The Biometric Vox System for the ASVspoof 2021 Challenge

Joaquín Cáceres, Roberto Font, Teresa Grau, Javier Molina

Biometric Vox S.L.

name.surname@biometricvox.com

Abstract

This paper describes the systems developed by Biometric Vox for the ASVspoof 2021 challenge Logical Access (LA) and Physical Access (PA) tracks. The Logical Access track aims at detecting the use of speech synthesis or voice conversion techniques. In the case of the Physical Access track, the task is the detection of replayed speech. We experiment with different input features and neural network architectures. In particular, we propose a lightweight Time Delay Neural Network architecture and the use of Focal Loss as a way to handle class imbalance and emphasize hard-to-classify samples. Additionally, we explore the use of neural networks as embedding extractors and propose a one-class Gaussian classifier on top of these embeddings. Our final system for the PA track obtains min-tDCF=0.6658 and EER=24.44% on the progress set and min-tDCF=0.7462 and EER=29.00% on the evaluation set. On the LA track, our best system obtains min-tDCF=0.2371 and EER=4.54% on the progress set and min-tDCF=0.2747 and EER=5.58% on the evaluation set.

1. Introduction

In the last decades, Automatic Speaker Verification (ASV) has reached the degree of maturity required to see widespread adoption in many applications like home banking, smart assistants or forensic analysis, to name just a few. With this growing adoption, oftentimes in high-stakes scenarios, comes the need of strong countermeasures against any kind of malicious attack. This, together with the recent advances in Deep Learning, that make possible the creation of sophisticated attacks, makes the development of reliable countermeasures against these attacks an important area of research. Although, compared with the maturity of the ASV field, the research on ASV spoofing countermeasures is on a more embryonic state, significant progress has been made in the last years.

The ongoing automatic speaker verification spoofing and countermeasures (ASVspoof) challenge series has been a major force driving this progress by promoting research in this area and providing the community with a common ground in terms of data, protocols and metrics. ASVspoof 2021 [1, 2], the fourth edition in this series, comprises three different tracks:

- A Logical Access (LA) track similar to those of ASVspoof 2015 and 2019, where the aim is to develop countermeasures capable of discriminating between real *bona-fide* speech and speech samples generated using Speech Synthesis and Voice Conversion techniques. New to the 2021 edition is the presence of transmission channel effects on test data: different codecs, bandwidths, etc.
- A Physical Access (PA) track where the task is to develop countermeasures capable of discriminating between real *bona-fide* speech and speech replayed using some kind of replay device. The particularity, in this case, is that training data is simulated while evaluation data is real, thus providing participants with a challenging data mismatch scenario.
- A Speech Deepfake (DF) track including compressed audio similar to the LA task but not involving a speaker verification system.

This paper describes the systems we developed for the ASVspoof 2021 challenge, in particular for the Physical Access (PA) and Logical Access (LA) tracks. We explore the use of different neural network architectures and data augmentation strategies and propose the use of Focal Loss as the objective function and the linear fusion of classifiers with different input features.

The rest of the paper is organized as follows: Section 2 presents the details about ASVspoof 2021 challenge and data, Section 3 details or approach to data augmentation, and Section 4 describes the models we developed for both tasks. Experimental results are presented in Section 5, and, finally, we summarize our conclusions in Section 6.

2. Task Description and Data

The ASVspoof 2021 challenge continues the lines set by ASVspoof 2019 [3] but with a shift towards more practical scenarios. For system development, no new training or development data was distributed. Instead, systems should be built exclusively by using ASVspoof 2019 *training* and *development* partitions. The use of any other speech data was not allowed. The contents of the ASVspoof 2019 database are summarized in Table 1.

Table 1: Summary of ASVspoof 2019 database contents.

Subset	# speakers		# utterances			
	Male	Female	Logical access		Physical access	
			Bona fide	Spoof	Bona fide	Spoof
Training	8	12	2,580	22,800	5,400	48,600
Development	8	12	2,548	22,296	5,400	24,300
Evaluation	21	27	13,049	113,282	18,088	24,300

For the Physical Access task, evaluation data consisted in $943,110$ utterances distributed as 16 kHz, 16 bits per sample FLAC files. While systems should be built using 2019 data, which contains simulated data, evaluation data contained mostly real replayed speech, with a smaller proportion of simulated replayed speech. The task was therefore to develop robust countermeasures with the ability to perform reliably in this data mismatch scenario.

Evaluation data for the Logical Access task consisted in $181,566$ utterances distributed is the same format as in the PA case. The evaluation set contained attacks generated using

10.21437/ASVSPOOF.2021-11

the same text-to-speech (TTS), voice conversion (VC) or hybrid (voice conversion systems fed with synthetic speech) algorithms present in the ASVspoof 2019 LA evaluation partition. However, samples were transmitted through either the public switched telephone network (PSTN) or a voice over Internet Protocol (VoIP) network thus exhibiting real-world codec and transmission channel effects. The aim, in this case, was to develop countermeasures robust to these variations on input data.

The challenge itself ran in two phases. During the *progress* phase, each team was allowed to make up to 3 submissions per day. Results were computed from a subset of the evaluation data and were available to all participants through a leaderboard. In the *evaluation* phase, teams did a final submission that was scored according to the remainder of the evaluation trials.

Evaluation was performed in terms of minimum Tandem Detection Cost Function (tDCF) [4, 5], as primary metric, and Equal Error Rate (EER) as a secondary metric. We refer the reader to [1, 2] for the specific details and choice of parameters.

3. Data augmentation

One of the main objectives of ASVspoof 2021 is the development of countermeasures that are robust to channel effects and out-of-domain data. Data augmentation plays an essential role in achieving that goal, especially when those effects or data variability are not present in the training data. This section describes our approach to data augmentation for both PA and LA tasks. In the first case, we used data augmentation to increase the amount of data available for neural network training, accounting, in part, for the relatively small size of the training set. In the case of the LA task, data augmentation played a central role in our approach to the problem, in order to achieve the desired robustness to unseen coding and transmission artefacts.

3.1. Physical Access

For the PA task, we generated 3 new versions of each training sample: 2 speed-perturbed versions at, respectively, 0.9 and 1.1 the original speed, and a reverberated version created convolving the original audio with room impulse responses (the set described in [6] was used[1]). The size of the resulting training set was therefore 4 times the original training set.

3.2. Logical Access

Since the LA task focuses on robustness against codecs and transmission channel effects, we generated different versions of each training sample by applying a number of transformations in order to simulate these effects. We grouped these transformations into two categories: multimedia and telephony.

Multimedia transformations consisted in:

- MP3 encoding using a Constant Bit Rate (CBR) of 24, 64 and 192 Kbps.
- AAC encoding using 16, 32 and 112 Kbps CBR.
- OGG encoding at $\sim$80, $\sim$128 and $\sim$256 Kbps Variable Bit Rate (VBR).

In the case of telephony transformations, utterances were first downsampled to 8kHz prior to applying the transformation and then upsampled back to 16kHz. The following transformations were used:

- a-law encoding.
- u-law encoding.
- g.729 encoding.
- VAD. Some switchboard configurations can apply an aggressive Voice Activity Detection (VAD) over the speech signal. To simulate this, we applied an energy-based VAD to remove silence portions.

For each set of transformations, a subset of twice the size of the original training set was randomly selected. This way, the size of the final training set, when both sets of transformations were used, was 5 times the original training set size.

[1] https://www.openslr.org/28/

4. Model Description

4.1. Physical Access

Our approach to the Physical Access track was two-fold: on the one hand, we tried to develop a main system with the best possible performance. On the other hand, we developed a set of systems that could provide complementary information, thus increasing the discriminative power of a system fusion.

All systems are based on a Time Delay Neural Network (TDNN) architecture that is well-known in the Speaker Recognition literature [7], but differ on network depth and input features. Complementary systems use the TDNN architecture summarized in Table 2 while the main system uses a larger version [8] that is shown in Table 3. Note that our architectures, shown in tables 2 and 3, differ in some implementation details, in line with [9], with respect to the original implementation in [7, 8].

Table 2: Complementary systems TDNN architecture. T denotes the number of input frames. (Number of parameters in the first layer assumes 24-dimensional feature vectors.)

Layer Type	Filter/Stride	Output	Params
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	64K
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	1.313M
Conv1D-batchnorm-ReLU	$7 \times 1/1 \times 1$	$T \times 512$	1.837M
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Dense-batchnorm-ReLU	-	$T \times 1500$	775.5K
Stats Pooling (mean+stddev)	-	3000	-
Dense-batchnorm-ReLU	-	512	1.538M
Dense-batchnorm-ReLU	-	512	264.7K
Softmax	-	2	1026
Total			6.06M

Table 3: Main system large-TDNN architecture.

Layer Type	Filter/Stride	Output	Params
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	104.9K
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 512$	1.313M
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Conv1D-batchnorm-ReLU	$7 \times 1/1 \times 1$	$T \times 512$	1.837M
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Conv1D-batchnorm-ReLU	$7 \times 1/1 \times 1$	$T \times 512$	2.361M
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Dense-batchnorm-ReLU	-	$T \times 512$	264.7K
Dense-batchnorm-ReLU	-	$T \times 1500$	775.5K
Stats Pooling (mean+stddev)	-	3000	-
Dense-batchnorm-ReLU	-	512	1.538M
Dense-batchnorm-ReLU	-	512	264.7K
Softmax	-	2	1026
Total			9.52M

The first layers of the TDNN operate at the frame level, with a small temporal context centered around the current frame t. The pooling layer receives the output of these layers as input, aggregates over time, and computes its mean and standard deviation. These statistics are concatenated and passed to the final layers, which operate at the utterance level. In the Speaker Recognition field, these final utterance-level layers are usually used to extract embeddings known as *x-vectors*. In our case, we use the network as a binary classifier, by taking the output of the final classification layer. However, we also experimented with the idea of using the network to compute embeddings and use these embeddings as features to train new classifiers (see Section 4.1.2).

The choice of this architecture was motivated by the good and consistent performance that these models have shown in speaker recognition tasks and its relatively small number of parameters, which could be an important factor given the limited amount of training data available. The TDNNs were implemented using TensorFlow [10].

Next sections describe both our main individual system (Section 4.1.1) and the complementary systems (Section 4.1.2).

4.1.1. Main system

Our main system uses as input features 40-dimensional Mel Frequency Cepstral Coefficients (MFCCs) computed from 50 filters between 20 and 7900 Hz and using a 25 ms window with 15 ms overlap. No feature normalization or Voice Activity Detection (VAD) was used, since we found both techniques to degrade performance during our preliminary experiments.

The neural network architecture is the *large-TDNN* model summarized in Table 3 above. Training was performed using Stochastic Gradient Descent with an exponentially decaying learning rate with initial value of 0.005. Mini-batches were constructed so that they contained 30 utterances for each class. Training stopped when validation loss did not improved for 10 epochs.

We used Focal Loss as the objective function. This loss, originally proposed in the context of object detection [11], has been recently used successfully for spoofed speech detection [12]. It is particularly well suited for class-imbalanced tasks, placing more emphasis on hard-to-classify samples. Our preliminary results (see Section 5) showed that, indeed, focal loss provided higher accuracy than the usual cross-entropy objective function.

The system was trained on ASVspoof 2109 *train* and *dev* partitions augmented as described in Section 3.

4.1.2. Complementary systems

As mentioned before, we built a number of systems in order to capture information that could complement the main system. All DNN systems share the same TDNN architecture shown in Table 2 above but differ on input features. Additionally, we explored the use of the embeddings computed by one of these systems as input for different classifiers.

Neural network training followed the same approach described in previous section with two exceptions: a) cross-entropy was used instead of focal loss and b) no data augmentation was used except for the *tdnn-MFCC* system, which used the same training data as our main system.

tdnn-MFCC system: This system uses 24-dimensional cepstral coefficients extracted using a 30 Mel-spaced filterbank with f_{min}=20 Hz, and f_{max}=7900 Hz. A window of 25 ms with 15 ms overlap is used.

tdnn-logFBE system: The input to this system are log-filterbank energies computed from 60 triangular filters linearly distributed between 0 and 8000 Hz and using a 25 ms window with 15 ms overlap.

tdnn-CQCC system: For this system, 30 static Constant-Q Cepstral Coefficients (CQCCs) [13] are computed with f_{min}= 15, f_{max}=8000 Hz and 96 bins per octave.

tdnn-SCMC system: Previous work [14, 15] has shown that Spectral Centroid Magnitude Coefficients (SCMCs) can provide a robust front-end for spoofing detection. This system uses as input 90-dimensional feature vectors composed of 30 static Spectral Centroid Magnitude Coefficients together with Δ and $\Delta\Delta$ coefficients. (Note that this is the only subsystem for which we used Δ and $\Delta\Delta$ as input).

Embedding systems: We used the embeddings computed from the *tdnn-MFCC* as input features to train a downstream classifier. Different classifiers, like Support Vector Machines or Random Forests were tried. The best results, however, were obtained using a Gaussian Linear Classifier (GLC). This model assumes that *bona fide* embeddings w_b are generated by a Normal distribution

$$w_b \sim \mathcal{N}\left(\overline{w}_b, \Sigma_b\right),$$

where $\overline{w}_b$ is the mean embedding for *bona fide* data and Σ_b the corresponding covariance matrix. This way, given a test embedding w_t, we compute the corresponding score as the following log-likelihood:

$$score = \log p\left(w_t | \overline{w}_b, \Sigma_b\right)$$

4.2. Logical Access

For the Logical Access task we focused in data augmentation as described in Section 3. We explored three different models: the two deep learning baselines provided by the organization and a lightweight TDNN architecture, similar to the one used for the PA task but with a much smaller number of parameters.

Next sections describe in more detail each one of these models.

4.2.1. RawNet2

RawNet2 is a convolutional neural network architecture that operates directly upon the raw speech waveform, avoiding the limitations of using knowledge-based, hand-crafted acoustic features. It was originally proposed for speaker recognition [16] and was recently adapted for antispoofing in [17]. The appeal of this end-to-end approach is that, not relying in features handcrafted to detect specific cues, it could perform better against unforeseen attacks.

We used the Python baseline provided by the organizers and explored different data augmentation strategies. The input to the network is a fixed-duration waveform of $\approx$ 4 sec (64000 samples). Training uses Adam optimisation and a learning rate of 0.0001.

4.2.2. LFCC-LCNN

LCNN is an architecture that has been applied to great success to antispoofing in previous work [18]. The ASVspoof 2021 organizers provided a LFCC-LCNN Python baseline that includes

some modifications to the original architecture in line with the findings in [19]. In particular, the layers after the CNN part are replaced by two Bidirectional Long Short-Term Memory (Bi-LSTM) layers, an average pooling layer and a fully connected (FC) layer, thus notably reducing the number of parameters.

Input features are 20 high-spectral resolution Linear Frequency Cepstral Coefficients (LFCCs) with first and second derivatives as in [20]. Neither voice activity detection (VAD) nor feature normalization is used.

During training, Adam optimizer is used with an initial learning rate of of 3×10^{-4} which was multiplied by 0.5 every ten epochs. The mini-batch size was 64, with each mini-batch containing similar duration random trials, and an early stop criterion of 50 epochs without improvement was used.

4.2.3. Lightweight TDNN

Our third system for the LA task is a Time Delay Neural Network (TDNN), similar to those used for the PA task, but with a much lower number of parameters. This lightweight TDNN architecture is shown in Table 4. The input to the system are 24-dimensional cepstral coefficients extracted using a 30 Mel-spaced filterbank with f_{min}=20 Hz, and f_{max}=7600 Hz and a window of 25 ms with 15 ms overlap. No feature normalization or VAD is used.

Training was performed using Stochastic Gradient Descent with initial learning rate of 0.005 and exponential decay. Mini-batches contained 30 utterances for each class and an early stopping criterion of 10 epochs without validation loss improvement was set. As in the case of the main PA model, Focal Loss was used as the objective function.

Table 4: lightweight TDNN. T denotes the number of input frames.

Layer Type	Filter/Stride	Output	Params
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 64$	8K
Conv1D-batchnorm-ReLU	$5 \times 1/1 \times 1$	$T \times 64$	20.8K
Conv1D-batchnorm-ReLU	$7 \times 1/1 \times 1$	$T \times 64$	28.9K
Dense-batchnorm-ReLU	-	$T \times 64$	4.4K
Dense-batchnorm-ReLU	-	$T \times 500$	34.5K
Stats Pooling (mean+stddev)	-	1000	-
Dense-batchnorm-ReLU	-	64	64.3K
Dense-batchnorm-ReLU	-	64	4.4K
Softmax	-	2	130
Total			165.5K

5. Results and Discussion

5.1. Physical Access

As already mentioned, our approach for the PA task was to a) try to have a main system with the best possible individual performance and b) train some systems that could provide complementary information and improve the performance of a system fusion.

Table 5 summarizes some preliminary results obtained during the development of our main system. Comparing the first two rows, we can see that using focal loss as the objective function provides an appreciable improvement over cross entropy, particularly in terms of min-tDCF. Focusing on the choice of neural network architecture, Table 5 shows that the large-TDNN architecture (second row) provides the best results, followed by the standard TDNN architecture. The last row shows the results obtained with the lightweight-TDNN architecture proposed in Section 4.2.3. Although it is behind the other two architectures in terms of performance, it still provides competitive results, particularly considering the reduced number of parameters (165K compared to 6 or 9.5M for TDNN and large-TDNN respectively).

Table 5: Physical Access main system preliminary results.

System	Loss	Progress		Evaluation	
		min-tDCF	EER	min-tDCF	EER
large-TDNN	CE	0.7950	28.69	0.8718	33.71
large-TDNN	Focal	0.7467	27.40	0.8045	32.09
TDNN	Focal	0.7778	28.03	0.8468	32.39
lightweight-TDNN	Focal	0.8578	31.51	0.8998	35.68

Table 6 shows the results obtained by the baselines provided by the organizers (italics), our individual systems and system fusions (fusions were performed by simply combining all scores with equal weights). We can see that our best individual system provides a significant improvement over the baselines, obtaining a 14.72% relative reduction in terms of min-tDCF and a 15.70% relative reduction in terms of EER on the *evaluation* dataset.

The fusion of all individual systems A–F further improves over our best single system achieving min-tDCF $= 0.7462$ and EER $= 29.00\%$ on the *evaluation* portion (a 7.24% and 9.63% relative reduction respectively). This was the system that we submitted for the official evaluation.

Adding the GMM baselines to the system fusion, provides further improvement obtaining min-tDCF $= 0.6658$, EER $= 24.44\%$ on the *progress* set and min-tDCF $= 0.7383$, EER $= 28.41\%$ on the *evaluation* set. This was our best performing system during the progress phase and, to the best of our knowledge, the best performing system overall on that phase. It is interesting to note that all complementary systems B$-$F contribute favorably to the performance of the final ensemble, even though the individual performance of some of these systems is poor compared to the best system and the fusion is performed with equal weights. This is in line with previous findings [21] that highlight the need of combining different sources of information in order to detect different kinds of attacks.

Finally, Table 7 shows some post-evaluation results obtained by using the TDNN as an embedding extractor and using those embeddings as input for different classifiers. The difference with respect to system F in Table 6 is that, for these post-evaluation experiments, we used the large-TDNN system as embedding extractor, while system F used the TDNN-MFCC system. As already discussed, the best results are obtained by the Gaussian Linear Classifier (see Section 4.1.2). Indeed, this system provides a significant improvement with respect to using the large-TDNN system as a classifier (first row), which was our best individual system during the evaluation phase. Moreover, it is worth noting that, although the embedding extractor was trained using both *bona fide* and *spoofed* speech and therefore both classes are, in principle, needed to compute discriminative embeddings, the GLC itself is a one-class classifier, built using exclusively the *bona fide* samples. This could have positive implications in the sense of providing stability against unknown attacks or avoiding overfitting to specific attacks, and is something we plan to explore further in future work.

5.2. Logical Access

As discussed above, our focus for the LA task was on data augmentation. Table 8 shows some preliminary results to test the

Table 6: Results on the Physical Access task.

ID	System	Progress		Evaluation	
		min-tDCF	EER	min-tDCF	EER
	CQCC-GMM baseline	*0.9033*	*36.33*	*0.9434*	*38.07*
	LFCC-GMM baseline	*0.9741*	*39.79*	*0.9724*	*39.54*
	LFCC-LCNN baseline	*0.9809*	*42.16*	*0.9958*	*44.77*
	RawNet2 baseline	*0.9992*	*46.03*	*0.9997*	*48.60*
A	large-TDNN	**0.7467**	**27.40**	**0.8045**	32.09
B	TDNN-MFCC	0.7875	28.38	0.8500	**31.95**
C	TDNN-SCMC	0.8769	33.46	0.9339	36.10
D	TDNN-logFBE	0.9061	36.62	0.9217	39.15
E	TDNN-CQCC	0.9940	41.33	0.9966	43.17
F	embedding-GLC	0.8282	31.18	0.8813	33.63
	A + B + C + D + E + F	0.6675	24.80	0.7462	29.00
	+ GMM baselines	**0.6658**	**24.44**	**0.7383**	**28.41**

Table 7: Results for different classifiers on top of large-TDNN embeddings.

Classifier	Progress		Evaluation	
	min-tDCF	EER	min-tDCF	EER
None (System A)	0.7467	27.40	0.8045	32.09
Support Vector Machine	0.7548	26.41	0.8024	29.39
Random Forest	0.7862	28.53	0.8377	34.06
GLC	**0.7384**	**25.70**	**0.7793**	**28.51**

validity of our data augmentation approach (see Section 3). Although these results refer to a lightweight-TDNN architecture, the general trends were the same for all of the 3 considered architectures. We can see that both sets of transformations, multimedia and telephony, provide a substantial improvement. Moreover, combining both sets achieves the best results almost halving both min-tDCF and EER with respect to the baseline system with no data augmentation. However, increasing the size of each set from 2 to 4 times the size of the original training set did not bring any further improvement. This seems to indicate that the observed improvement is not caused by an increase on the amount of training material but an increase in its variety.

Table 8: Preliminary data augmentation experiments. Lightweight-TDNN and CE loss were used in all cases.

Data Aug.	**min-tDCF**	**EER(%)**
None	0.6129	17.89
Multimedia	0.4817	14.20
Telephony	0.4688	13.18
Both	0.3626	8.71

Table 10 shows our final challenge results (baseline systems provided by the organizers are shown in italics). We can see, again, that data augmentation proved to be effective for all the three systems providing a consistent reduction in terms of both min-tDCF and EER. Comparing systems F and G, we can also see the effectiveness of using focal loss as the objective function. The best performing individual system is RawNet2 with min-tDCF $= 0.3168$, EER $= 6.36\%$ on the *evaluation* set, followed by the LFCC-LCNN system and the lightweight-TDNN system which obtains min-tDCF $= 0.3645$ and EER $= 7.51\%$. It is worth noting that despite its small size (a comparative of the three systems in terms of number of parameters is shown in Table 9) the lightweight-TDNN system is able to obtain competitive results across both the LA and PA tasks.

The fusion of all three systems, B, D and G, with equal weights, obtains our best results with min-tDCF $= 0.2371$, EER $= 4.54\%$ on the *progress* set and min-tDCF $= 0.2747$, EER $= 5.58\%$ on the *evaluation* set. This represents a 20.26% and 39.74% relative improvement over the best baseline in terms of min-tDCF and EER respectively. This was our official submission for the challenge.

Table 9: Number of parameters of each considered architecture.

Model	**# Params**
RawNet2	17M
LFCC-LCNN	290K
lightweight-TDNN	165K

6. Conclusions

We have presented our systems for the ASVspoof 2021 challenge LA and PA tasks. In both cases, our best systems significantly outperformed the baseline systems provided by the organization. We proposed the use of focal loss as the objective function during neural network training and showed that it provides an improvement with respect to the usual cross entropy loss that is consistent across different architectures and tasks. One of our proposed architectures, the lightweight-TDNN, provided, despite its small number of parameters, a performance comparable to that of much larger models. Moreover, this was the case for both LA and PA tasks. The development of countermeasures that take into account practical restrictions, like model footprint, is something we plan to explore in future work. Another line of work that we plan to continue working on is the use of classifiers like the GLC on top of neural network embeddings developed for the PA task. On the one hand, this classifier provided the best individual performance, improving the results obtained by using the neural network as a classifier. On the other hand, although the training of the embedding extractor was performed using both bona fide and spoofed speech, we do believe that the fact that the final classifier is built using only genuine samples holds a lot of promise, particularly if the need for spoofed samples for embedding extractor training could be relaxed to some extent.

Table 10: Results on the Logical Access task.

ID	System	Data Aug	Progress		Evaluation	
			min-tDCF	EER	min-tDCF	EER
	CQCC-GMM baseline	-	*0.4948*	*15.80*	*0.4974*	*15.62*
	LFCC-GMM baseline	-	*0.5836*	*21.13*	*0.5758*	*19.30*
	LFCC-LCNN baseline	-	*0.3152*	*8.90*	*0.3445*	*9.26*
	RawNet2 baseline	-	*0.4152*	*9.49*	*0.4257*	*9.50*
A	LFCC-LCNN	No	0.3207	10.50	0.3477	10.48
B	LFCC-LCNN	Yes	**0.2817**	**5.60**	0.3303	6.76
C	Rawnet2	No	0.3852	9.02	0.4286	9.39
D	Rawnet2	Yes	0.2824	5.79	**0.3168**	**6.36**
E	lightweight TDNN CE	No	0.6129	17.89	0.6478	19.20
F	lightweight TDNN CE	Yes	0.3626	8.71	0.3928	9.21
G	lightweight TDNN Focal	Yes	0.3263	7.33	0.3645	7.51
	B + D		0.2555	5.36	0.2936	5.97
	B + D + F		0.2457	4.94	0.2855	**5.47**
	B + D + G		**0.2371**	**4.54**	**0.2747**	5.58

7. Acknowledgements

The authors would like to thank the organizers of ASVspoof 2021 challenge and Juan Manuel Espín López for his help on developing the TDNN architectures.

8. References

[1] "ASVspoof 2021: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan [online]," `https://www.asvspoof.org/asvspoof2021/asvspoof2021_evaluation_plan.pdf`, 2021.

[2] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, José Patino, Andreas Nautsch, Kong-Aik Lee X. Liu, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado, "ASVspoof2021: accelerating progress in spoofed and deep fake speech detection," in *Proc. ASVspoof 2021 Workshop*, 2021.

[3] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi H. Kinnunen, and Kong Aik Lee, "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *Proc. Interspeech 2019*, 2019, pp. 1008–1012.

[4] Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A. Reynolds, "t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification ," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 312–319.

[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, 2020.

[6] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[7] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[8] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5796–5800.

[9] Yi Liu, Liang He, and Jia Liu, "Large Margin Softmax Loss for Speaker Verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[12] Mari Ganesh Kumar, Suvidha Rupesh Kumar, M. Saranya, B. Bharathi, and H. Murthy, "Spoof detection using time-delay shallow neural network and feature switching," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1011–1017, 2019.

[13] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech and Language*, vol. 45, pp. 516–535, 2017.

[14] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Dresden, Germany, Sept. 2015, pp. 2087–2091.

[15] Roberto Font, Juan M. Espín, and María José Cano, "Experimental analysis of features for replay attack detection — results on the asvspoof 2017 challenge," in *Proc. Interspeech 2017*, 2017, pp. 7–11.

[16] Jee weon Jung, Seung bin Kim, Hye jin Shim, Ju ho Kim, and Ha-Jin Yu, "Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms," in *Proc. Interspeech 2020*, 2020, pp. 1496–1500.

[17] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher, "End-to-End anti-spoofing with RawNet2," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6369–6373.

[18] 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.

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

[20] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco, "Spoofing Attack Detection Using the Non-Linear Fusion of Sub-Band Classifiers," in *Proc. Interspeech 2020*, 2020, pp. 1106–1110.

[21] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco, "An Explainability Study of the Constant Q Cepstral Coefficient Spoofing Countermeasure for Automatic Speaker Verification," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 333–340.