2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge
16 September 2021, Online

STC Antispoofing Systems for the ASVspoof2021 Challenge

Anton Tomilov, Aleksei Svishchev, Marina Volkova,
Artem Chirkovskiy, Alexander Kondratev, Galina Lavrentyeva

STC-innovations Ltd., Russia
{tomilov, svishchev, volkova, chirkovskiy, kondratev, lavrentyeva}@speechpro.com

Abstract

This paper describes Speech Technology Center (STC) antispoofing systems submitted to the ASVspoof 2021 challenge in three tracks: logical access (LA), physical access (PA) and new deep-fake (DF) tracks. Proposed solutions in all three tracks were a weighted score-level fusion of several deep neural network models, namely ResNet18, LCNN9, RawNet2, and their modifications. As input features, we used various frontends, including Mel-scaled short-time Fourier transform of an audio signal, linear or rectangular frequency filter banks, and trainable transforms such as LEAF or SinConv. This paper mainly focuses on several approaches that were used to raise generalizing ability of our systems. Augmentation with emulation of codecs frequency distortions based on FIR filters significantly improved results in LA and DF tracks. The problem of overfitting in all tracks was partly solved by applying the mixup technique. To solve out-of-domain data problems and adapt to real replay attacks in the PA track, microphone and room impulse responses augmentation was used. Applied augmentation techniques allowed us to significantly increase the quality and robustness of the proposed spoofing detection systems in all tracks. That is confirmed by the EER values on progress and eval sets: our best system achieved an EER of 1.32 % on the evaluation part of the Challenge corpora in the LA track.

1. Introduction

Malicious attack countermeasures (CM) increase their importance being pushed by fast development of commercial automatic speaker verification (ASV) systems. Moving towards simplification of verification procedures for users lead to high quality requirements for both CM and ASV systems. This increases the relevance of ASVspoof initiative and its challenges, that are aimed to support the development of spoofing detection methods for ASV systems.

In this year ASVspoof 2021 Challenge [1] was represented by three tracks: logical access (LA), physical access (PA) similar to ASVspoof 2019 but in telephone channel, and new deep fake (DF) track. All three tracks contain only new evaluation data with bona fide and spoofed utterances. LA attacks includes utterances created by text-to-speech (TTS), voice conversion (VC) or hybrid algorithms. Spoofed utterances in PA track contain both real and simulated results of replaying and recording bona fide utterances using a variety of electronic devices and acoustic environments. In contrast to previous challenges LA track was focused on the development of spoofing countermeasures that are robust to codec and transmission channel variability. Accordingly, all the evaluation data in these tracks were transmitted across either a public switched telephone network (PSTN) or a voice over Internet Protocol (VoIP) network. DF attacks are almost the same as in LA track but DF utterances were processed using different (not telephone) audio codecs for audio data compression-decompression process.

According to the evaluation plan [1] spoofing countermeasure systems were trained and optimised using ASVspoof 2019 training and development data only. This was a serious challenge for all three tracks because of a domain mismatch between training and evaluation data. Our previous experience in spoofing detection in telephone channel confirm high importance of careful in-domain channel emulation [2]. And one of the objectives declared in the Challenge was to evaluate impacts of data augmentation to replace in-domain training and development data.

We considered several augmentation techniques with two goals: increase the versatility of the training dataset to prevent overfitting and improve the robustness of the final system to out-of-domain data. These were additional noise augmentation, acoustic codecs, high and low pass filters, using time-warping and SpecAug [3], mixup augmentation [4] and impulse response filters.

Analysis of ASVspoof 2019 systems demonstrates that current spoofing detection systems are based on deep neural networks (DNN) [5]. While the best performance is obtained by the score level fusion of several single systems, which varied by training data, feature extractions procedures, architectures or training policy (loss functions, learning rate schedulers, optimizers, etc.).

Most of ASVspoof 2019 systems used spectral features based on short-time Fourier transform (STFT) [6], but today there are several papers with different approaches such as Gabor transform with trainable parameters (LEAF [7] for instance) or even raw signal itself (as in RawNet2 [8]). In our systems we focus on both: traditional spectral features and novel approaches. Nowadays commonly spread architectures for ASV countermeasure systems are LCNN [9] and RawNet [8]. To construct our systems we used LCNN and RawNet proposed in baseline solution. In the case of LCNN we modified architecture to include some features from last years solution [9]. We also included a modified ResNet [10] architecture.

The rest of the paper is organized as follows. Section 2 describes our systems for all the tracks independently and their setups. Section 3 contains detailed description of the augmentation techniques we used to deal with overfitting and mismatch conditions in train and eval sets. Section 4 illustrates the results obtained by the proposed systems and discusses the impact of different augmentation techniques on the final systems performance.

10.21437/ASVSPOOF.2021-10

2. Proposed systems overview

2.1. Features

Cepstral features such as Mel frequency cepstral coefficients and constant Q transform cepstral coefficients (CQCCs) are now considered as the standard feature extraction techniques in spoofing detection [6]. They have already shown best detection performance, especially for unknown attacks in previous ASVspoof Challenges. At the same time top 5 single systems in PA track in ASVspoof2019 used spectral features rather than cepstral [6]. Taking into account the results and our own experience in spoofing detection we investigated both approaches and also considered novel approaches.

Most part of the single systems in all tracks were based on STFT transform with hop (h) 128, window length (n) were varied from 384 till 768 to gain diversity and increase generalization ability. Convolutional implementation of Fourier transform allows to adopt easily to varying window length. Number of bins (b) in Mel-scale transform (MSTFT or mel spectrogram) were varied from 20 to 80. Higher number of bins show better performance especially for LA and DF tracks. We also experimented with alternative features: CQT, MFCC and LEAF transforms [7], as well as applying linear, rectangular or inverse Mel-scale transforms after STFT. Single systems based on MSTFT show better performance in all tracks. Almost all single systems included in final fusion for LA and DF use MSTFT features with slight difference in extraction parameters (see Tables 1 and 3). On the other hand in PA track the extension of feature maps by LEAF and LSTFT proved to be reasonable (see Table 2). Results of individual systems on progress set for fusions (Tables 1, 3 and 2) can be found in Tables 9, 10 11 in Section 4.

RawNet2 uses raw signal as an input applying SincConv1d transformation with hop equals to 1, number of channels 20 and length of kernels 512.

All the models were trained on full train part of the corresponding track until convergence on dev dataset by default.

2.2. LA system

2.2.1. LA data preparation and feature extraction

During our experiments for LA track we discovered that elimination of the normalization step of input features improves the performance of the spoofing detection systems. Additionally we explored the question of better utterance length. We found out that training on full length utterances leads to better system quality. However systems trained on fixed length fragments can be effectively used in fusion as additional systems. Due to this we used 6-sec fixed fragments as well.

As an input features most single systems (except of RawNet2) used MSTFT features described above (see also Table 1).

Mixups on feature level and FIR filters were used as augmentation (the detailed description in section 3). We found it very helpful during training of all the models.

2.2.2. LA single models

For deep embedding extraction in LA track we explored most common DNN architectures and its modifications: LCNN9, ResNet18 and RawNet2 [8]. ResNet18 architecture was modified in a way to lighten it: we used only one layer in every block and limited the amount of channels in input and output to 32. Resulting ResNet model has similar number of parameters to LCNN9. Initial baseline RawNet2 used GRU output as final embedding, we used all LSTM hidden states instead. LCNN9 was modified in a way of combining some parts of baseline [11] and our last [9] solutions.

Table 1: *Final submission system for Logical Access subchallenge*

Feature	Model	Weight
MSTFT (b=80, n=512)	LCNN	0.162
MSTFT (b=60, n=512)	ResNet	0.162
MSTFT (b=60, n=256)	LCNN	0.081
MSTFT (b=60, n=512)	LCNN	0.081
MSTFT (b=60, n=512)	LCNN	0.162
MSTFT (b=60, n=512)	LCNN	0.162
SincConv(b=20, n=512)	RawNet	0.09
MSTFT (b=60, n=512)	LCNN	0.02
MSTFT (b=80, n=384)	LCNN	0.02
MSTFT (b=80, n=768)	LCNN	0.02
MSTFT (b=80, n=512)	LCNN	0.02
MSTFT (b=80, n=512)	LCNN	0.02

During training we used Adam optimizer with initial $lr = 0.0003$ for LCNN, $lr = 0.001$ for ResNet, $lr = 0.0001$ for RawNet2. As a scheduler stepLR was used with step size of 10 epochs (20 for ResNet) and coefficient 0.5. Batch size was 128. Almost in all the experiments we used Center Loss which penalizes intra class embedding distance [12].

Embedding for each system were extracted from the convolutional layers. Then in all the cases we aggregate them as a sequence applying biLSTM and then averaging sum of embeddings and hidden states of biLSTM as it proposed in LFCC-LCNN baseline [11]. Then final fully connected layer was used to project 128 dimensional vectors onto 2 dimensions. During experiments we used multihead attention, GATs [13] and Attentive Statistic Pooling (ASP) [14] blocks to aggregate embeddings but they show worse performance.

2.2.3. LA fusion

Our final LA system combines 12 single subsystems: 10 LCNN and one ResNet models that use mel-scaled STFT (MSTFT) as input features and one RawNet2 model that operated with raw signal.

The details of final system is demonstrated in Table 1. Fusion was performed on the score level. Fusion weights were selected manually with respect to the performance of each single system on the progress part of evaluation set.

2.3. PA system

2.3.1. PA data preparation and feature extraction

During training all PA models, we used chunks with fixed 1 second length that were randomly sampled from full-length utterances. During evaluation step we processed the whole utterance using 1 sec window and 0.5 sec step. The final score was estimated as the average mean of all scores for the utterance.

Similar to the LA track MSTFT features show better performance for PA scenario. Adding LEAF [7] to MSTFT as a second channel of the feature map led to quality improvement as well. Further addition of features as new channels to feature map generally didn't perform well except for the case LEAF/MSTFT/LSTFT which also improved the fusion result. For details see Table 2.

All models were trained with an addition MUSAN Noise

augmentation [15]. To increase performance of a single model we also used mixups [4] on feature maps and augmentation with RIR, IR and MIR [16] discussed in section 3.

Table 2: *Final submission system for Physical Access sub-challenge*

Feature	Augs	Weight
LEAF, MSTFT (h=128, m=40, n=1024)	IR	0.3
LEAF, MSTFT (h=128, m=40, n=1024)	MIR	0.2
LEAF, MSTFT (h=256, m=40, n=1024)	RIR	0.15
LEAF (h=128, m=40, n=2048)	IR	0.1
LEAF, MSTFT, LSTFT (h=256, m=40, n=1024)	RIR	0.1
LEAF, MSTFT (h=128, m=40, n=1024)	IR + RIR	0.1
LEAF, MSTFT (h=256, m=60, n=1024)	IR	0.05

2.3.2. PA single models

All the single models trained for this track were based of vanilla ResNet18 [10]. We also tried to use LCNN and RawNet but they showed low generalization ability and a tendency to overfitting.

Adam with $lr = 0.001$ was used as optimizer (batch size 128). As a scheduler inverted square with linear warm-up during 4 epochs showed the best performance.

We use ResNet convolution layers to construct 128-dimensional embeddings sequence from time-spectral features. Similar to LA systems we analyzed different aggregation techniques to get single vector from sequence of embeddings. But in contrast to LA, the best performance in PA track was achieved by using ASP [14].

2.3.3. PA fusion

To construct final PA system we used 7 single systems based on ResNet18 models with LEAF, MSTFT or linear scaled STFT (LSTFT) extracted with different parameters (Table 2). Fusion weights were selected manually as in LA track.

2.4. DF system

2.4.1. DF data preparation and feature extraction

The train part of LA dataset was used during models training. Features are mostly similar to LA case except of the parameters described in Table 3. Training datasets and augmentation techniques in this case vary depending on the system. For some models we included train data, augmented by SoX and FFMPEG utilities to emulate several well-known compression formats.

2.4.2. DF single models

LCNN9, ResNet18 and RawNet2 like models used in DF track were very similar to LA track ones.

Training conditions for LCNN9 were almost the same as in LA track. For training ResNet models we chose higher learning rates up to 0.001 with batch size 128. All the RawNets and some ResNets were trained using train data set with appliance of different codecs (sbc, mp3, aac, opus, vorbis). During RawNet train we used special kinds of mixups on raw signal which will be described in section 3.

Table 3: *Final submission system for Deep Fake sub-challenge*

Model	Feature	Weight
ResNet18	MSTFT (b=60, n=512)	0.315
	MSTFT (b=60, n=512)	0.045
	MSTFT (b=60, n=512)	0.045
	MSTFT (b=60, n=512)	0.027
	MSTFT (b=60, n=512)	0.018
RawNet2	SincConv (b=20, n=512)	0.035
	SincConv (b=20, n=512)	0.125
	SincConv (b=20, n=512)	0.075
	SincConv (b=20, n=512)	0.015
LCNN	MSTFT (b=80, n=512)	0.06
	MSTFT (b=60, n=512)	0.03
	MSTFT (b=60, n=256)	0.06
	MSTFT (b=60, n=512)	0.06
	MSTFT (b=60, n=512)	0.09

2.4.3. DF fusion

DF final system was our biggest fusion. It consists of 14 single subsystems which can be divided into three parts according to the architecture they are based on (Table 3). These are 5 Light ResNet18-like, 5 LCNN-like models and 4 RawNet2-like models. All of the models have similar aggregation policy using averaging of embeddings weighted with hidden states of biLSTM. The fusion is performed similar to LA system.

3. Augmentation techniques

3.1. Emulation of codecs by FIR filters

Channel emulation by using codecs and signal compression approaches is a common practice for training DNN based systems in signal processing fields, such as speaker and speech recognition [17, 18, 19]. At the same time we found out that not all regular telephone channel emulation match the real spoofing attacks scenario [2] and augmentation techniques should be carefully selected.

According to [20] all codecs may be roughly divided into four parts: narrowband (NB), wideband (WB), ultra wideband (UWB) and fullband codecs (see Table 4).

Table 4: *Frequency limits of different codecs.*

Name	Low freq. limit	High freq. limit
NarrowBand (NB)	300 Hz	3.4 kHz
WideBand (WB)	100 Hz	7 kHz
Super WideBand (SWB)	50 Hz	14 kHz
FullBand (FB)	20 Hz	20 kHz

Since the sampling rate of the train/dev and eval sets is 16 kHz [1], we can eliminate the Nyquist frequency by 8 kHz, and consequently concern only NB and WB codecs due to the high frequency limit. At low frequency side we can concern all variety of codecs.

On the one hand, only a small amount of codecs are publicly available and on the other hand, we discovered that testing

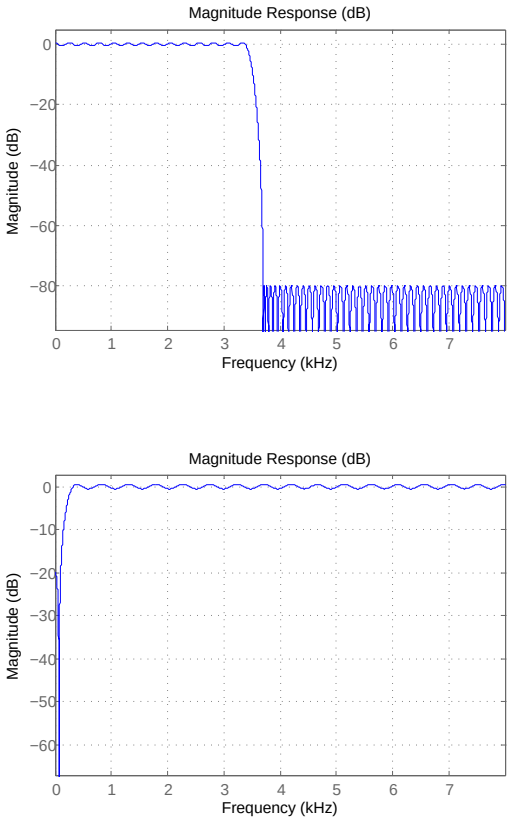

Figure 1: *Examples of FIR filter magnitude responses. Top: low pass filter (LPF) emulates narrowband codecs. Bottom: high pass filter (HPF) emulates wideband codecs [20].*

of each available codec implementation can be extremely time-consuming. Considering that we focused on the online augmentation approaches applied on the fly during the training process, which can be implemented in multi-threaded mode.

Our solution was to emulate magnitude responses of codecs by raw signal convolution with randomly selected low or high frequency filter kernels.

Used in our research finite impulse response (FIR) filters were constructed with an equiripple method by Matlab Filter Design and Analysis Tool. F_s was 16000 Hz. Pass frequencies (F_{pass}) were chosen as low frequency limits for low pass filters (LPF) and as high frequency limits for high pass filters (HPF) (see Table 4). Stop frequencies F_{stop} were randomly selected from $1.05 * F_{pass}$ to $1.2 * F_{pass}$ for low pass filters. For high pass filters F_{stop} values were chosen in range from $0.5 * F_{pass}$ till $0.8 * F_{pass}$. Magnitude responses of LPF and HPF are shown in Figure 1.

Described filters allow to decrease the impact of spectral features in range 0-300 Hz and 3.4-8 kHz without rejecting them at all. Thus one of important hyperparameter in this case was a probability of applying LPF or HPF. It should be noted that using convolutions with FIR filter kernels is similar to SpecAug frequency masking techniques [3] with two restrictions. The first one is to mask only low and high frequency bins instead of randomly selected ones. The second restriction is to use as a mask a signal itself multiplied by a random small value,

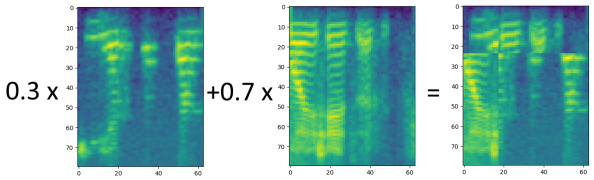

Figure 2: *Example of applying Mixed Concat mixup [4] to logarithm of Mel-scaled STFT magnitude.*

instead of its local mean, to imitate the magnitude of spectral power suppression.

3.2. Mixups

Mixup is a technology of data augmentation which helps to improve generalization and robustness of the model via smooth interpolation between labels as well as between signals [21]. In the case of the audio signals, mixups can be applied in two ways: on the raw signal level and on the features level. In the first case, we typically use a concatenation of two parts of signals. In the second case, we use the same variety of mixups as in [21] applied to time-spectral maps. The particular case of applying Mixed Concat mixup to spectrogram is depicted in Figure 2.

In both cases training samples are obtained by formula 1.

$$S = \lambda \times S_1 + (1 - \lambda) \times S_2 \tag{1}$$

where S_1 is initial signal, S_2 is another signal randomly chosen from the train dataset and λ is a constant laying in range from 0 till 1. Commonly, λ is sampled from the beta distribution with $\beta(\alpha, \alpha)$. Both parameters of the distribution are equal to each other to have a mean of such a distribution equals to $1/2$.

Using the mixup we should also modify a loss (L) calculation procedure in a following way:

$$L = \lambda \times loss(T_1) + (1 - \lambda) \times loss(T_2) \tag{2}$$

where T_1 and T_2 are labels of signals S_1 and S_2, consequently.

Experiments with mixups reveals that choosing the right rate of applying mixup is crucial.

3.3. RIRs and Microphone IRs

During experiments for PA track we discovered the efficiency of using room impulse responses (RIR) or microphone impulse responses (MicIR) for training data augmentation. There were two databases with RIRs:

- MIRaGe [16]
- Data set of real RIRs which is composed of three databases: the RWCP sound scene database, the 2014 REVERB challenge database and the Aachen impulse response database (AIR). Overall there are 325 real RIRs.

Microphone IR database was collected from different sources by ourselves. It contains IRs of different microphones, phones and other electronic devices. In addition, for some models, a performance improvement was achieved with the help of additional noise augmentation, taken from the MUSAN database [15]. We randomly sampled audio noise fragments, re-normalized them to random SNR from range 5-15 dB, and

added them to the initial raw signal. The impact of specific augmentation techniques can be seen in Table 7.

4. Results and Discussion

In order to show the effectiveness of the proposed augmentation techniques and facilitate reproducibility of results we applied them to LFCC-LCNN baseline system calculated on post-eval LA Dataset (Table 5). Code is available via link [1]. We removed 8 kHz restriction in LFCC for FIR and Mixup cases (because FIRs were constructed for 16 kHz). In the LFCC-LCNN baseline case removing that restriction leads to high EER. As it was shown, all augmentations improve results drastically (Tables 6 and 7). To provide results that were closer to those we used in final model fusions (Table 12) we change the baseline LFCC-LCNN model to our MSTFT-LCNN model which is a combination of baseline [11] and our LCNN model from the last competition [9]. We set $\alpha = 0.6 - 0.8$ during mixup experiments (for details see 3.2).

Table 5: *Impact of augmentation techniques applied to the challenge baseline (LA post-eval data set).*

Name	min t-DCF	EER
LFCC-LCNN	0.3445	9.26
LFCC-LCNN + FIR	0.3119	6.22
LFCC-LCNN + FIR + Mixup	0.3082	5.63

As it is shown in Table 6 mixup didn't improve result by itself in eval part but it gains a positive effect of using FIR filters. RS stands for Random Square mixup, RRI - for Random Row Intervals, MC - Mixed Contat mixup [4]. Bare means that the models were trained on LA train dataset without any data augmentations. Moreover, we noticed that models trained without mixup prone to overfitting. This fact makes it hard to choose right checkpoint without good validation set. Almost all the mixup techniques [4] improve performance of model in LA and DF tracks (Table 6), but we noticed that Random Square mixup lead to the best performance during the competition.

Table 6: *Influence of augmentation techniques on MSTFT-LCNN model performance in LA and DF tracks (post-eval datasets).*

Model content	EER (LA)	EER (DF)
Bare	3.32	21.9
RS Mixup	8.5	19.14
FIR	3.1	16.96
RS Mixup and FIR	2.21	20.16
RRI Mixup and FIR	2.88	18.63
MC Mixup and FIR	2.83	19.83
BC+ Mixup and FIR	2.53	-

Investigating the influence of augmentation techniques on PA track model performance, we revealed that baseline single model and model trained with MUSAN's Noise augmentation have a similar equal error rate (Table 7). As expected, the room impulse response augmentation had the most significant effect on model performance. MirAGE RIR augmentation [16] outperformed all other methods. It may be related to the fact that MirAGE dataset contains RIRs which was obtained from measurements on dense mesh of transmitters and receivers. Authors of paper [16] investigated nonsmooth changes in kernels with small varying of receiver and transmitter positions.

Further combination of augmentation techniques leads to minor improvements of both EER and min t-DCF (Table 7). The best result was obtained by using Noise from MUSAN, RIRs and MirAGE. Models with similar performance were included into final PA fusion (Table 12).

Table 7: *Influence of augmentation techniques on MSTFT-ResNet18 model in PA track (post-eval PA dataset).*

Model content	min t-DCF	EER
Bare	0.83	30.9
Noise [15]	0.84	31.2
RIR	0.81	30.3
MirAGE [16]	0.78	28.4
Noise + MirAGE	0.77	27.6
Noise + IR	0.84	30.2
Noise + RIR	0.79	29.1
RIR + MirAGE	0.78	28
Noise + RIR + MirAGE	0.76	27.5

We carried on short study of FIR filters impact on LA track (Table 8). It was shown that filters that emulate narrowband (NB) codecs have a stronger influence on the system performance comparing to wideband (WB) counterparts. For the NB situation, suppression of low frequencies by applying a high pass filter has a greater impact on EER than low pass filters (Figure 1). Contrarily, using LPF in WB case is more crucial than HPF. Such a behavior can be explained by the fact that TTS and VC systems have significant features in high-frequency region [22] which were taught during the train part and, on the other hand, maybe hidden by NB codecs in the eval part.

Table 8: *Impact of different FIR (Table 4) augmentation on the MSTFT-LCNN single systems in LA and DF tracks (trained with mixup, evaluated on post-eval dataset).*

Model content	EER(LA)	EER(DF)
Bare	3.32	21.9
NB LPF [15]	2.35	20.7
NB HPF [15]	2.22	20.9
WB LPF [15]	2.59	21.8
WB HPF [15]	2.83	16.7

During experiments it was revealed, that using voice ativity detectors (VAD) provides no improvement in LA and DF tracks, and lowers quality in PA track.

Our research devoted to an evaluation of the effectiveness of augmentation techniques made a significant contribution to the development of single systems resistant to channel distortion in all tracks. The performance evaluation on the private eval set of all single systems used further for final fusion are demonstrated in Tables 9, 10 11 (dash means that performance of a model was not measured on the progress eval data). As it seen from Tables 9 and 10, the single models error rates on progress part of eval are similar to the final system performances scored on post-eval data (Table 12). Contrarily, the DF single model EER tends to be very low on the progress eval data set (Table 11), but the system performs significantly worse on the post-eval (Table 12).

[1] https://github.com/MacJieHokNSU/2021

Also, all PA models show low performances. Most probably, it is because of using more comprehensive environmental and attacker factors in the Competition and the fact that the PA track eval set contains predominantly real replayed speech, in addition to a smaller proportion of simulated replayed speech. It was also discovered that LEAF [7] gives an improvement only in PA track. Contrarily, using it in LA or DF tracks leads to slightly worse results and therefore not published.

Table 9: *Final single systems performance for Logical Access sub-challenge (on progress eval data)*

Feature	Model	EER
MSTFT (b=80, n=512)	LCNN	1.48
MSTFT (b=60, n=512)	ResNet	1.91
MSTFT (b=60, n=256)	LCNN	-
MSTFT (b=60, n=512)	LCNN	1.97
MSTFT (b=60, n=512)	LCNN	1.97
MSTFT (b=60, n=512)	LCNN	1.49
SincConv(b=20, n=512)	RawNet	-
MSTFT (b=60, n=512)	LCNN	-
MSTFT (b=80, n=384)	LCNN	-
MSTFT (b=80, n=768)	LCNN	-
MSTFT (b=80, n=512)	LCNN	-
MSTFT (b=80, n=512)	LCNN	-

Table 10: *Final single systems performance for Physical Access sub-challenge (on progress eval data)*

Feature	Augs	EER	min t-DCF
LEAF, MSTFT (h=128, m=40, n=1024)	IR	26.00	0.7469
LEAF, MSTFT (h=128, m=40, n=1024)	MIR	26.42	0.7504
LEAF, MSTFT (h=256, m=40, n=1024)	RIR	27.72	0.767
LEAF (h=128, m=40, n=2048)	IR	28.18	0.7775
LEAF, MSTFT, LSTFT (h=256, m=40, n=1024)	RIR	28.55	0.794
LEAF, MSTFT (h=128, m=40, n=1024)	IR + RIR	27.43	0.7643
LEAF, MSTFT (h=256, m=60, n=1024)	IR	28.36	0.7839

The best results obtained for each track on the private evaluation part of the Challenge corpora of our systems are shown in Table 12.

5. Conclusion

Current paper contains the overview of the single and final fusion systems submitted in all three tracks of the ASVspoof2021. LA solution seems more robust to condition changes than other. PA track contains utterances hard to deal with. Possibly they were recorded and re-recorded in pretty similar conditions with using high quality devices. It may explain a low quality of the final system on the private eval set.

DF track solution seems to be overfitted on the progress eval data set. Obviously that private part of DF eval contains some unseen codecs and conditions.

Table 11: *Final single systems performance for Deep Fake sub-challenge (on progress eval data)*

Model	Feature	EER
ResNet18	MSTFT (b=60, n=512)	1.04
	MSTFT (b=60, n=512)	-
	MSTFT (b=60, n=512)	-
	MSTFT (b=60, n=512)	-
	MSTFT (b=60, n=512)	-
RawNet2	SincConv (b=20, n=512)	1.06
	SincConv (b=20, n=512)	1.63
	SincConv (b=20, n=512)	1.5
	SincConv (b=20, n=512)	2.03
LCNN	MSTFT (b=80, n=512)	1.58
	MSTFT (b=60, n=512)	1.46
	MSTFT (b=60, n=256)	1.13
	MSTFT (b=60, n=512)	-
	MSTFT (b=60, n=512)	0.89

Table 12: *General system performance (fusion of models on post eval data).*

Name	EER
LA system (Tab. 1)	1.32
PA system (Tab. 2)	26.24
DF system (Tab. 3)	15.64

In this work, we also paid special attention to the study of the effectiveness of using various augmentations techniques for training robust systems in all three tracks. Significant gain to final result was given by using:

- mixup techniques to prevent overfitting in all three tracks;
- FIR filters as emulation of general codecs magnitude response in LA and DF tracks;
- together the mixup and FIR always give better results than by one;
- RIR, IR and noise augmentation in PA track.

6. References

[1] "ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," `https://www.asvspoof.org/asvspoof2021/asvspoof2021_evaluation_plan.pdf`.

[2] Galina Lavrentyeva, Sergey Novoselov, Marina Volkova, Yu Matveev, and Maria De Marsico, "Phonespoof: A new dataset for spoofing attack detection in telephone channel," in *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 05 2019, pp. 2572–2576.

[3] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019.

[4] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[5] Tomi Kinnunen, Héctor Delgado, Nicholas Evans, Kong-Aik Lee, Ville Vestman, Andreas Nautsch, Massimiliano Todisco, Xin Wang, Md Sahidullah, Junichi Yamagishi, and Douglas A Reynolds, "Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification: Fundamentals," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 2195 – 2210, July 2020.

[6] Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi H. Kinnunen, Ville Vestman, Massimiliano Todisco, Hector Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee, "Asvspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, Apr 2021.

[7] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi, "LEAF: A learnable frontend for audio classification," *CoRR*, vol. abs/2101.08596, 2021.

[8] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher, "End-to-end anti-spoofing with RawNet2," in *ICASSP*, Toronto, Canada, June 2021.

[9] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 1033–1037.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] Xin Wang and Junich Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," *Interspeech 2021*, 9 2021.

[12] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *Lecture Notes in Computer Science book series*, 10 2016, vol. 9911, pp. 499–515.

[13] Jee-Weon Jung, Hee-Soo Heo, Ha-Jin Yu, and Joon Son Chung, "Graph attention networks for speaker verification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 06 2021, pp. 6149–6153.

[14] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," *Interspeech 2018*, Sep 2018.

[15] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," 2015, arXiv:1510.08484v1.

[16] Jaroslav Cmejla, Tomás Kounovský, Sharon Gannot, Zbynek Koldovský, and Pinchas Tandeitnik, "Mirage: Multichannel database of room impulse responses measured on high-resolution cube-shaped grid," in *28th European Signal Processing Conference, EUSIPCO 2020, Amsterdam, Netherlands, January 18-21, 2021*. 2020, pp. 56–60, IEEE.

[17] Hossein Zeinali, Luka Burget, Johan Rohdin, Themos Stafylakis, and Jan Cernocky, "How to improve your speaker embeddings extractor in generic toolkits," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 05 2019, pp. 6141–6145.

[18] Daniel Garcia-Romero, Greg Sell, and Alan Mccree, "MagNetO: X-vector Magnitude Estimation Network plus Offset for Improved Speaker Recognition," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 1–8.

[19] Thi-Ly Vu, Zhiping Zeng, Haihua Xu, and Eng-Siong Chng, "Audio codec simulation based data augmentation for telephony speech recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 198–203.

[20] ETSI TR 102 949 V1.1.1 (2014-09), "Speech and multimedia transmission quality (STQ); wideband and super-wideband speech terminals; perceptually motivated parameters," .

[21] C. Summers and M. J. Dinneen, "Improved mixed-example data augmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Los Alamitos, CA, USA, jan 2019, pp. 1262–1270, IEEE Computer Society.

[22] Dipjyoti Paul, Monisankha Pal, and Goutam Saha, "Novel speech features for improved detection of spoofing attacks," in *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1–6.