2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

ANALYSING REPLAY SPOOFING COUNTERMEASURE PERFORMANCE UNDER VARIED CONDITIONS

Bhusan Chettri[1], *Bob L. Sturm*[1,2], *Emmanouil Benetos*[1]

[1]School of EECS, Queen Mary University of London, United Kingdom
[2]School of EECS, KTH Royal Institute of Engineering, Stockholm, Sweden

ABSTRACT

In this paper, we aim to understand what makes replay spoofing detection difficult in the context of the ASVspoof 2017 corpus. We use FFT spectra, mel frequency cepstral coefficients (MFCC) and inverted MFCC (IMFCC) frontends and investigate different backends based on Convolutional Neural Networks (CNNs), Gaussian Mixture Models (GMMs) and Support Vector Machines (SVMs). On this database, we find that IMFCC frontend based systems show smaller equal error rate (EER) for high quality replay attacks but higher EER for low quality replay attacks in comparison to the baseline. However, we find that it is not straightforward to understand the influence of an acoustic environment (AE), a playback device (PD) and a recording device (RD) of a replay spoofing attack. One reason is the unavailability of metadata for genuine recordings. Second, it is difficult to account for the effects of the factors: AE, PD and RD, and their interactions. Finally, our frame-level analysis shows that the presence of cues (recording artefacts) in the first few frames of genuine signals (missing from replayed ones) influence class prediction.

Index Terms— Automatic speaker verification, spoofing detection, replay attack, spoofing countermeasure.

1. INTRODUCTION

Automatic speaker verification (ASV) [1] systems aim at verifying whether a person is who they claim to be. Recently, ASV technologies have been widely adopted commercially for user authentication (eg. mobile phones) [2]. These systems however are vulnerable towards spoofing attacks [3], which involve using artificial/synthetic speech to bypass the ASV systems of a registered user. The four common attacks are: (1) text-to-speech; (2) voice conversion; (3) mimicry; and (4) replay. Thus a spoofing countermeasure is deployed to protect ASV systems from such attacks. It comprises of a frontend which aims at capturing discriminative attributes from the speech signal that is used by the backend for classification (often in binary setting).

Among all, the simplest spoofing approach is the replay attack, which involves playing pre-recorded speech of a registered user to the ASV system. Many researchers have explored the vulnerability of replay attacks using their in-house databases and protocols, eg. [4, 5, 6, 7]. These results are often difficult to reproduce which adds limitation to the growth of anti-spoofing research. However, with the availability of the ASVspoof 2017 Challenge replay corpus which has standard protocols and evaluation metrics, research has become more transparent and results comparable. Some of the best reported systems on the version 1.0 corpus used fusion approaches either at the feature or score level. For example, the best system [8] used score-level fusion of three different systems (two of which uses deep neural networks, DNNs) and [9] used score-level fusion of GMM systems trained on rectangular filter cepstral coefficients and linear filter cepstral coefficients. Authors in [10] use feature fusion (constant-Q-cepstral coefficients and high frequency cepstral coefficients) to train a DNN as a feature extractor. Using the DNN features they further train a binary SVM for replay spoofing detection.

In this paper we explore CNN, GMM and SVM backends for replay spoofing detection using the ASVspoof 2017 version 2.0 corpus. Motivated from the findings of [11] about ambient and reverberation noise being reliable indicators of replay spoofing detection, we investigate the use of IMFCC, MFCC and spectrogram frontends. Next, we analyse the performance of these systems across different spoofing conditions (section 2) to understand how the following factors: acoustic environment, playback and recording devices, influence spoofing detection. On this database, however, we find it hard to analyse these factors in isolation for two reasons: (1) Unavailability of meta-data for genuine recordings; (2) Segregating the three factors AE, PD and RD from a replayed signal is difficult. Our further analysis on frame-level energy and log-likelihood distributions shows existence of the cues in the genuine signals, similar to the findings of [12] on version 1.0 of the corpus.

The rest of the paper is organized as follows. We provide a background on the ASVspoof 2017 corpus in the next section. In section 3 we present details of our countermeasures used in this work. We discuss the results in section 4. We provide performance analysis under different replay conditions in section 5 followed by conclusions in section 6.

2. THE ASVSPOOF 2017 CORPUS

The ASVspoof 2017 version 1.0 corpus [13] has been released as a part of the second automatic speaker verification spoofing and countermeasures challenge [14] designed to foster research in “replay spoofing” countermeasures. Post-evaluation, [12] demonstrated how class predictions could be manipulated using the cues present in some of the genuine audio recordings of the corpus. Subsequently, version 2.0 [11] has been released online[1] addressing these data anomalies. Table 1 shows the database statistics. The distribution of audio files and their class label remains the same in both the versions. However, significant updates in the replay meta-data have been made in the new version.

EB is supported by a RAEng Research Fellowship (RF/128). This research was supported by an NVIDIA GPU Grant.

[1] https://datashare.is.ed.ac.uk/handle/10283/3017

978-1-5386-5477-4/18/$31.00 ©2018 IEEE

Table 1. Statistics of the ASVspoof 2017 2.0 corpus. RC denotes replay configurations. Dur: Duration in hours.

Subset	**# Spk**	**# RC**	**# Genuine**	**# Replay**	**Dur**
Train	10	3	1507	1507	2.22
Dev	8	10	760	950	1.44
Eval	24	57	1298	12008	11.94
Total	42	61	3565	14465	15.6

Next we briefly discuss different replay conditions/configurations that shall be referred throughout the paper. A replay configuration (RC) comprises of a unique combination of three elements: a recording device (RD), a playback device (PD) and an acoustic environment (AE) where the replay is simulated. These three elements are the factors of interest in a replay attack. In this study we seek to understand the influence of these factors both in isolation and conjunction towards replay spoofing attack. The organisers used a total of 26 different AE, 26 PD and 25 RD to build the ASVspoof 2017 corpus. However, only 61 unique RCs were used to simulate the replay attacks from a space of 26 × 26 × 25 possible combinations. We find one overlapping RC between the training and evaluation subsets and seven between the development and evaluation subsets. Each of the replay factors has been grouped into three categories [11] depending on the level of threat they present to an ASV system. (1) **Low**, signifies use of a low quality RD, PD and noisy AE (eg. balcony) to simulate a replay attack. (2) **Medium**, signifies use of a medium quality RD, PD and a medium noise AE (eg. office). (3) **High**, indicates the use of a high quality RD, PD and a low noise AE (eg. studio).

As demonstrated in [11], low quality replay attacks (all AE, PD and RD are of low quality) are easily detected by any countermeasure due to the accumulation of reverberation and background noise in a replayed signal. On the other hand, high quality replay attacks (all AE, PD and RD are of high quality) pose greater threats to ASV systems leaving no trace to be detected by such countermeasures.

3. EXPERIMENTAL SETUP

In this section we discuss the baseline [15, 11] and the systems we investigated: (1) Convolutional Neural Networks (CNNs) trained on spectrograms (2) Gaussian Mixture Models (GMMs) trained on MFCC and IMFCCs (3) Support Vector Machines (SVMs) trained on MFCC i-vectors and IMFCC i-vectors and finally (4) Fusion systems. We use pooled (train+development) data for training all our systems except the CNNs. The CNNs are trained on the training subset and validated using the development subset.

We use the equal error rate (EER) metric to assess our system performance. We compute the EER using the Bosaris toolkit [16].

3.1. Baseline systems

The original baseline system [15] is a 512-component Gaussian Mixture Model (GMM) trained on 29 dimensional constant-q cepstral coefficients (CQCC) including the 0^{th}, delta and acceleration coefficients. Several experiments were carried out by [11] to improve the original baseline system. The best baseline, referred to as *enhanced baseline*, applies Cepstral Mean Variance Normalization, replaces the zeroth coefficient with log energy and uses 19 base coefficients instead of 29.

3.2. CNN-based systems

First, we investigate two utterance-based end-to-end replay countermeasures using CNNs. Motivated from [8], we use power spectrogram as the input representation to the CNNs. We choose the initial 3 seconds from each audio signal to obtain a consistent[2] input representation. For this, we replicate the audio samples if the duration is smaller or truncate the samples to a 3 second duration. We use a 256 point FFT[3], and a 16 ms window with a hop size of 10 ms. Thus, our input to the CNNs is a mean-variance normalized log power spectrogram of 300 × 129 (time × frequency) dimension, where time denotes the number of frames and frequency the number of bins. We use the Librosa[4] library for computing the spectrograms.

We train the network to optimize cross entropy loss between a genuine and a spoof class. We initialize the network weights using Xavier initialization [17]. We set all biases to zero. We use the standard ReLU non-linearity, a learning rate of 1e-4, a batch size of 32, and the ADAM [18] optimizer. We use the TensorFlow [19] for CNN implementation with early stopping: if the validation loss does not improve for 30 epochs we abort the training. We use a maximum of 300 training epochs and choose the model that shows the best performance on the validation data. We apply different dropout rates to the inputs of fully connected layers depending on the model architecture. At inference time, for each test utterance we convert the posterior probability distribution of the genuine and spoof class into a log-likelihood ratio (used as a score) and compute the EER.

The architecture of our first CNN system, CNN1, is adapted from $LCNN_{FFT}$ [8] that showed promising results on the version 1.0 database. It comprises of 5 convolution (Conv) layers, 4 Network in Network (NIN) layers, 10 Max-Feature-Map (MFM) layers, 5 max-pooling layers and 2 fully connected (FC) layers. Our preliminary experiments showed similar performance for MFM and ReLU activations on the version 2.0 database, therefore we do not use the Max-Feature-Map layers. We apply 70% and 50% dropouts to the inputs of the two FC layers respectively. The network has 99.3K free parameters.

The second CNN system, CNN2, is adapted from [20]. We are motivated to explore this architecture on the version 2.0 database since it has a comparatively small number of free parameters (about 11K). It has three Conv layers and two FC layers. Each Conv layer has 16 output filters/feature maps and uses a small rectangular filter of 1 × 9 with a stride of 1 × 1 along time and frequency. We apply a max-pooling operation after each Conv layer. We use 3 × 3 kernel and 3 × 3 stride in all max-pooling layers. We use 32 neurons in the first FC layer with a linear activation and two neurons in the output layer. We apply 80% and 50% dropouts to the inputs of FC layers to counter overfitting.

3.3. GMM-based systems

We use two standard short-time spectral features: mel frequency cepstral coefficients (MFCCs) [21] and inverted mel frequency cepstral coefficients (IMFCCs) [22] to train a Gaussian Mixture Model (GMM) backend. We train one GMM for each genuine and spoof class using the expectation maximization algorithm with random initialization. We use the MSR-Identity toolkit [23] for implementation. At test time, for each test utterance a score is obtained using the log-likelihood ratio:

[2] The average duration of the training subset is 2.66 sec

[3] We find a comparable performance for 256,512 and 1024 FFT points

[4] http://librosa.github.io

$$\Lambda(X) = \log P(X|\theta_g) - \log P(X|\theta_s) \quad (1)$$

where X is a sequence of feature vectors, P denotes the likelihood function, θ_g and θ_s represents the genuine and spoof GMMs respectively. We develop two GMM systems: GMM1 and GMM2. GMM1 uses 512 mixture components trained on MFCC features while GMM2 is trained on IMFCC features with 256 mixture components. Both systems use 40 dimensional delta+acceleration (DA) coefficients.

3.4. SVM-based systems

We do not apply any normalization or voice activity detection to the MFCC and IMFCC frontends. The same holds true for section 3.3.

We train two utterance-level binary support vector machine (SVMs) backends using i-vectors [24]. SVM1 is trained using MFCC i-vectors and SVM2 uses IMFCC i-vectors. We use 40 dimensional DA coefficients (for both MFCC and IMFCC) to learn the i-vector extractor (also called total variability matrix T) and the universal background model (UBM). We use the pooled (train+development) data to train the UBM with 128 mixture components and the T matrix with 100 rank. i-vectors are extracted for the train, development and evaluation subsets and a SVM classifier with a linear kernel is trained on pooled i-vectors to discriminate a genuine and a spoof class. SVMs are implemented using Scikit-learn [25] and the i-vector extractor is trained using the MSR-Identity toolkit.

3.5. Ensemble systems

We argue that a single feature and a single classifier may not adequately model the diverse spoofing conditions that appear in the ASVspoof 2017 evaluation subset. To this end, we investigate performance using ensemble approaches which have shown promising results in version 1.0 ASVspoof 2017 [8] and also on ASVspoof 2015 database (text-to-speech and voice-conversion spoofing attacks) [26, 27]. We build four score-level fusion[5] systems using the linear logistic regression implementation of [16]. Fused1 combines scores from all the six countermeasures: CNN1, CNN2, GMM1, GMM2, SVM1 and SVM2. Fused2 combines the GMM1 and GMM2 scores while Fused3 fuses the scores of the SVM1 and SVM2 systems. Finally, Fused4 combines the two GMMs and the two SVMs.

4. RESULTS

Table 2 shows the performance of our systems on the evaluation subset of the ASVspoof 2017 version 2.0 corpus. The original baseline and the enhanced baseline produces an EER of 23.4% and 12.2% respectively. See [15] and [11] for details on the original and enhanced baseline results. The end-to-end CNN systems show poor performance on the evaluation subset. CNN1 shows an EER of 28.2% and CNN2 yields an EER of 27.81%. A possible reason for poor generalization might attribute to a small amount (only 2.22 hours) of data available for training the models. Data augmentation approaches may help improve generalization.

The IMFCC feature based GMM system (GMM2) shows an EER of 18.3%, clearly outperforming the MFCC feature based system (GMM1) with an EER of 27.8%. We find a similar trend in performance for i-vector based SVM systems. The SVM2 system trained on IMFCC-based i-vectors shows an EER of 16.3%, outperforming the MFCC-based i-vector system (SVM1) having 24.6% EER. Thus IMFCCs that emphasize higher frequencies seem to give better performance over MFCCs in general.

Table 2. Performance of the baselines, GMMs, SVMs, CNNs and fused systems on the evaluation subset. Z,S,D,A denote zero, static, delta and acceleration coefficients.

Id	System	Features	EER%
1	Original baseline [15]	29 ZSDA CQCC	23.4
	Enhanced baseline [11]	19 E_SDA CQCC	12.2
2	CNN1	Spectrograms	28.2
	CNN2		27.8
3	GMM1	MFCC	27.8
	GMM2	IMFCC	18.3
4	SVM1	MFCC_ivector	24.6
	SVM2	IMFCC_ivector	16.3
5	Fused1	Systems in 2-4	12.3
	Fused2	GMM1+GMM2	15.9
	Fused3	SVM1+SVM2	11.5
	Fused4	Systems in 3-4	**11.0**

By now, it is quite evident how hard it is for a single countermeasure to counter the diverse nature of replay attacks in the evaluation subset. Thus, we investigate the benefits that these countermeasures offer as an ensemble system. The first ensemble system, Fused1, produces an EER of 12.3%, offering about 11% absolute gain over the original baseline (23.4%) and comparable performance with the enhanced baseline. The Fused2 system shows an EER of 15.9% and the Fused3 system an EER of 11.5%. Our best fusion system Fused4 reports an EER of 11.0% outperforming the original and the enhanced baselines by an absolute of 13.4% and 1.2%, respectively. These results suggest that ensemble approaches could be one possible direction for further investigation.

The results seen so far do not explain what these systems have learned to make prediction or which factor (AE, PD, RD or RC) influences the prediction most. Thus, we perform analysis on the countermeasure performance under different replay conditions in the next section.

5. ANALYSIS

We compare and analyse the performance of various countermeasures under different replay conditions/configurations we highlighted in section 2.

5.1. Impact under different quality of AE, RD and PD

We pool all the evaluation subset scores according to low, medium and high qualitative categories for each factors: acoustic environment (AE), playback device (PD) and recording device (RD). We do this for all our systems, except CNN1; as both CNNs show similar performance, we choose CNN2 for analysis. Results are shown in Table 3. Note that the number of replayed utterances varies across different conditions (eg. 9336 for medium quality and 1633 for high quality acoustic environments) but the number of genuine utterances remains same[6]. We make several interesting observations. (1) The

[5] We use greedy approach for fusion and report the 4 best fused systems.

[6] Evaluation subset has 1298 genuine utterances. Whether we compute EER for the low AE or high AE the number of genuine utterances is always 1298.

Table 3. Spoofing detection performance (EER%) for different quality of acoustic environments (AE), playback devices (PD) and recording devices (RD) in the evaluation subset of the ASVspoof 2017 2.0 corpus. Low, medium and high quality indicator has the same meaning as in section 2. RU indicates replay utterances. Evaluation subset has 1298 genuine utterances. Bold numbers highlight the systems outperforming the enhanced baseline.

Replay factor	Quality	# RU	Enh. Baseline	Fused2	CNN2	GMM1	GMM2	SVM1	SVM2
Acoustic Environment (AE)	Low	1039	16.6	**13.3**	24.2	**13.5**	18.8	**15.5**	20.4
	Medium	9336	18.7	**11.4**	29.2	25.4	**18.5**	22.2	**16.2**
	High	1633	21.8	**14.3**	22.3	44.4	**16.9**	41.3	**13.5**
Playback device (PD)	Low	4612	16.6	**12.9**	36.4	18.3	21.9	**10.5**	21.6
	Medium	1568	16.4	**9.9**	23.9	**11.9**	**16.2**	16.4	**11.8**
	High	5828	18.3	**11.6**	20.7	35.7	**14.5**	34.8	**11.7**
Recording device (RD)	Low	5092	10.8	12.2	27.9	22.5	18.4	21.4	17.6
	Medium	1592	15.6	**10.0**	38.6	36.3	16.4	21.7	**12.7**
	High	5324	17.7	**12.3**	24.5	28.7	18.8	28.2	**15.9**

Fused2 system outperforms the enhanced baseline under each category except for low quality recording devices, indicating that ensemble approaches does help improve detection performance. (2) CNN2 shows worse performance compared to the enhanced baseline. (3) Generally, SVM systems tend to show better performance over GMMs (with few exceptions in some cases). (4) **AE:** for the low quality AE, the MFCC-based system shows better performance over the IMFCC-based systems. However, we see an opposite trend for replay attacks using medium and high quality AE. (5) **PD:** for the low quality PD, the MFCC-based ivector system SVM1 outperforms IMFCC-based systems, but on the medium and high quality PD, the SVM2 system outperforms the MFCC-based systems. (6) **RD:** for all low, medium and high quality RDs, the IMFCC-ivector based SVM2 system outperforms the MFCC-based systems.

The experiments conducted here make an assumption that while analysing the influence of one factor say, AE the other two factors PD and RD have negligible impact. This however is not true because it is difficult to mask out the information related to RD and PD from a replayed audio signal. Had this been true, the problem of replay attack would have been easy to solve already. Therefore, we argue that the results reported here may not be completely insightful to understand a replay spoofing detection system. This leads us to perform analysis according to different qualities of replay configurations[7] in the next section.

5.2. Impact under different quality of RC

We present the results of RC-wise analysis in Table 4. We consider three low quality RCs: RC15, RC16 and RC19, three medium quality RCs: RC30, RC33, RC34 and three high quality RCs: RC55, RC56 and RC57. It is worth to note that these high quality RCs use analog wire acoustic conditions, meaning there is no physical sound propagation and hence are considered to be the most difficult replay attacks to be detected by a countermeasure.

In general, the fused2 system show the best performance outperforming the enhanced baseline. **Low:** Under low quality, we observe worse performance for CNN2 and IMFCC-feature based GMM2 and SVM2 systems. Though we expected IMFCC features to show better performance as they emphasize higher frequency regions which enable capturing ambient noises, we find contradictory results. The MFCC-based systems show comparable performance with the enhanced baseline. For RC19, the GMM1 system shows 7.0% EER which further reduces to 3.5% for the i-vector based system SVM1, clearly outperforming the baseline (10.5%). **Medium:** except CNN2 all other systems show comparable or improved performance in comparison to the enhanced baseline. **High:** For high quality RCs, the MFCC-features based systems (GMM1 and SVM1) show worse performance indicating that low frequency information is not very helpful for discriminating high quality replay attacks. All other systems including CNN2 clearly outperforms the baseline for the high quality RCs we investigated. On the RC55 configuration, the GMM2 system shows a remarkable performance of 3.8%, in comparison to the baseline (15.0%) which further reduces to 3.5% for the SVM2 system using IMFCC i-vectors.

Overall, we make following observations. Within the context of ASVspoof 2017 2.0 dataset, (1) MFCCs seem to show better performance for low and medium quality replay attacks. IMFCCs on the contrary show poor performance in general. This suggests that information at low frequencies are helpful for detecting low quality attacks. (2) For the high category (the hardest ones), MFCC show the worse performance while IMFCC-feature based systems show superior performance, with the IMFCC+ivector based SVM2 system taking the lead. A possible explanation for this could be that these high quality devices may use low pass filters that mask out high frequency information in a replayed signal, leaving cues for discrimination. This hypothesis however needs further investigation which we look at in our future work.

5.3. Frame-wise energy and log-likelihood analysis

Now we conduct frame-level analysis to see if we can derive any understanding about what the MFCC and IMFCC feature-based GMM systems (GMM1 and GMM2) have learned about high quality replay attacks. For this we look at log energy and log-likelihood distributions across the frames of the most confidently classified spoof and genuine audio files under RC55[8].

Figure 1 shows the energy and log-likelihood distribution plots across the first 100 frames of genuine and spoof files for GMM2. For the spoof file $E\_1005573.wav$, the energy distribution across frames seems to be uniform and smooth. The genuine and spoof model log-likelihood across the frames show competitive behaviour, indicating how hard it is to have a clear boundary of discrimination between genuine and replayed signals. Further, we find that the log-

[7] Note that the terms: “replay conditions” and “replay configurations” are used alternatively in the paper. They have the same meaning.

[8] Among the three high quality replay attacks (RC55, RC56, RC57) we analysed, the IMFCC frontend show lowest EER for RC55, so we chose RC55 for analysis.

Table 4. Performance (EER%) under different quality of replay configurations (RC) in the evaluation subset of the ASVspoof 2017 2.0 corpus. The baseline numbers (fifth column) are estimated from the Fig. 2 of [11]. The letter E, P, R in the third column denote acoustic environment, playback device and recording device. Low, medium, high, RU and bold numbers have the same meaning as in Table 3. Evaluation subset has 1298 genuine utterances.

Replay Quality	Id	Replay Config.	# RU	Enh. Baseline	Fused2	CNN2	GMM1	GMM2	SVM1	SVM2
Low	RC15	E02 P21 R18	150	8.0	10.7	19.9	**8.0**	23.2	13.6	24.2
	RC16	E02 P21 R14	116	9.0	**6.6**	21.4	12.5	13.9	13.3	18.0
	RC19	E02 P20 R14	120	10.5	**8.5**	49.9	**7.0**	23.0	**3.5**	26.0
Medium	RC30	E15 P19 R20	74	7.0	**1.9**	16.0	**5.5**	7.1	**6.1**	**4.1**
	RC33	E13 P14 R04	183	8.5	**5.0**	12.3	**6.4**	8.9	10.5	**7.9**
	RC34	E17 P12 R04	181	9.0	**4.3**	12.5	**5.8**	**8.5**	10.2	**7.5**
High	RC55	E26 P24 R24	178	15.0	**11.0**	**9.8**	42.9	**3.8**	47.0	**3.5**
	RC56	E25 P13 R08	182	36.0	**29.2**	**22.5**	43.4	**26.5**	48.1	**22.1**
	RC57	E24 P23 R23	183	33.0	**27.4**	**26.6**	44.3	**26.4**	49.6	**22.3**

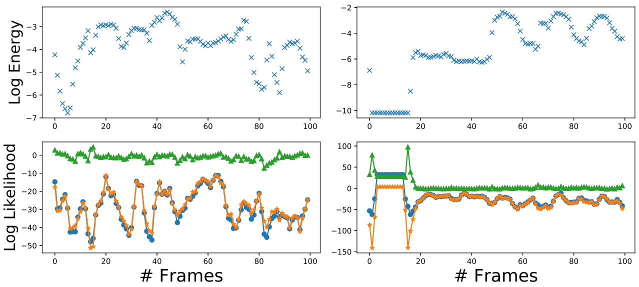

Fig. 1. Left column shows the log energy (top) and log-likelihood distribution (bottom) across the first 100 frames of the most confident spoof audio $E\_1005573.wav$ in RC55 condition for GMM2. The column on the right shows the same for the most confident genuine audio file $E\_1002092.wav$. The blue and orange profile depicts the genuine and spoof GMM log-likelihood while the green profile denotes the log-likelihood difference.

likelihood difference (green profile) across the frames seems to be around zero indicating ambiguity in the decision boundary. However, on the genuine file $E\_1002092.wav$, we find significant cues about a genuine class in the first few frames. We find lower energy for these frames in comparison to remaining frames of the signal. Further, the spoof model for these frames gives a very low likelihood score indicating that such instances were not seen during spoof GMM training. As a result, the log likelihood ratio (green profile) in these frames dominates the other frames in the signal, thus serving as a key indicator of being genuine. We refer these frames as outliers. We find 286 genuine files in the training subset, 35 in the development and 96 in the evaluation subset with such outliers. We observe a similar trend as in Fig.1 for MFCC-based GMM1 system. However, we do not include the figures due to space limitations.

From these observations, it appears that these systems are also using the class-dependent data cues (outliers) found in the genuine signals as one of the factors for making predictions. However, this would not be the case if a voice activity detector (VAD) was in place that would automatically eliminate non-speech frames. But this is not the case: these countermeasures use both speech and non-speech frames. Therefore, a realistic-real-world replay countermeasure would have to be smart enough to automatically tackle such outliers during model training and testing and make a reliable prediction.

Table 5. Confusion matrix of GMM1 and GMM2 systems for RC15 (low quality) and RC55 (high quality). G: genuine, S: spoof. Columns denote ground-truth and rows the predicted.

		RC15 (Low)		RC55 (High)	
		G	**S**	**G**	**S**
MFCC+GMM (GMM1)	**G**	1208	22	1208	162
	S	90	128	90	16
IMFCC+GMM (GMM2)	**G**	1197	116	1197	2
	S	101	34	101	176

As we notice from Table 4 (fourth column), the number of replay utterances is significantly lesser than genuine utterances (which is always 1298 for every RC condition), therefore, reported EERs do not provide significant statistical insights on the correct and incorrect classification for the genuine and spoof test utterances. So, we look at the confusion matrix for RC15 and RC55 using the GMM1 and GMM2 systems to understand the proportion of correct and incorrect classification. Table 5 shows the resulting confusion matrix. For RC15, GMM1 has high true negative (85.33%) but small false positive (14.66%) rates while GMM2 shows the opposite trend: high false positive (77.33%) and small true negative (22.66%) rates. On RC55, we see an opposite trend in contrast to RC15. Here, the GMM1 system shows high false positive (91.01%) and low true negative (8.98%) while GMM2 shows small false positive (1.12%) but high true negative (98.87%) rates. For genuine cases, both GMM1 and GMM2 show comparable performance (in terms of true positives and false negatives).

6. CONCLUSION

In this paper, we investigated and analysed various countermeasures for replay spoofing detection on the version 2.0 ASVspoof 2017 corpus. We find that the systems using MFCC frontends have a smaller EER than the systems using IMFCC frontends in the evaluation subset when looking at replay conditions with supposed low quality. We find the opposite when looking at replay conditions with supposed high quality. However, gaining in-depth understanding of what is causing this behaviour seems challenging because the original RedDots [28] corpus of genuine recordings includes both clean and noisy recordings collected from heterogeneous devices, but lacks documentation on the meta-data (acoustic conditions, recording devices). This means that "high-quality spoofing conditions" may actually be low quality since the genuine files were of low quality and vice-versa.

Thus, on this dataset it is difficult to perform evaluation on factors (AE, PD, RD and RC) influencing replay attacks in controlled conditions and provide significant conclusions whether reverberation noise or some device-specific (recording or playback) attributes provide a cue to replay signal discrimination. Further, our analysis shows that the models also use dataset-specific cues (outliers), found only in few genuine files but missing in the replayed versions, during prediction. Therefore, on this dataset, a reliable replay detector should automatically take care of such outliers and allow learning algorithms to exploit only the information related to replay factors to make a reliable prediction. However, an open question that remains is: (1) What kind of replay attacks can be detectable in the first place? (2) Is it possible to design a frontend that can automatically tackle these variabilities and uncertainties?

Our future work aims to look into countermeasures that would automatically tackle such outliers during training and testing. We will also look at how speech (the ten phrases used in the corpus) and the speakers influence replay detection on the evaluation subset.

7. REFERENCES

[1] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1, pp. 91–108, 1995.

[2] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *IEEE Signal Processing Society SLTC Newsletter*, 2013.

[3] Z. Wu et al., "Spoofing and countermeasures for speaker verification: a survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[4] Z. Wu et al., "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *APSIPA*. IEEE, 2014, pp. 1–5.

[5] H. Malik, "Securing speaker verification system against replay attack," in *46th International Conference: Audio Forensics*. Audio Engineering Society, 2012.

[6] J.Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *ICCST*. IEEE, 2011, pp. 284–291.

[7] F. Alegre, A. Janicki, and N. Evans, "Re-assessing the threat of replay spoofing attacks against automatic speaker verification," in *BIOSIG*. IEEE, 2014, pp. 1–6.

[8] G. Lavrentyeva et al., "Audio Replay Attack Detection with Deep Learning Frameworks," in *Interspeech*, 2017, pp. 82–86.

[9] R. Font et al., "Experimental analysis of features for replay attack detection – results on the asvspoof 2017 challenge," in *Interspeech*, 2017.

[10] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay Attack Detection Using DNN for Channel Discrimination," in *Interspeech*, 2017, pp. 97–101.

[11] H. Delgado, M. Todisco, Md. Sahidullah, N. Evans, T. Kinnunen, K.A Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," in *Speaker Odyssey*, 2018.

[12] B. Chettri and B. L. Sturm, "A Deeper Look at Gaussian Mixture Model Based Anti-Spoofing Systems," in *ICASSP*. IEEE, 2018.

[13] T. Kinnunen et al., "RedDots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research," in *ICASSP*. IEEE, 2017.

[14] T. Kinnunen et al., "ASVspoof 2017: Automatic speaker verification spoofing and countermeasures challenge evaluation plan.," 2017.

[15] T. Kinnunen et al., "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Interspeech*, 2017.

[16] N. Brümmer and E. D. Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[17] X. Glorot and Y. Bengio, "Understanding the difculty of training networks," in *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, vol. 9, pp. 249–256.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[19] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," Software available from tensorflow.org.

[20] B. Chettri, S. Mishra, B. L. Sturm, and E. Benetos, "A Study On Convolutional Neural Network Based End-To-End Replay Anti-Spoofing," *preprint arXiv:1805.09164 [eess.AS]*, May 2018.

[21] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[22] S. Chakroborty et al., "Improved closed set text-independent speaker identification by combining mfcc with evidence from flipped filter banks," *IJSP*, vol. 4, no. 2, pp. 114–122, 2007.

[23] S. O. Sadjadi et al., "MSR Identity Toolbox v1.0: A matlab toolbox for speaker recognition research," *Speech and Language Processing Technical Committee Newsletter*, 2013.

[24] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[25] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech.," in *Interspeech*, 2015, pp. 2062–2066.

[27] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: the ntu approach for asvspoof 2015 challenge," in *Interspeech*, 2015.

[28] K. A. Lee et al., "The RedDots Data Collection for Speaker Recognition," in *Interspeech*, 2015.