Synthetic Audio Detection: Combination of Audio Features and Deep Learning Models

Dr. Tahsina Farah Sanam
Tasnim Nishat Islam, Imtiaz Ahmed, Md Boktiar Mahbub Murad, Swojan Datta, Md.Fahim Abid,
Sayonto Khan, Utsab Saha, Voktho Das, Tahsin Saad Chowdhury,Farsia Kawsar Chowdhury, Naima Tasnim
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

[1]

[2]

***Abstract*—Although comparatively shallower ML networks with hand crafted features like MFCC, mel spectrogram have dominated automatic speaker recognition tasks for a long time, in recent years end to end DNN architectures are outperforming them in practical applications. In this paper, we explored such possibilities to correctly categorize the particular dataset provided in Sp Cup 2022 into multiple known and unknown classes. Starting from the traditional feature extraction to modern DL methods, we have tried our several training pipelines and explored their appropriateness to our particular scenario. Afterwards, we have shown that raw audio data based DL pipelines such as YAMnet, resnet style tssd, and our proposed DilatedIncNet can generalize excellently. Finally, In order to more robust performance majority voting based final prediction is introduced to decrease variance in their inferences.**

***Index Terms*—yamnet, synthetic, audio, resnet, spectrogram**

I. Introduction

Due to the obvious importance of data security, the ability to recognize synthetic audio has become a reasonable concern in recent years. Bio-metric signals such as voice can be altered through synthetic audio creation. Artificial Intelligence (AI) and especially deep learning driven technologies make potentially harmful manipulations incredibly realistic and persuasive. As a preventive measure we have focused on synthetic audio speech detection in this work.

The technique of listening to and evaluating audio signals is known as audio classification and sound categorization.The detection of synthetic audio is a challenging task since there is a wide variety of possible methods for generating false speech. Various audio properties, such as MFCC (Mel Frequency Cepsrtal Coefficients), LPC (Linear Predictive Coding), PSD (Power Spectral Density), Box-Windowing and so on have traditionally been employed to detect synthetic audio [1]. However, recognizing synthetic audio just with these features may not be enough for certain applications . In this paper, we have focused on building an end to end pipeline where recombination of audio feature extraction and deep learning methods are harmonically combined.

As mentioned before, Magnitude-based features such as MFCC have been used extensively in speech signal analysis. However, they do not perform as well as phase related features in detecting synthesized or converted speech [2]. Based on the assumption that modulation features derived from magnitude/phase spectrum may be able to detect temporal artifacts caused by frame-by-frame processing in the synthesis of speech signals, these features are fused in [1] in order to differentiate synthetic speech from human speech. Among hand-crafted features, constant Q cepstral coefficient (CQCC) has been found to be the best choice, which is also the baseline feature in the ASVspoof2019 challenge [3]. In recent years, the manual feature extraction process has been significantly replaced by deep learning. It has been shown that deep models, even with raw wave forms, can outperform traditional methods [4].
Early investigations on identifying synthetic speech were done on data sets developed for this specific purpose. The issue with this methodology is that a *priori* knowledge of the data does not reflect the practical scenario in which the origin and nature of the data can never be known beforehand [5]. Regardless of the source of data, traditional feature extraction methods can provide useful insights, whereas deep neural network models can exploit this knowledge to learn distinguishable characteristics of speech data better than traditional classifier networks.

II. Dataset Description

The dataset consists of one training data and two evaluation data. A sample audio wave is shown in Figure 2. All the audios have sampling rate of 16000 and number of channels = 1. The training data has a length of (min)1.35s - (max)14.76s, for evaluation part 1, (min)2.43 - (max)18.50s. For training dataset, we have average length of 8.255s, 6.426s, 6.359s, 8.136s, 5.6159s, 6.79s consecutively for class 0-5, where class 5 is the partial representation of unknown data. Inspecting the zero crossing rates, we have 25035, 26186, 26066, 34955, 12960, 26439 for each class.

[1]Special gratitude to Hafiz Imtiaz, PhD, Associate Professor, Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology has been very generous to help and guide throughout the work

[2]The team Registration ID is: 27592, Team name: ”Students Procrastinating” Please start with ’README.md’ to run our code that has been submitted

III. Feature Extraction

Because data preparation is so important, determining the optimal feature for every given issue solution is vital. In deep learning algorithms or network learning processes, feature extraction plays a critical role in anticipating the correct conclusion. A useful feature aids the network model's ability to learn in the proper and accurate manner.

A. Mel spectrogram

Mel spectrogram is a strong technique for classifying audio in this procedure. A spectrogram that has been converted to a Mel scale is known as a Mel spectrogram. The Mel scale is a perceptual scale of sounds that listeners interpret to be equally spaced apart. A spectrogram is a visual representation of a signal's frequency spectrum, where the frequency spectrum of a signal refers to the frequency range that the signal covers. According to [6] and [7], people do not detect frequencies on a linear scale, therefore the Mel scale matches how the human ear works. At lower frequencies, humans are better at perceiving differences than at higher frequencies.

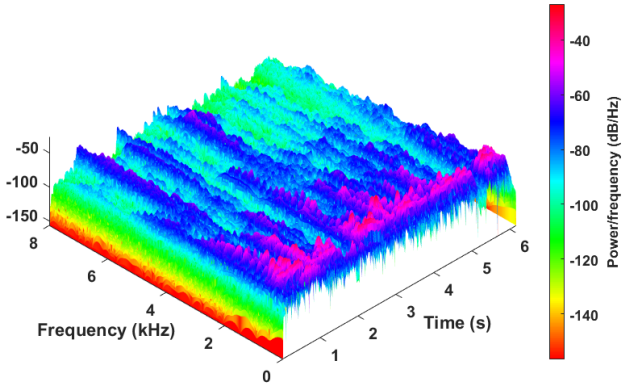

Fig. 1. Mel-spectrogram of an audio signal

B. Other Feature Extraction

1) Power Spectral Density, 2) Box Windowing, 3) Fast Fourier Transform, 4) Short Time Fourier Transform, 5) Constant-Q transform, 6) Chromatogram, and 7) Raw Audio are the additional features we have explored for extracting from our dataset configuration.

Here we have extracted PSD of raw audio and gave it to the network for train, which didn't come up with a good result. Then we took STFT of raw audio and tried to use it as feature, which eventually failed in scoring better. We have also used a combination of *256 samples* box windowing overlapping 16 samples of which we did stft and got the PSD, but it could not perform well in this dataset setup as well. We also tried MFCC, which likewise failed miserably. We also have experimented with CQT, fft and Chromatogram. All of which produced unsatisfactory results in the assessment dataset.

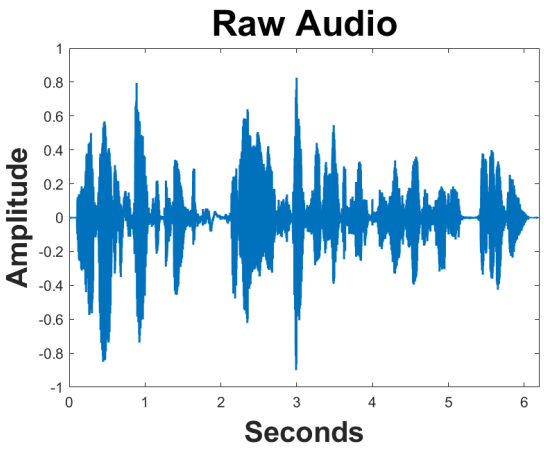

Fig. 2. plot of sample audio

IV. Model Description

For our training we have mainly considered CNN based architectures on raw waveform and spectrograms. In synthetic speech detection an important feature is digital footprint left in audio which does not contain any semantic information. So, we have implemented our models on 1D data (raw waveform) along with 2D data (spectrograms) for making use of voice generation related characteristics embedded in them. We chose a supervised learning approach to enable us to learn better representation of the known class, and differentiate with the random distribution in the unknown class. Later we experimentally ensembled our best performing models to reduce variance among their differences in class representations. We have gone through different kinds of methodologies to reveal the best training pipeline for our problem. Firstly, our preliminary model was a multi stage classifier utilizing type distinctive features, zero crossing rates and spectral centroid of the windowed audio signal. As the evaluation dataset has noise in part 2, we have augmented noise in the dataset where we included reverberation and filter distortions. Secondly, we tried out transfer learning using YAMnet. For this two distributions of datasets were used. Here, we used an augmented dataset with 2000 data for each class (balanced). Later, we trained the YAMnet with class 5 data having the majority class and other classes having less number of samples. Thirdly, to make our model more robust against unknown data and noise as well as accurate for known classes we have introduced variation in our unknown class using different combinations of publicly available dataset and different algorithms and also augmented distortions to it. At this point, we also tried LSTM based WaveNet, which also failed us scoring better results. Finally for the combined dataset, we have tried out models including but not limited to YAMnet [8], Resnet style TSSD [9], DilatedIncNet and combined their results with majority voting method. Here, resnet style tssd and DilatedIncNet is

trained on raw wave data. So we fine tuned our best performing models and ensembled them using majority voting to generate the best predictions.

Firstly, our preliminary model was of multi stage classifier utilizing type distinctive features, zero crossing rates and spectral centroid of windowed audio signal. As the evaluation dataset has noise in part 2, we have augmented noise in the dataset where we included reverberation and filter distortions.

Secondly, we tried out transfer learning using YAMnet. For this two distributions of datasets were used. At first, we have used 2000 data for each class (balanced) where for the known class additional 1000 was augmented noisy audio, and for the unknown class we added 1000 random audio sample from the publicly available ASVPoof dataset [10] with our given unknown 1000 samples. Secondly, we have trained the YAMnet with class 5 data having the majority class and other classes having less number of samples which gives us more robust result.

Thirdly, to make our model more robust against unknown data and noise as well as accurate for known classes we have introduced variation in our unknown class using different combinations of publicly available dataset and different algorithms and also augmented distortions to it. At this point, we also tried LSTM based WaveNet, which also failed us scoring better results.

Finally for the combined dataset, we have tried out models including but not limited to YAMnet [8], Resnet style TSSD [9], DilatedIncNet and combined their results with majority voting method. Here, resnet style tssd and DilatedIncNet is trained on raw wave data. So we fine tuned our best performing models and ensembled them using majority voting to generate the best predictions.

A. Preparing the mixed dataset

At first we trained using 6000 data in 6 classes. Although the models were performing well, we had a hypothesis that the models were not doing well in the unknown classes. To handle that, we introduced variation in the unknown classes, and fine tuned our experimentation to get the best result out of our models. For the YAMnet transfer learning part as mentioned our balanced dataset was created by using 2000 data per class. For known classes 1000 clean data and 1000 repeating them and adding augmentations(300 noise injection, 300 reverberation, 300 mp3 compression and 100 noise and reservations). The provided matlab script was used for this augmentation. The unknown classes followed the same process, just instead of repeating we took 500 data from each of the ASVspoof19 [10] and Librispeech [11] datasets to introduce more variability in the unknown class. Secondly, for the YAMnet we tried making the class 5 majority by taking out some samples from known classes. Afterwards, at our final resnet style tssd and DilatedInceptionNet we have used 150 data each from the ASVspoof19 and Librispeech datasets in the unknown class making 1000 data per known class and 1300 in the unknown class. We have chosen this imbalance in class 5 as intuitively in real life setting also, the unknown data will be introduced more often than the known dataset.

B. YAMNet

Our used model, YAMNet (Yet Another Mobile Network), is a popular pre-trained deep neural network or classifier for audio signal processing that takes audio signal or signal features as input and predicts on the given signal independently. Yamnet was originally trained on 521 class AudioSet [8] ontology dataset.Yamnet is used here as a high level feature extractor. We used MobileNet v1 architecture for our model.

YAMNet Preprocessing: For getting the best learning and result out from the YAMNet, we have extracted spectrogram feature out of raw audio signal. This feature extraction can help YAMNet for better learning and prediction. Then we have take these features one step ahead for giving a boost to our classifier. We then pass the spectrogram to mel log bank filter and get the Mel-spectrogram fetaures which was later used with weights and biases for the best result.

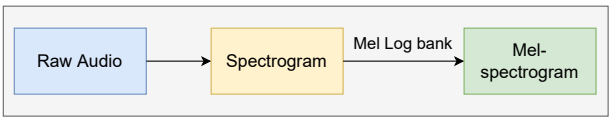

Fig. 3. YAMNet Preprocessing

C. DilatedIncNet

This network is designed for 1D raw waveform data using Dilated 1D convolution. At first the input is passed through a conv1d block followed by BatchNormalization, RELU and MaxPooling layer. Then successive DilatedConvModule blocks with different filters are added to extract inherent features in a sequential Manner. At last, a GlobalAveragePooling block is added before flattening and passing through the Dense layers for final prediction. The DilatedConvModule consists of multiple Conv1d, BatchNormalization, RELU with variable kernel size and dilation rate followed by a concatenation layer to extract features from different stages. The full model is shown in figure 4. Here 'Dilated Convolution' block in the figure represents 'Dilated Convolution' Module.

D. Resnet style TSSD

We have followed the basic architecture of the Resnet Style TSSDNet for implementing this model. It consists of a repeating Residual Block with multiple conv1d layers followed by BatchNormalization and RELU. At last, the input and final output is concatenated followed by a RELU. This Residual block is used 4 times with variable filters in the Resnet architecture to generate desired predictions.

V. Experiments and Results

We have tried our various models starting from various ML algorithms based on hand crafted features to end to end DL methods. Among them scores from the best performing models are included in Table 1. For our DL models we have

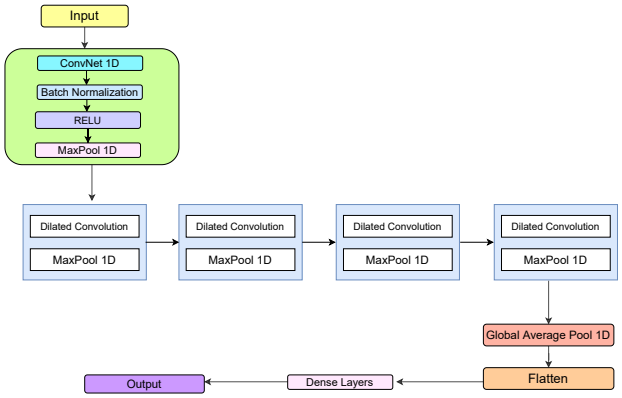

Fig. 4. DilatedIncNet

used Adam as optimizer, Sparse categorical Cross Entropy as the loss function, Accuracy as the metrics and learner ReduceLROnPlateau in all of our training setup. Yamnet has been trained for 30 epochs and Other nets were mostly trained for 100 epochs.Our final proposition is the yamnet, tssd-resnet and Dilatedinceptnet, we have ensembled the three models' results using majority voting. We have prioritized the model of tssd-resnet if half of the models disagree on any result. In the table we have presented some of our experiments which were done on different feature extraction, different models, and individual results on the final models as well as the ensembled versions. In the training process we have experimented with the cross validation numbers, epoch numbers, learning rates, dataset combinations, which were highlighted in the table.We chose our best performing models in terms of accuracy of detecting each part of the dataset, we could access the partial open evaluation set and have judged the score from Codalab. The performance matrix is defined by

$$Score = 0.7 * Part1_{acc} + 0.3 * Part2_{acc} \quad (1)$$

Table 1. Experimental Result Summary

Model	Part1	Part2	Score
mfcc-wavenet	0.2707	0.2012	0.2498
stft-psd-wavenet	0.3510	0.3527	0.3515
resnet-fft	0.7785	0.7074	0.7571
yamnet-tssd-ensemble	0.8222	0.6395	0.7673
tssd-resnet-awgn	0.8821	0.8834	0.8825
DilatedIncNet(50 epoch)	0.9067	0.9221	0.9113
resnet-tssd(50 epoch)	0.9041	0.9392	0.9146
yamnet-imbalanced-noisy	0.9631	0.8438	0.9273
resnet-tssd-mixed	0.9365	0.9384	0.9371
resnet-tssd-cross-val-original	0.9461	0.9611	0.9506
resnet-tssd-cross-val-mixed	0.9720	0.9821	0.9750
yam-DilatedIncNet-tssd-res	**0.978**	**0.985**	**0.9801**

VI. Conclusion

In this paper, we have shown that by combining decisions from raw waveform and spectrograms based end to end Deep Learning pipelines we can implement a robust speaker recognition system. We have also explored that the unknown class became more distinguishable from the known class as we have added external dataset into the unknown class. Here our CNN based architecture has shown excellent generalization capability provided with proper dataset and training methods. In near future, we wish to add semi supervised themes to our supervised experiments for making the solution more robust and generalized of detecting synthetic audio.

References

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

[2] Z. Wu, E. S. Chng, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[3] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[4] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, 2021.

[5] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.

[6] R. R. Huizen and F. T. Kurniati, "Feature extraction with mel scale separation method on noise audio recordings," *arXiv preprint arXiv:2112.14930*, 2021.

[7] S. Umesh, L. Cohen, and D. Nelson, "Frequency warping and the mel scale," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 104–107, 2002.

[8] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[9] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, 2021.

[10] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," 2019. [Online]. Available: https://arxiv.org/abs/1911.01601

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.