waveform	3689	4914

Table 1. ASVspoof 2019 Logical Access spoofing techniques including all the techniques where the bit rate is significantly discriminative because it overcomes the threshold of 160 000 (bit rate value) for the quantities shown in the table with all speakers present in the dataset.

- *frequencies_q25*: first quantile (in kHz)
- *frequencies_q75*: third quantile (in kHz)
- *IQR*: interquantile range (in kHz)
- *freqs_skewness*
- *freqs_kurtosis*
- *spectral values*: entropy, flatness, centroid, spread, rolloff, mean, RMS, standard deviation, variance
- *energy*: magnitude spectrum
- *zcr*: zero crossing rate (mean)
- *fundamental and dominant frequency*: min, mean and maximum fundamental frequency and dominant frequency measured across acoustic signal
- *dfrange*: range of dominant frequency measured across acoustic signal
- *modindex*: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
- *bit_rate*: extracted using *pydub mediainfo*

Through the analysis of these features, it has been found that mode frequency, peak frequency, energy and bit rate are surprisingly discriminative to distinguish the synthetic audio from the bonafide ones.

4.1. Analysing the bitrate

There are only 516 bonafide utterances over 12483 with a bit rate value greater than 160000, so, all the utterances with a bit rate greater than this value are, with a high probability, synthetic utterances. In the table 1 there are the utterances which overcome the threshold of 160000 per spoofing technique, it is possible to see that not only the spoofing techniques in the table 1 are easily detectable by the bit rate but also other techniques have several utterances after the bit rate threshold (Figure 1).

4.2. Analysing mode frequency

All the bonafide utterances have a *mode_frequency* lower than 250, in particular the values of the bonafide are 63, 94, 125, 156 with the exception of 258 over 12483 utterances. The utterances based on the spoofing technique A02 - TTS vocoder have a *mode_frequency* value of 250, 436 and 467, so, in order to detect such utterances, one can set a threshold and identify all of them as it is possible to see in Figure 2.

The utterances based on the spoofing technique A05 - VC vocoder have a *mode_frequency* value greater than 156, except for 840 of them (Figure 3). By considering that the bonafide values with a mode frequency greater than 156 are 262 over 12483, it is possible to detect the A05 synthetic utterances using the threshold 156 for *mode_frequency*; moreover, a similar behavior is observed in the *peak_frequency* feature.

4.3. Analysing energy spectrum

The bonafide utterances have an *energy* value greater than 10^{-6}, except for 717 over 12483 utterances.
On the oter hand, the utterances based on A03 - TTS vocoder spoofing technique shows an *energy* value between 10^{-7} and 10^{-5} except for 279 over 7516 utterances.

The utterances based on A09 - TTS vocoder spoofing technique have an *energy* value lower than 10^{-6}, except for 1205 over 4914 utterances, so, it is possible to detect more than the 75% of the synthetic utterances (see Figure 4).

The utterances based on A13 - TTS VC waveform concatenation + waveform filtering spoofing technique have an *energy* value greater than 0.0002 with the exception of 89 over 4914 utterances, indeed, it is possible to detect the synthetic utterances.

5. Experiments and results

In order to verify how much the features affect the dataset we trained several classifiers from the *scikit-learn* python library [20] considering 11 speakers (7 for A01-A06, 5 for A07-A19) and checking the accuracy of them

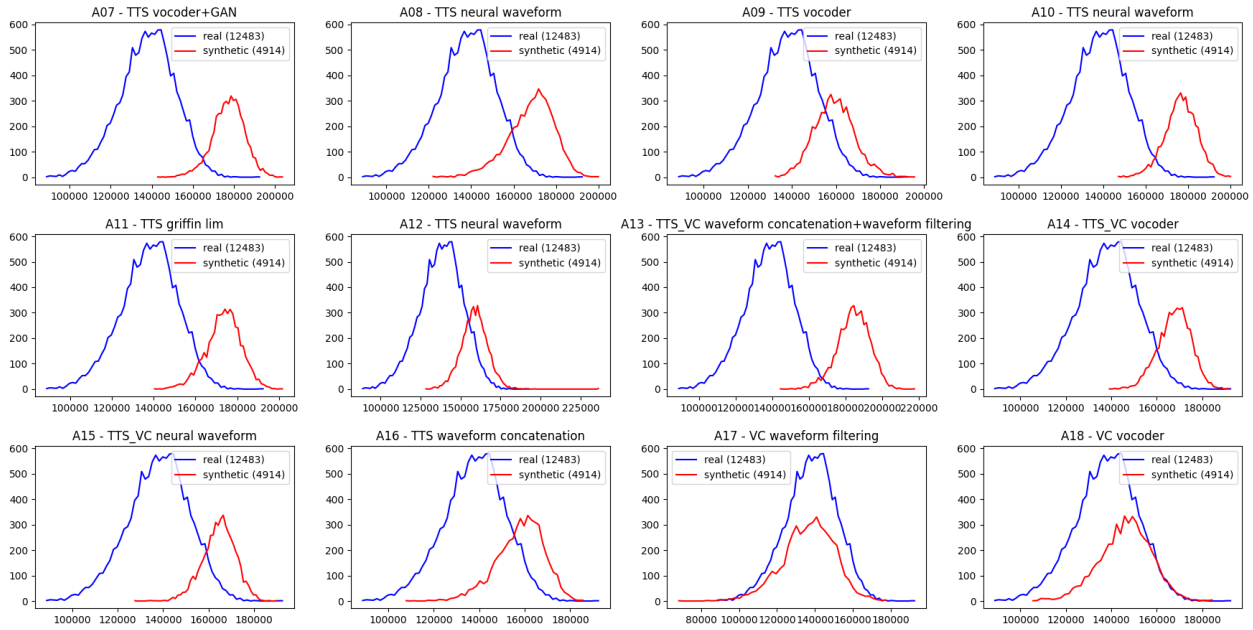

Figure 1. All speakers bonafide utterances (blue) and synthetic (red) per spoofing technique (A07-A18). The x-axis value is the bit rate, the y-axis is the number of utterances per the related x-axis bit rate.

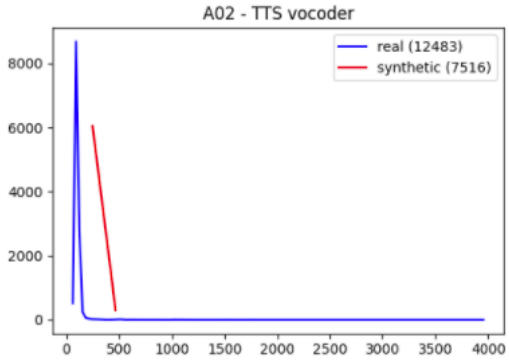

Figure 2. All speakers bonafide utterances (blue) and synthetic (red) ones of the spoofing technique A02. The x-axis value is the mode frequency, the y-axis is the number of utterances per the related x-axis mode frequency value.

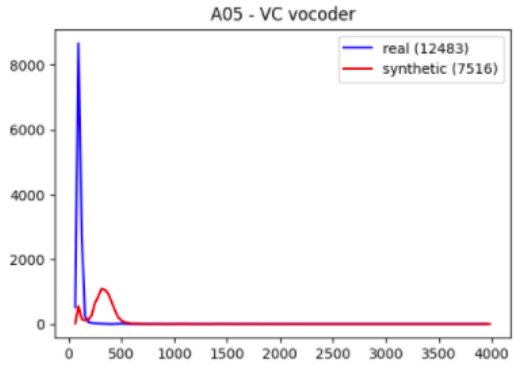

Figure 3. All speakers bonafide utterances (blue) and synthetic (red) ones of the spoofing technique A05. The x-axis value is the mode frequency, the y-axis is the number of utterances per the related x-axis mode frequency value.

with and without the bitrate feature. Additionally, we decided to generate two version of the dataset setting a different bitrate using the *pydub* library and applying a loud normalization using *pyloudnorm* [22].

The speakers selected related to the spoofing system for the experiments are: LA_0069, LA_0070, LA_0071, LA_0072, LA_0073, LA_0074, LA_0075 for A01-A06 and LA_0012, LA_0013, LA_0047, LA_0023, LA_0038 for A07-A19.

From the Figure 5 it is possible to see that after the bit rate changes the relation between bonafide and synthetic bit rate values remained. This is confirmed also when employing loud normalization (see supplementary material links for complete results).

We employed a wide set of classifiers from those available in the *scikit-learn* python library. All the classifiers have been trained on regular train-test split and it has been calculated the accuracy for each one, in particular, it has been calculated the accuracy in different scenarios:

- 1 - standard dataset without any normalization or modification without the feature bit rate

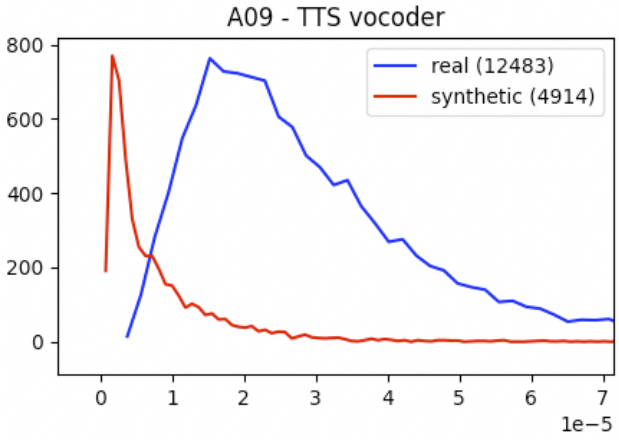

Figure 4. All speakers utterances bonafide (blue) and synthetic (red) ones of the spoofing technique A09. The x-axis value is the energy, the y-axis is the number of utterances per the related x-axis energy value.

- 2 - dataset with the bit rate changed using as frame rate 16000 and bit rate 90000, depending on the audio the output bit rate is not 90000 but it is less than the initial one
- 3 - dataset with the loud normalization applied
- N* - all the training and classification results using in addition the bit rate feature

As a result, by employing only the bit rate most of the time the accuracy is surprisingly high (Figure 6, 7 and 8).

Among all experiments, it is possible to see from the histogram graphs that the best classifier is AdaBoostClassifier (ADC) (Accuracy = 89.7%, AUC = 63.2%, EER = 5%). As a result, an experiment has been made with all the datasets with the ADC classifier with the results in Table 3 compared to the current state of the art. All the experiments shown in the histograms graphs have been made using several models (Table 2) provided by *scikit-learn* and standard parameters mentioned in the library documentation. All the experiments consist of a training and evaluation phase, using respectively the training set and the evaluation set provided by ASVspoof 2019 LA where the models have been trained and used to predict the results. These processes have been repeated multiple times (100) and we took the average of the evaluation metrics results. The accuracy and AUC values were calculated employing the python library sklearn.metrics. The EER was calculated using the mean of False Acceptance Rate (FAR) and False Rejection Rate (FRR) defined as follows:

$$FAR = \frac{FP}{(TP + FP + TN + FN)} \quad (1)$$

$$FRR = \frac{FN}{(TP + FP + TN + FN)} \quad (2)$$

$$EER = \frac{FAR + FRR}{2} \quad (3)$$

6. Discussion

The overall experiments and results show that the ASVspoof 2019 Logical Access dataset is not so robust to simple classifiers with a feature extraction without any specific strategy but based on the standard tools available on the web. What is really unexpected is the significance of different features like the bitrate which is a well-known feature of the audio samples and that can often determine if an utterance is synthetic or bonafide. Are the involved spoofing techniques not enough various and too much detectable? Or is it hard to hide some specific feature that characterizes the bonafide and synthetic utterances? In this paper we analyzed and showed different features that are extremely linked to the label, setting a threshold to allow to easily detection several synthetics utterances, this can be considered as a drawback of this dataset or of the spoofing techniques. Exhaustive experiments results can be found at the following URL: https://unict-fake-audio.github.io/ASVspoof2019-feature-webview/dataset-webview/.

7. Conclusion

In order to find specific characteristics and features related to synthetic utterances, it is not necessary to use deep learning techniques with a complex feature extraction [26]. Exploring a dataset can give the opportunity to find the right feature to detect the synthetic utterances without using deep learning strategies which require a huge pipeline and workflow computationally complex and expensive for feature extraction and classification.

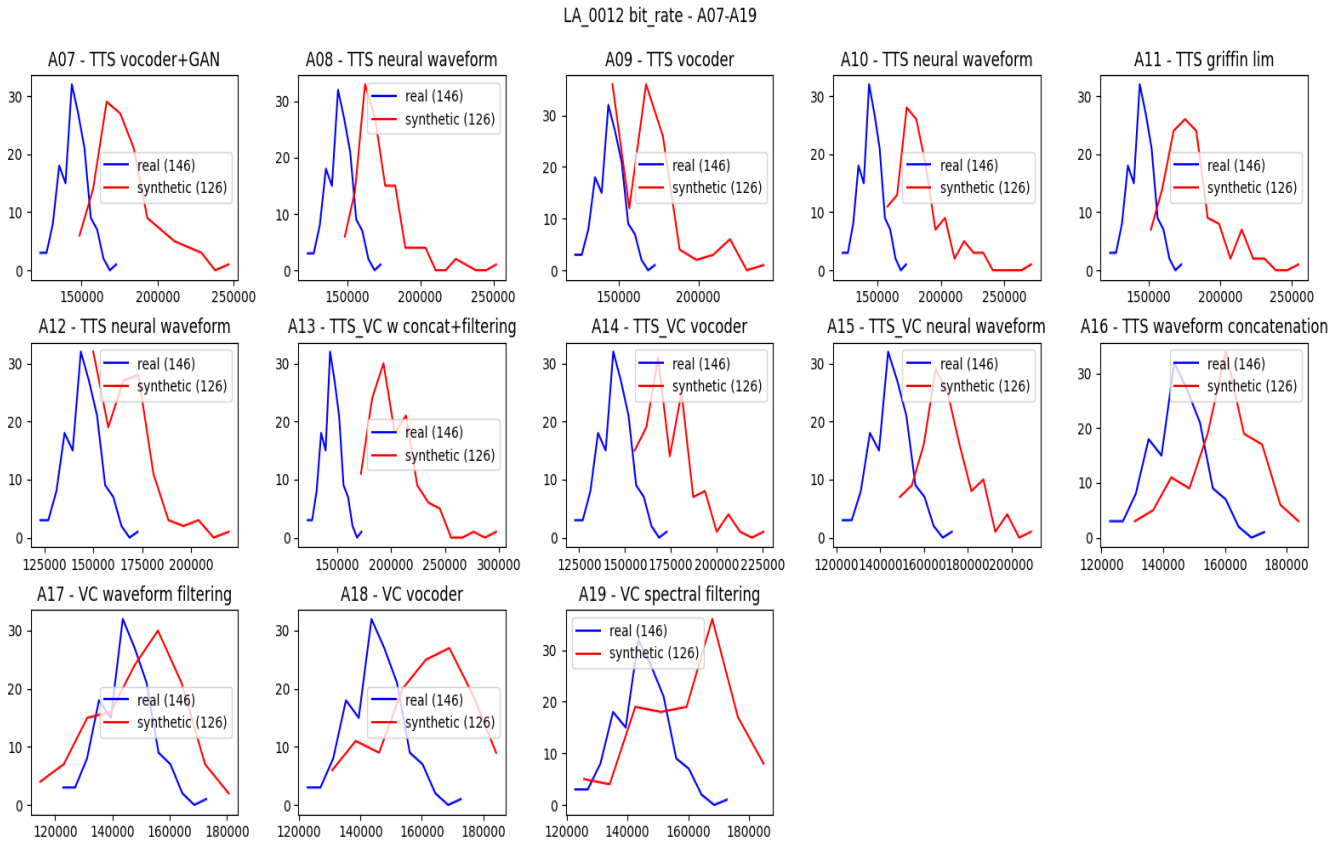

Figure 5. Bonafide (blue) and synthetic (red) bit rate utterances of the speaker LA_0012 of spoofing techniques A07-A19 with the normalized bit rate dataset.

Acronym	**Full name**	**Parameters**
CART	DecisionTreeClassifier	
SVM	Nu-Support Vector Classification	
LR	LogisticRegression	
KNN	KNeighborsClassifier	11
GMM	GaussianMixture	components=2, state=0
LDA	LinearDiscriminantAnalysis	
SVC1	C-Support Vector Classification	gamma=2, C=1
SVC2	C-Support Vector Classification	kernel=”linear”, C=0.025
GPC	GaussianProcessClassifier	1.0 * RBF(1.0)
RFC	RandomForestClassifier	depth=5, est.=10, feat.=1
MLP	MLPClassifier	alpha=1, max_iter=1000
ADC	AdaBoostClassifier	
GNB	GaussianNB	
QDA	QuadraticDiscriminantAnalysis	
NB	BernoulliNB	

Table 2. All the classifiers imported from scikit-learn during the experiments

.

Author	Classification Technique	Best Evaluation Performances
Li et al. [16]	Res2Net	EER=2.502
Yi et al. [30]	GMM/LCNN	EER=19.22 (GMM) EER=6.99 (LCNN)
Das et al. [7]	LCNN	EER=3.13
Aljasem et al. [2]	Asymmetric bagging	EER=5.22
Ma et al. [17]	CNN	EER=9.25
AlBadawy et al. [1]	logistic regression classifier	AUC=0.99
Singh et al. [21]	Quadratic SVM	Acc=96.1%
Gao et al. [8]	ResNet	EER=4.03
Aravind et al. [3]	ResNet34	EER=5.87
Monteiro et al. [19]	LCNN / ResNet	EER=6.38
Chen et al. [5]	ResNet	EER=1.81
Huang et al. [12]	DenseNet-BiLTSSTM	EER=0.53
Wu et al. [28]	LCNN	EER=4.07
Zhang et al. [32]	TEResNet	EER=5.89 ERR=3.99
Zhang et al. [31]	ResNet-18+OC-softmax	EER=2.19
Gomez-Alanis et al. [9]	LCG-RNN	EER=6.28
Hua et al. [11]	Res-TSSDNet	EER=1.64
Jiang et al. [14]	CNN	EER=5.31
Wang et al. [26]	DNN	EER=0.021
Our	**AdaBoost**	ACC = 89.7%, AUC = 63.2%, EER = 5%

Table 3. Deepfake survey comparison [18] with, in addition, our experiment results created by using all features described in this paper.

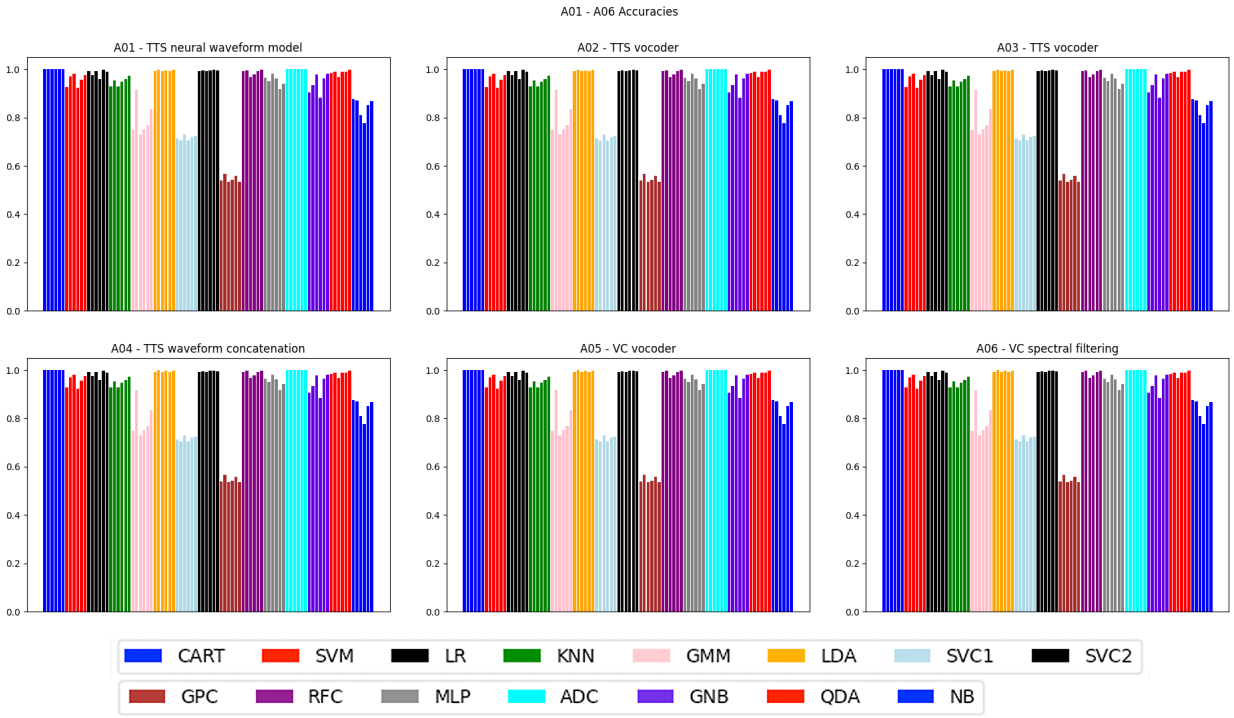

Figure 6. All accuracies per classifier and spoofing technique A01-A06; each bin per color shows the 1, 1*, 2, 2*, 3 and 3* accuracy values

.

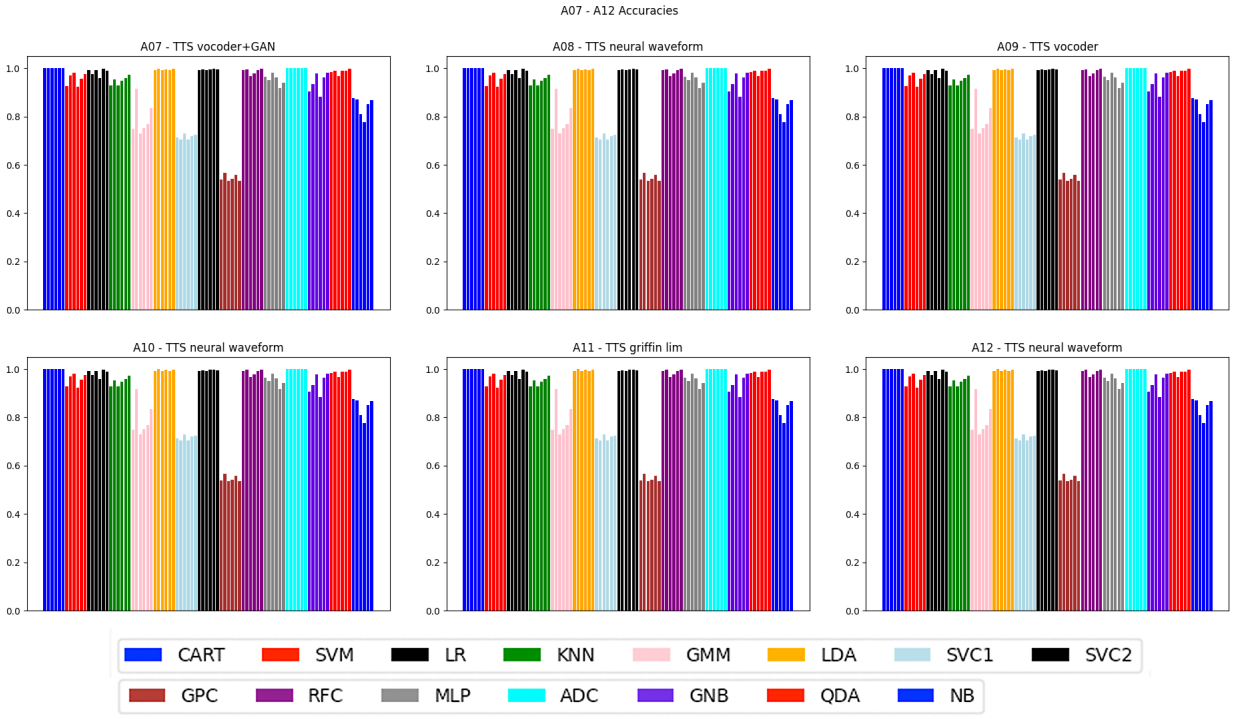

Figure 7. All accuracies per classifier and spoofing technique A07-A12; each bin per color shows the 1, 1*, 2, 2*, 3 and 3* accuracy values

.

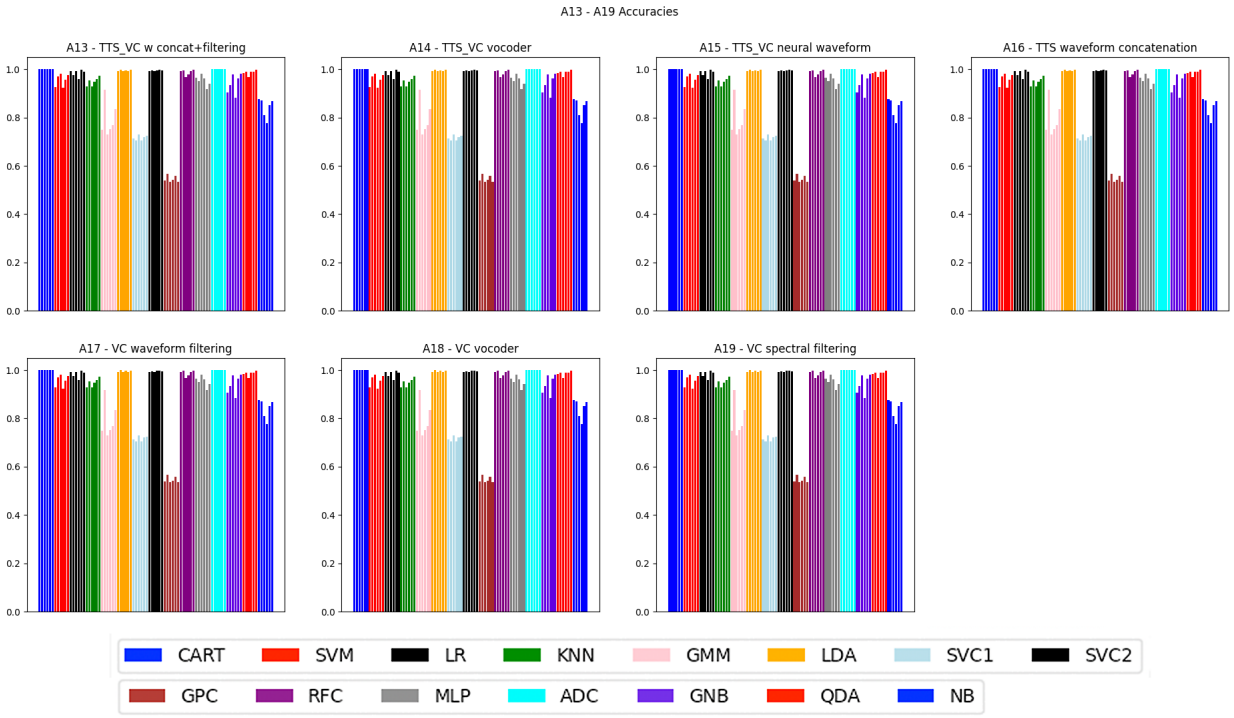

Figure 8. All accuracies per classifier and spoofing technique A13-A19; each bin per color shows the 1, 1*, 2, 2*, 3 and 3* accuracy values

.

References

[1] Ehab A AlBadawy, Siwei Lyu, and Hany Farid. Detecting ai-synthesized speech using bispectral analysis. In *CVPR Workshops*, pages 104–109, 2019. 7

[2] Muteb Aljasem, Aun Irtaza, Hafiz Malik, Noushin Saba, Ali Javed, Khalid Mahmood Malik, and Mohammad Meharmohammadi. Secure automatic speaker verification (sasv) system through sm-altp features and asymmetric bagging. *IEEE Transactions on Information Forensics and Security*, 16:3524–3537, 2021. 7

[3] PR Aravind, Usamath Nechiyil, Nandakumar Paramparambath, et al. Audio spoofing verification using deep convolutional neural networks by transfer learning. *arXiv preprint arXiv:2008.03464*, 2020. 7

[4] Sebastiano Battiato, Oliver Giudice, and Antonino Paratore. Multimedia Forensics: discovering the history of multimedia contents. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, pages 5–16, 2016. 1

[5] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 132–137, 2020. 7

[6] Rohan Kumar Das, Tomi Kinnunen, Wen-Chin Huang, Zhenhua Ling, Junichi Yamagishi, Yi Zhao, Xiaohai Tian, and Tomoki Toda. Predictions of subjective ratings and spoofing assessments of voice conversion challenge 2020 submissions. *arXiv preprint arXiv:2009.03554*, 2020. 2

[7] Rohan Kumar Das, Jichen Yang, and Haizhou Li. Data augmentation with signal companding for detection of logical access attacks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6349–6353. IEEE, 2021. 2, 7

[8] Yang Gao, Tyler Vuong, Mahsa Elyasi, Gaurav Bharaj, and Rita Singh. Generalized spoofing detection inspired from audio generation artifacts. *arXiv preprint arXiv:2104.04111*, 2021. 7

[9] Alejandro Gomez-Alanis, Antonio M Peinado, Jose A Gonzalez, and Angel M Gomez. A light convolutional gru-rnn deep feature extractor for asv spoofing detection. In *Proc. Interspeech*, volume 2019, pages 1068–1072, 2019. 7

[10] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Fighting Deepfake by Exposing the Convolutional Traces on Images. *IEEE Access*, 8:165085–165098, 2020. 1

[11] Guang Hua, Andrew Beng Jin Teoh, and Haijian Zhang. Towards end-to-end synthetic speech detection. *IEEE Signal Processing Letters*, 28:1265–1269, 2021. 7

[12] Lian Huang and Chi-Man Pun. Audio replay spoof attack detection by joint segment-based linear filter bank feature extraction and attention-enhanced densenet-bilstm network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1813–1825, 2020. 7

[13] Nils Hulzebosch, Sarah Ibrahimi, and Marcel Worring. Detecting CNN-generated facial images in real-world scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 642–643, 2020. 1

[14] Ziyue Jiang, Hongcheng Zhu, Li Peng, Wenbing Ding, and Yanzhen Ren. Self-supervised spoofing audio detection scheme. In *INTERSPEECH*, pages 4223–4227, 2020. 7

[15] Tomi Kinnunen, Md. Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In *Proc. Interspeech 2017*, pages 2–6, 2017. 1

[16] Xu Li, Na Li, Chao Weng, Xunying Liu, Dan Su, Dong Yu, and Helen Meng. Replay and synthetic speech detection with res2net architecture. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6354–6358. IEEE, 2021. 7

[17] Haoxin Ma, Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Chenglong Wang. Continual learning for fake audio detection. *arXiv preprint arXiv:2104.07286*, 2021. 7

[18] Momina Masood, Marriam Nawaz, Khalid Mahmood Malik, Ali Javed, and Aun Irtaza. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *arXiv preprint arXiv:2103.00484*, 2021. 2, 7

[19] Joao Monteiro, Jahangir Alam, and Tiago H Falk. Generalized end-to-end detection of spoofing attacks to automatic speaker recognizers. *Computer Speech & Language*, 63:101096, 2020. 7

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[21] Arun Kumar Singh and Priyanka Singh. Detection of ai-synthesized speech using cepstral & bispectral statistics. In *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 412–417. IEEE, 2021. 7

[22] Christian J. Steinmetz and Joshua D. Reiss. pyloudnorm: A simple yet flexible loudness meter in python. In *150th AES Convention*, 2021. 2, 4

[23] Catherine Stupp. Fraudsters used ai to mimic ceo's voice in unusual cybercrime case. *The Wall Street Journal*, 30(08), 2019. 1

[24] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. Technical report, University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2017. 2

[25] Luisa Verdoliva. Media Forensics and DeepFakes: An Overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 1

[26] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. Deepsonar: Towards effective and robust detection of ai-synthesized fake voices. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1207–1216, 2020. 2, 5, 7

[27] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, et al. Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 64:101114, 2020. 1, 2

[28] Zhenzong Wu, Rohan Kumar Das, Jichen Yang, and Haizhou Li. Light convolutional neural network with feature genuinization for detection of synthetic speech attacks. *arXiv preprint arXiv:2009.09637*, 2020. 7

[29] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Sixteenth annual conference of the international speech communication association*, 2015. 1

[30] Jiangyan Yi, Ye Bai, Jianhua Tao, Zhengkun Tian, Chenglong Wang, Tao Wang, and Ruibo Fu. Half-truth: A partially fake audio detection dataset. *arXiv preprint arXiv:2104.03617*, 2021. 7

[31] You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941, 2021. 2, 7

[32] Zhenyu Zhang, Xiaowei Yi, and Xianfeng Zhao. Fake speech detection using residual network with transformer encoder. In *Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security*, pages 13–22, 2021. 2, 7