FAKE AUDIO DETECTION BASED ON UNSUPERVISED PRETRAINING MODELS

Zhiqiang Lv, Shanshan Zhang, Kai Tang, Pengfei Hu

TEG AI, Tencent Inc, Beijing 100193, China

{zhiqianglv, susanszhang, aydentang, alanpfhu}@tencent.com

ABSTRACT

This work presents our systems for the ADD2022 challenge. The ADD2022 challenge is the first audio deep synthesis detection challenge, which aims to spot various kinds of fake audios. We have explored using unsupervised pretraining models to build fake audio detection systems. Results indicate that unsupervised pretraining models can achieve excellent performance for fake audio detection. Our final EER results for low-quality fake audio detection and partially fake audio detection are 32.80% and 4.80% relatively. For partially fake audio detection, our results ranked first in the competition. Even trained with totally mismatched data, our method still generalizes well for partially fake audio detection.

Index Terms— Fake audio detection, Unsupervised pretraining models, XLS-R, ECAPA-TDNN

1. INTRODUCTION

With the rapid development of speech synthesis and voice conversion technologies, generating fake audios that are hard to distinguish by humans becomes very easy. Softwares and applications such as Deepfake supply a simple way to make fake news or videos, which may lead to great social stability problems. ASVspoof challenges have made a lot of efforts in automatic speaker verification spoofing detection. In ASVspoof 2021, detecting deepfake speech generated by TTS and voice conversion is also included as a track. Considering the complexity of audios in realistic scenarios, it is still very hard to distinguish fake audios from real ones, especially for partially fake audios [1].

The ADD2022 challenge [2] focuses on detecting various kinds of fake audios. It includes three tracks: low-quality fake audio detection (Track1), partially fake audio detection (Track2) and audio fake games (Track3). Track1 is similar to deepfake audio detection in ASVspoof 2021, but more challenging due to diverse background noises and disturbances. Track2 aims at detecting small fake clips. Fake clips can be speech synthesized, voice converted or bona fide ones from other utterances. Track3 is an attack and defense simulation, exploring the limit of fake audio detection technologies.

This paper presents our work for Track1 and Track2 of the ADD2022 Challenge. In this paper, we will introduce our methods for both low-quality and partially fake audio detection. Besides classical supervised classification methods, we also explored to use unsupervised pretraining models to construct fake audio detection systems.

As is known to all, unsupervised pretraining has played an important role in artificial intelligence areas such as NLP and image processing. Several pretraining models such as BERT have made great success and change the way to build AI solutions. For speech processing, more and more unsupervised or self-supervised methods also gain much improvement. Our work used Wav2vec-style models to build fake audio systems. Preliminary experiments on Track1 and Track2 adaptation datasets have obtained superior results with unsupervised pretraining models. However, our system didn't generalize well on Track1 evaluation, resulting in a final EER of 32.80%. As for Track2, our system reached an EER of 4.80% and ranked first in the final evaluation.

Rest of this paper is organized as below: section 2 is about related work, section 3 is about datasets, section 4 introduces our methods for low-quality and partially fake audio detection, section 5 is about experiment setup. In section 6, results are presented and some further discussions have been made. Section 7 are our conclusions about this work.

2. RELATED WORK

ASVspoof challenges have played very active roles in accelerating progress in spoofed speech detection. Past ASVspoof challenges (ASVspoof 2015, ASVspoof 2017, ASVspoof 2019) mainly aimed to detect fake audios in terms of speaker verification. However, fake audios with small clips hidden in bona fide audios is usually much more common. ASVspoof 2021 added a track of deepfake audio detection ignoring speaker verification. ADD2022 challenge is similar to the new-added track in ASVspoof 2021 to some degree.

For deepfake audio detection in ASVspoof 2021, [3, 4, 5] have adopted several kinds of handcrafted features for fake audio detection, including product spectral cepstral coefficient (PFCC) [6], linear frequency cepstral coefficient features (LFCC) [7], DCT-DFTspec [8], log-linear filterbank energy features and constant-Q transform features [9, 10, 11, 12, 13, 14, 15]. Previous works indicate LFCC and constant-Q features are quite suitable for fake audio detection.

On the other hand, [16, 17, 18] adopted raw waveforms as inputs to build end-to-end detection systems. ASVspoof 2021 baselines are also based on LFCC, CQCC and raw waveform features.

For better performance, most systems from ASVspoof 2021 participators adopted various data augmentation. Codec augmentation are used in [3, 19, 20, 5, 21]. Noise augmentation, time warping, SpecAug and impulse response augmentation work very well in [5, 20, 19].

ASVspoof 2021 baselines used traditional GMM, light convolutional neural network (LCNN) [22] and end-to-end Rawnet2 [16] models for fake audio detection. TDNN [23] and squeeze-and-excitation residual network-18 (SE-ResNet-18) [24, 25] are adopted in [3, 4, 20]. [17] built an end-to-end spectro-temporal graph attention network from raw waveforms. In facts, most models of ASVspoof 2021 can be categorized into Resnet-style networks, Rawnet-style end-to-end networks and LCNN-style networks, with some minor changes. [18] tried to use automatical network architecture search to obtain a optimized model for speech deepfake and spoofing detection.

Unsupervised or self-supervised pretraining models have achieved significant improvement on a lot of speech tasks such as ASR [26, 27, 28]. Some researchers also try to extract representation using pretraining models or fine-tune pretraining models with small amount of labeled speech [29]. Preliminary experiments have shown that pretraining models carry useful information from large-scale unlabelled speech. Researchers have explored using unsupervised pretraining technologies to boost language identification[30]. As far as we know, no related works have used unsupervised pretraining models for fake audio detection. In this work, we try to use open-sourced unsupervised pretraining models to build fake audio detection solutions.

3. DATASETS

The ADD2022 Challenge aims to distinguish fake audios from bona fide ones. Track1 is low-quality fake audio detection (LF). Fake audios of this track are fully fake utterances generating by text-to-speech or voice-conversion algorithms. Track2 is much harder, detecting partially fake audios (PF). Partially fake audios are generated by manipulating original bona fide utterances with real or synthesized audio clips. Track3 is audio fake game (FG), in which we didn't participate.

The training dataset consists of 3012 bona fide utterances and 24072 fully fake utterances. The development dataset consists of 2307 bona fide utterances and 26017 partially fake utterances. The ADD2022 Challenge also provides two adaptation datasets for both Track1 and Track2. The adaptation dataset of Track1 consists of 300 bona fide utterances and 700 fully fake ones, while the adaptation dataset of Track2 only contains 1052 partially fake utterances. In fact, the training dataset size is about 23.71 hours, while the development dataset size is about 24.88 hours. Therefore, we re-adjust the datasets like Table 1.

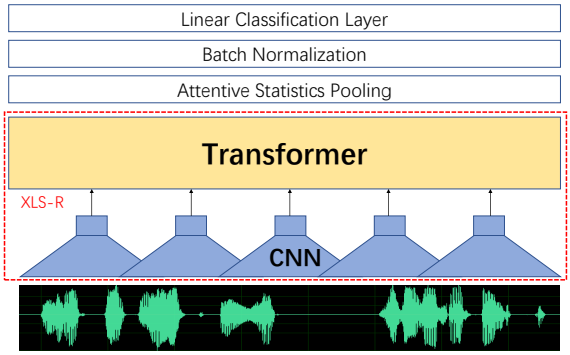

Fig. 1. *XLS-R Fake Audio Detection System.*

Table 1. *Utterance counts of re-adjusted datasets.*

	train	dev	adaptation1	adaptation2
genuine	5088	231	300	300
fake	47487	2602	700	1052

As shown in Table 1, we combine 90% of the development dataset with the training dataset into a new training dataset. Only 10% of the development dataset is kept. Besides, all the 300 genuine utterances are added into the adaptation dataset for Track2 so as to compute Equal Error Rate (EER). The adaptation dataset for Track1 remains unchanged.

4. METHODS

For Track1 of fully fake audio detection, we have constructed ECAPA-TDNN models [31] to classify fake and genuine audios. The model is a big version that contains 21.2M parameters as in Speechbrain [1]. As for feature extraction, we have adopted widely used LFCC features as well as filterbank features.

For unsupervised pretraining models, we utilize Wav2vec-style models. Specially, we have used XLS-R models [27] pre-trained on 436k hours of unlabeled speech. XLS-R models are almost the biggest open-source pretraining models for speech processing. We use XLS-R models to extract speech representations or embeddings. Then we stack an attentive statistic pooling layer on output embeddings. Finally, a fully-connected classification layer is added on the top. The XLS-R fake audio detection system is illustrated in Figure 1.

The classical ECAPA-TDNN and XLS-R based models are in fact classification models. The key is to map variable-length input audios into fixed-dimension embeddings. This

[1] https://github.com/speechbrain/speechbrain

procedure is done by attentive statistic pooling. Before pooling, we have to obtain good enough representations of original audios. For ECAPA-TDNN models, representations are learned from scratch. For XLS-R based models, representations are decided by weights learned from large-scale unlabelled speech. In addition, we can also fine-tune the XLS-R classification model with labelled data.

5. EXPERIMENTAL SETUP

Track1 can be regarded as traditional classification problems like speaker identification or language identification. Therefore, configurations in speaker and language identification can be used in Track1. For Track1 of fully fake audio detection, we randomly sampled segments of 3 seconds from the re-adjusted training dataset to make batches. We adopted AAM loss for ECAPA-TDNN and XLS-R models, with 2 output classes. For ECAPA-TDNN models, the batch size is set to 32, learning rate 0.0001. For XLS-R models, we have adopted two versions: 300 million parameters and 1 billion parameters. For the biggest 2 billion parameter version, we didn't manage to put it in a V100 GPU. Batches for the two versions of XLS-R models are 8 and 4 relatively, learning rate 0.0001 and 0.00001.

Augmentation are also used, including speed perturbation, adding noises and reverberations using MUSAN, time warping and SpecAug. Two versions of features including filterbank features and LFCC features are adopted. As for XLS-R models, only raw waveforms are valid as inputs.

Partially fake audio detection of Track2 is a little more complicated. First of all, we tried to employ models trained in Track1 to detect fake audio segments. Unfortunately, this method doesn't seem to work at all. Therefore, we proposed a simulation method to generate fake audios during training. We kept genuine utterances and less than 10% of fake audios from the re-adjusted training dataset. The we randomly insert audio clips from other utterances to target utterances with a probability of 0.5. Durations of inserted audio clips are sampled from 0.5s to 1s uniformly. Each segment of training batches is also about 3s. All manually manipulated utterances and other fake ones without modification are labelled as fake. Only unmodified genuine utterances are labelled as genuine.

6. RESULTS

6.1. Track1 (LF): Low-quality fake audio detection

Detailed experimental setup is described as above. Table 2 reports our individual system performance for low-quality fake audio detection. Results are shown in terms of EER.

Results in Table 2 indicate that for ECAPA-TDNN models, LFCC features outperform FBANK features slightly. XLS-R based models also show good performance even with freezing parameters in XLS-R, which means XLS-R learned relatively high-quality representations of raw inputs. If we fine-tune XLS-R models with labelled data, XLS-R based models can reach a very low EER of 0.33%, which outperforms ECAPA-TDNN models significantly.

Table 2. *EER for Track1 adaptation dataset and evaluation*

	adaptation	evaluation
ECAPA-TDNN FBANK	6.33%	-
ECAPA-TDNN LFCC	6.00%	37.07%
XLS-R 300M frozen	7.00%	-
XLS-R 300M fine-tuned	2.57%	**34.02%**
XLS-R 1B frozen	7.33%	-
XLS-R 1B fine-tuned	**0.33%**	34.17%

However, when applying our models on the evaluation dataset, we can only get a minimum EER or 34.02%, which is much worse than that on the adaptation dataset. The reason may be that there exists quite much mismatch between the adaptation and evaluation dataset. The best EER of Track1 evaluation is only 21.7% among all the participators. The possible mismatch brings more challenges for us to build a better detection system before final evaluation phase begins.

Fine-tuning a pre-trained XLS-R model with more parameters works best on the adaptation dataset. While on the evaluation dataset, bigger models don't make much difference. Big performance gaps between the adaptation dataset and evaluation indicate that our XLS-R based models maybe over-fitted.

In fact, genuine utterances in the training set is far less than fake ones. That may lead to serious class imbalance during training. To obtain better performance for evaluation, we try to sample genuine and fake utterances with different weights. We set the sampling weight ratio to 5:1 between the two classes. Results after sampling weights adjusting is listed in Table 3.

Table 3. *Track1 EER after adjusting class sampling weights*

	adaptation	evaluation
ECAPA-TDNN LFCC	5.00%	34.69%
XLS-R 300M fine-tuned	**2.00%**	**32.80%**

From results in Table3, we can see that training class imbalance indeed affects final EER. After assigning bigger weights to genuine utterances, EER results on both datasets outperform those in Table 2. However, due to the tight challenge schedule, we didn't manage to experiment with more sampling weights.

6.2. Track2 (PF): Partially fake audio detection

For partially fake audio detection in Track2, fake clips are hidden in genuine utterances. In addition, not all fake clips are fake ones. Some fake clips are taken from other genuine

utterances. Therefore, it is very difficult to spot all kinds of fake clips.

First of all, we try to utilize models trained in Track1 to detect fake clips segmentally. Our fake audio detection system will predict whether the segment is fake every 3 seconds. Then we choose the highest fake probability as the whole utterance's fake probability. Results of applying models from Track1 are in Table 4.

Table 4. *Track2 EER of applying Track1 models directly*

	adaptation
ECAPA-TDNN FBANK	50.67%
XLS-R 300M fine-tuned	69.33%
XLS-R 1B fine-tuned	71.33%

From results in Table 4, we can see that directly applying Track1 models on Track2 doesn't work at all. We have to come up with another method to spot fake clips. Therefore, we simulate to generate fake clips in genuine utterances, as mentioned in experimental setup. Two key points of detecting fake clips are: 1) to find discontinuity on time axis caused by inserting a clip, especially for genuine clips 2) to find inserted fake clips generating by speech synthesis. Results of partially fake audio detection are in Table 5.

Table 5. *Track2 EER with simulated partially fake data*

	adaptation
ECAPA-TDNN FBANK (w/o fake audios)	11.33%
XLS-R 300M fine-tuned (w/o fake audios)	6.67%
XLS-R 300M fine-tuned (w/ fake audios)	8.00%
XLS-R 1B fine-tuned (w/o fake audios)	**3.33**%

From results in Table 5, we can see that simulation with fake audios leads to performance degradation. Partially fake audio simulation with only genuine utterances achieves the best EER of **3.33%** using the XLS-R 1B based model. All XLS-R based models outperform ECAPA-TDNN models on the Track2 adaptation dataset. Using the XLS-R 1B model, we could achieve EER of **4.90%** for evaluation with the best checkpoint. Averaging detection results from top 3 best checkpoints, we achieved final EER of **4.80%** for final evaluation, which ranks first among all the participators.

In fact, the best system by simulating partially fake audios only spots discontinuity on time axis. We don't manage to spot synthesized fake clips. The reason may be that we can't enumerate all kinds of fake types with so small training dataset. So we give up further experiments on data simulation with fake audios.

The big performance gap in Track1 between the adaptation and evaluation dataset inspires us that our excellent results of Track2 may be another kind of over-fitting. So we design extra experiments with totally mismatched training data for Track2. The CommonLanguage dataset [2] is utilized for training partially fake audio detection systems. This dataset consists of 45 hours speech from 45 different languages. EER on the adaptation dataset is in Table 6. We can see that using totally mismatched data, our methods still work well. Our XLS-R based models trained with CommonLanguage even outperform ECAPA-TDNN models trained using the ADD2022 challenge dataset significantly. This experiment has demonstrated: our method for partially fake audio detection doesn't depend on data heavily.

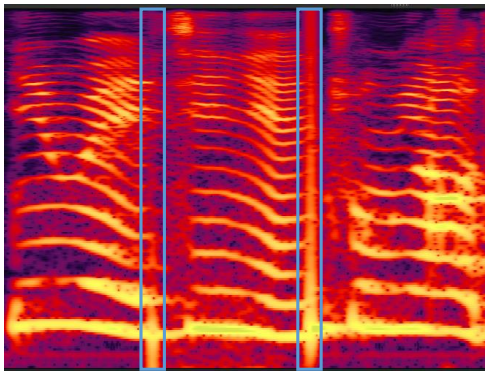

Fig. 2. *Changes of partially fake audio frequency spectrums.*

Table 6. *Track2 EER trained with CommonLanguage*

	adaptation
ECAPA-TDNN FBANK (w/o fake audios)	14.67%
XLS-R 300M fine-tuned (w/o fake audios)	5.33%
XLS-R 1B fine-tuned (w/o fake audios)	**5.00**%

We have also explored the reason why our method works. From frequency spectrums of Track2 adaptation dataset, we find spectrum blurs or sudden changes between genuine and fake clips. This phenomenon is illustrated in Figure 2. We suppose the sudden frequency spectrum change makes our method work finally.

7. CONCLUSIONS

We have presented our work for the ADD Challenge Track1 and Track2. In this paper, we explored using unsupervised pretraining models to construct fake audio detection solutions. XLS-R models with 300 million and 1 billion parameters have been used. Our XLS-R based models have shown superior performance on Track1 and Track2 adaptation datasets, which significantly outperform tradition totally supervised models. We achieved EER of 4.80% for Track2 evaluation, ranking first in this track.

[2] https://zenodo.org/record/5036977/files/CommonLanguage.tar.gz

8. REFERENCES

[1] J. Yi, Y. Bai, J. Tao, H. Ma, Z. Tian, C. Wang, T. Wang, and R. Fu, "Half-Truth: A Partially Fake Audio Detection Dataset," in *Proc. Interspeech*, 2021, pp. 1654–1658.

[2] J. Yi, R. Fu, J. Tao, S. Nie, H. Ma, C. Wang, T. Wang, Z. Tian, Y. Bai, S. Liang, S. Wang, S. Zhang, X. Yan, L. Xu, and H. Li, "ADD 2022: the First Audio Deep Synthesis Detection Challenge," in *Proc. ICASSP*. IEEE, 2022.

[3] W. H. Kang, J. Alam, and A. Fathan, "CRIM's System Description for the ASVSpoof2021 Challenge," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 100–106.

[4] W. H. Kang, J. Alam, and A. Fathan, "Investigation on activation functions for robust end-to-end spoofing attack detection system," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 83–88.

[5] A. Tomilov, A. Svishchev, M. Volkova, A. Chirkovskiy, A. Kondratev, and G. Lavrentyeva, "STC antispoofing systems for the ASVspoof2021 challenge," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 61–67.

[6] J. Alam, P. Kenny, V. Gupta, and T. Stafylakis, "Spoofing Detection on the ASVspoof2015 Challenge Corpus Employing Deep Neural Networks," in *Proc. Odyssey 2016 Workshop*, 2016, p. 270–276.

[7] M. Sahidullah, T. Kinnunen, and Hanilc¸i C., "A comparison of features for synthetic speech detection," in *Proc. Interspeech*, 2015, p. 2087–2091.

[8] J. Alam, Bhattacharya G., and Kenny. P., "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," in *Proc. Odyssey 2018 Workshop*, 2018, pp. 393–398.

[9] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Proc Odyssey 2016 Workshop*, 2016, p. 283–290.

[10] J. Yang, R. K. Das, and H. Li, "Extended constant-Q cepstral coefficients for detection of spoofing attacks," in *Proc. APSIPA ASC*, 2018, p. 1024–1029.

[11] J. Yang, R. K. Das, and N. Zhou, "Extraction of octave spectra information for spoofing attack detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, pp. 2373–2384, 2019.

[12] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech decetion," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2160–2170, 2020.

[13] R. K. Das, J. Yang, and H. Li, "Long range acoustic features for spoofed speech detection," in *Proc. Interspeech*, 2019, p. 1058–1062.

[14] J. Yang and R. K. Das, "Long-term high frequency features for synthetic speech detection," *Digital Signal Processing*, vol. 97, pp. 102622, 2020.

[15] R. K. Das, J. Yang, and H. Li, "Long range acoustic and deep features perspective on ASVspoof 2019," in *Proc. ASRU Workshop*, 2019, p. 1018–1025.

[16] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. ICASSP*, 2021, pp. 6369–6373.

[17] H. Tak, J. W. Jung, J. Patino, M. Kamble, M. Todisco, and N Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 1–8.

[18] W. Ge, J. Patino, M. Todisco, and N. Evans, "Raw differentiable architecture search for speech deepfake and spoofing detection," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 22–28.

[19] R. K. Das, "Known-unknown data augmentation strategies for detection of logical access, physical access and speech deepfake attacks: ASVspoof 2021," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 29–36.

[20] T. Chen, E. Khoury, K. Phatak, and G. Sivaraman, "Pindrop Labs' Submission to the ASVspoof 2021 Challenge," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 89–93.

[21] X. Chen, Y. Zhang, G. Zhu, and Z. Duan, "UR channel-robust synthetic speech detection system for ASVspoof 2021," in *Proc. ASVspoof 2021 Workshop*, 2021, pp. 75–82.

[22] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," in *Proc. Interspeech*, 2021.

[23] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proc. ICASSP*, 2018, p. 5329–5333.

[24] N. An, N. Thanh, and Y. Liu, "Deep CNNs with Self-Attention for Speaker Identification," *IEEE Access*, vol. 7, pp. 85327–85337, 2019.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, p. 7132–7141.

[26] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv*, vol. abs/2006.11477, 2020.

[27] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P.V. Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," *arXiv*, vol. abs/2111.09296, 2021.

[28] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv*, vol. abs/2110.13900, 2021.

[29] S.W. Yang, P.H. Chi, Y.S. Chuang, C.J. Lai, K. Lakhotia, Y.Y. Lin, A.T. Liu, J. Shi, X. Chang, G.T. Lin, T.H. Huang, W.C. Tseng, K.T. Lee, D.R. Liu, Z.L. Huang, S.Y. Dong, S.W. Li, S. Watanabe, A. Mohamed, and H.Y. Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021, pp. 1194–1198.

[30] H. Yu, J. Zhao, S. Yang, Z. Wu, Y. Nie, and W.Q. Zhang, "Language Recognition Based on Unsupervised Pretrained Models," in *Proc. Interspeech*, 2021, pp. 3271–3275.

[31] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv*, vol. abs/2005.07143, 2020.