A DEEPER LOOK AT GAUSSIAN MIXTURE MODEL BASED ANTI-SPOOFING SYSTEMS

Bhusan Chettri and Bob L. Sturm

School of Electronic Engineering and Computer Science
Queen Mary University of London, United Kingdom

ABSTRACT

A "replay attack" involves replaying pre-recorded speech of an enrolled speaker to bypass an automatic speaker verification system. The 2017 ASVspoof Challenge focused on this kind of attack. In this paper, we describe our evaluation work after this challenge. First, we study the effectiveness of Gaussian Mixture Model (GMM) systems using six different hand-crafted features for detecting a replay attack. Second, we take a deeper look at these GMM systems and perform a frame-level analysis of log likelihoods. Our analysis shows how system performance can depend on a simple class-dependent cue in the dataset: initial silence frames of zeros appear in the genuine signals but missing in the spoofed version. Third, we show how we can fool these systems using this cue. For example, we find the equal error rate (EER) of one GMM system dramatically rises from 14.82 to 44.44 when we add the cue to the evaluation data. Finally, we explore whether this problem can be mitigated by pre-processing the 2017 ASVspoof Challenge dataset.

Index Terms— Gaussian mixture model, automatic speaker verification, spoofing detection, countermeasure, i-vectors, CNN.

1. INTRODUCTION

Automatic Speaker Verification (ASV) [1] systems have found increasing demand and use for voice authentication across various sectors such as security firms, banks and mobile phones [2]. However, ASV systems can be highly vulnerable to spoofing attacks [3]. There is thus a growing interest in addressing this problem [4, 5].

Four common spoofing attacks are: mimicry; text-to-speech (TTS); voice-conversion (VC); and replay. Replay attacks are perhaps the simplest kind, involving only the replaying of pre-recorded speech [3]. Fig.1 illustrates how a replay attack can be performed using genuine audio recordings. The speech recorded by an ASV system during speaker enrollment is referred as genuine speech. On the contrary, a replayed speech is one that is obtained by playing back a pre-recorded genuine speech to an ASV system.

The potential threat of replay attacks on standard and state-of-the art ASV systems is highlighted in [6]. Replay attack detection for far-field audio recordings is studied in [7, 8]. Authors in [9] perform higher-order spectral analysis to capture differences between genuine and replayed speech. The robustness of countermeasures against channel variation and unseen replay configurations using spectral landmarks is studied in [10]. Most recently, the 2017 ASVspoof Challenge focused on text-dependent replay attack detection "in the wild" [11]. This challenge featured submissions from 49 teams.

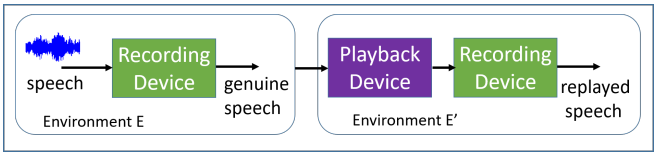

Fig. 1. Difference between a genuine and a replayed speech.

In this paper, we describe our post-evaluation analysis of our submissions to the 2017 ASVspoof challenge. We first investigate the effectiveness of six different features for the automatic detection of replayed speech using GMMs. We then analyse these systems to determine what factors influence their predictions. This helps uncover a cue that the models seem to be exploiting: several zero valued samples appear in genuine signals but are missing from the spoofed signals. We find that this cue can make the confident correct predictions of a GMM system become confident incorrect predictions. We test whether this problem with the 2017 ASVspoof Challenge dataset can be overcome by deleting the signature (initial frames of zeros) from the test files. Finally, we investigate the effect of this on utterance-based support vector machine and GMM systems trained using i-vectors and features extracted from a convolutional neural network (CNN).

2. THE ASVSPOOF 2017 CHALLENGE

Given a speech utterance s, the main goal is to build a system that determines if it is genuine speech or a recording. The text-dependent ASVspoof 2017 database is based on the *RedDots* corpus [12] and its replayed version, *RedDots Replayed* [13]. The latter was created by replaying the *RedDots* through various recording and replay configurations. Table 1 shows how the database is divided into three parts: training, development and evaluation. Performance is measured in terms of

Thanks to C4DM research group fund for providing the travel support.

Table 1. The ASVspoof 2017 Challenge database distribution between the training, development and evaluation subsets.

subset	# spkrs	# genuine	# spoofed	dur (hr)
train	10	1508	1508	2.22
dev	8	760	950	1.44
eval	24	1298	12922	11.95

a "threshold free" equal error rate (EER), which is an operating point in the detection error tradeoff (DET) curve where the false acceptance and miss rejection rate are equal.

The results of this challenge are summarized in [14]. The baseline system is a GMM trained on Constant-Q cepstral coefficients (CQCC) resulting an EER of 24.77% on the evaluation data. The best ranking system reported an EER of 6.73% on the evaluation data [15]. This system uses score-level fusion of three systems. The first is a GMM trained using features extracted from a CNN. The second is an i-vector based SVM system trained on linear prediction cepstral coefficients and the third system is an end-to-end CNN-RNN system. The second-best performing system [16], reported an EER of 10.85% on evaluation data, and uses score-level fusion of GMM systems trained on rectangular filter cepstral coefficients and linear filter cepstral coefficients. It uses higher static coefficients (30-60) augmented with delta and acceleration and performs cepstral mean normalization. Our work uses the features as in [16] but our feature parameterization and objective are completely different.

3. FRAME-LEVEL ANTI-SPOOFING

We explore the use of six different hand-crafted features: Mel-frequency cepstral coefficients (MFCC), inverted MFCC (IMFCC) [17], rectangular filter cepstral coefficients (RFCC) [18], linear filter cepstral coefficients (LFCC) [19], spectral centroid magnitude coefficients (SCMC) [20, 21] and CQCC [22]. We use the feature parameterization from [21]. All our systems use 40-dimensional features obtained by concatenating 20 delta and 20 acceleration coefficients, including energy. We do not use voice activity detection or normalisation. Our main motivation here is to study the generalization ability of GMM systems using these features on the ASVspoof 2017 database, and then to analyse the best system.

Given a speech utterance s, each system except CQCC extracts a series of Hamming-windowed frames of 20 ms duration with 50% overlap, and transforms it into a series of T feature vectors, $\mathcal{X}(s) := (\mathbf{x}_1, \ldots, \mathbf{x}_T)$. The system then computes a mean log-likelihood score by

$$\Lambda(s) := \frac{1}{|\mathcal{X}(s)|} \sum_{\mathbf{x}_t \in \mathcal{X}(s)} \log \frac{p(\mathbf{x}_t|G)}{p(\mathbf{x}_t|\neg G)} \qquad (1)$$

where $p(\mathbf{x}|G)$ is the probability density characterizing genuine speech features, and $p(\mathbf{x}|\neg G)$ is that of spoofed speech features. The larger $\Lambda(s)$ is, the more confidence the model has that s is genuine.

Table 2. Performance (EER%) of GMM systems on the development and evaluation data.

Test	IMFCC	MFCC	LFCC	RFCC	SCMC	CQCC
dev	8.5	7.17	3.33	5.15	5.46	1.51
eval	17.43	26.02	17.61	16.67	14.82	17.78

We estimate $p(\mathbf{x}|G)$ and $p(\mathbf{x}|\neg G)$ by a GMM using the expectation maximization algorithm [23, 24] on pooled training data. We find the optimal number of components for each GMM as: 512 for MFCC, LFCC and CQCC; 128 for IMFCC and RFCC; and 256 for SCMC.

Table 2 shows the results of six GMM systems on both the development and evaluation datasets. Except for the one using MFCC, all systems outperform the 24.77% baseline on the evaluation data [14] by a large margin. IMFCC features give more emphasis on high frequency information than MFCC, and seem to have more discriminability. LFCC and RFCC systems equally emphasize all frequency bands and have similar performance. Both CQCC and SCMC features show good generalizability on the replayed speech detection task, but the latter show the best result on the evaluation data. This suggests that the distribution of energy expressed by SCMC features is the most discriminative and generalizable of these six kinds of features.

4. ANALYSIS

We now take a closer look at the best GMM system which is based on SCMC features to discover the cues that influence its prediction. We look at how the log-likelihood scores for the genuine and spoofed GMM models are distributed across frames. We pick a genuine and spoofed example from the development set that the system confidently and correctly classifies: "*D_1000601.wav*" produces $\Lambda(s) = 14.66$; "*D_1001012.wav*" produces $\Lambda(s) = -0.96$. We also select the genuine signal "*D_1000300.wav*" that is confidently misclassified with a score $\Lambda(s) = -0.21$. For easy reference we define these signals as *genuine_correct*, *spoof_correct* and *genuine_incorrect*.

Figure 2 shows for each signal its spectrogram and frame-wise distribution of log-likelihoods in each model. We observe a marginal difference between genuine and spoofed model scores across frames for *genuine_incorrect* and *spoof_correct*, respectively. However, we see significantly different behavior for *genuine_correct*. The decision for this signal is dominated by its first few frames. We find that many genuine audio files in this dataset contain initial silence frames with zeros which do not appear in the spoofed version. As can be seen in Figure 2, the spoofed model assigns a very small probability to such a frame, thus pushing

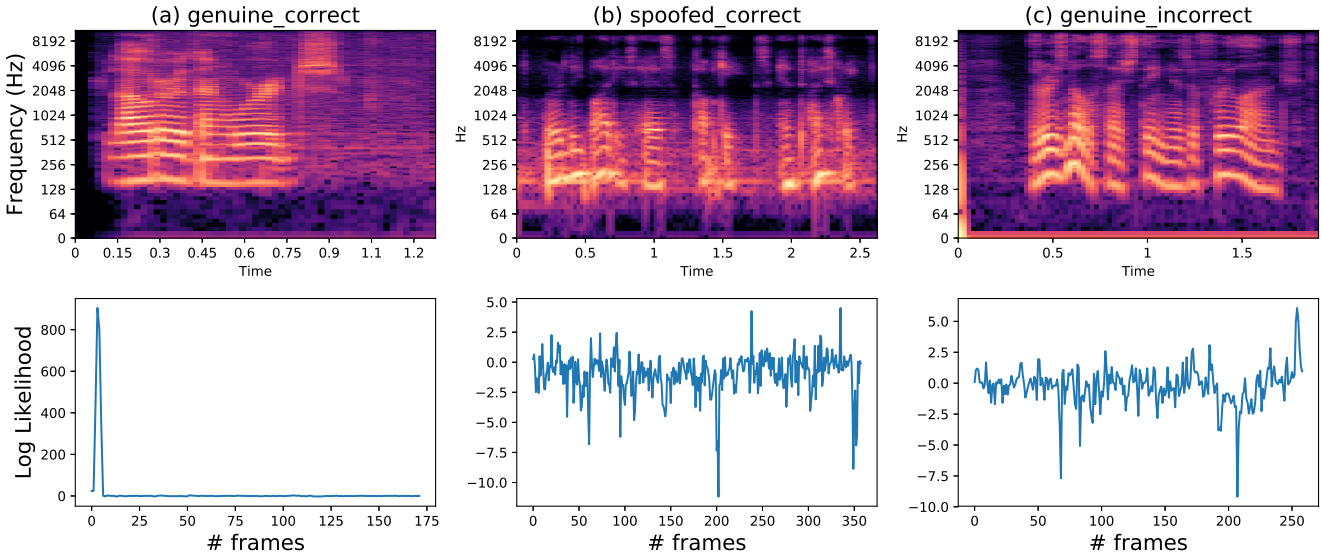

Fig. 2. Spectrograms of genuine_correct, spoofed_correct and genuine_incorrect along with frame-level log likelihood score difference between genuine and spoofed GMM.

Table 3. EER after adding the genuine signature to every utterance in the development and evaluation set.

Test	IMFCC	MFCC	LFCC	RFCC	SCMC	CQCC
dev	34.54	33.48	34.92	28.92	46.74	2.27
eval	34.46	35.95	38.23	34.22	44.44	18.71

the decision toward the genuine class. As a consequence, this has a large influence on the classifier decision (1).

We find that *genuine_correct* begins with 60ms of zeros (except four samples). Therefore, we define this 60ms segment of *genuine_correct* as a "genuine signature" and add it onto the beginning of the two other signals, *spoofed_correct* and *genuine_incorrect*. As expected, the model now scores both in favor of being genuine: $\Lambda(s) = 6.85$ and $\Lambda(s) = 11.63$ for *spoofed_correct* and *genuine_incorrect* respectively. When we repeat this process for all test files in the development and evaluation set and re-evaluate all our GMM systems we see dramatic increase in the EER of all systems except for CQCC. The IMFCC system that showed 8.5% and 17.43% EER before gives 34.54% and 34.46% EER on the development and evaluation data. We observe a similar trend for LFCC and RFCC systems. Our best performing SCMC system now gives the worst performance. We observe a very small effect on the EER for CQCC (from 17.78% to 18.81% on evaluation data) in comparison to other five features. Thus, the CQCC features that give higher frequency resolution for lower frequencies and a higher temporal resolution for higher frequencies seem to be robust against such frame-level presentation attacks.

Our analysis above casts doubt on the reliability of the evaluation results of the ASVspoof Challenge: are the other participating systems benefiting from this signature, which will not exist "in the wild"? How prevalent is this signature in the data? Can we improve the reliability of this challenge by simply deleting the first 60ms of each test audio file, and using the same trained models? Table 4 shows that removing the first 60 ms of each test audio file increases the EER of each system tested in Table 2, but not by a large amount (Approach 1). When we remove the first 60 ms of each genuine training and test audio file and retrain the genuine model (Approach 2), we also see a small increase in the EER of each system. These results suggest that the signature is not very prelavent throughout the data, but that it is prevalent enough to allow a simple means of bypassing an otherwise good performing replay attack spoofing detection system.

Table 4. EER for two cases of pre-processing. Approach1 removes first 60ms from all the test files and re-evaluates the performance. Approach2 is similar to Approach1 but here we also retrain the genuine GMM model on pre-processed training data.

System	Approach1		Approach2	
	dev	eval	dev	eval
IMFCC	8.78	19.18	8.66	19.10
MFCC	8.54	31.79	8.5	31.9
LFCC	4.01	21.46	4.41	21.06
RFCC	7.05	19.85	7.43	20.1
SCMC	6.4	17.98	6.39	17.7
CQCC	2.14	19.79	1.97	19.35

5. UTTERANCE-LEVEL ANTI-SPOOFING

The previous models we trained and tested in the 2017 ASVspoofing challenge dataset are frame-level. Will systems using utterance-level features suffer from the same vulnerability? We now investigate two models built using features learned from a convolutional neural network (CNN) and i-vectors [25]. We do not optimise these models for the best performance.

For the CNN-based features, we use the parameterization and network architecture of [15] for training the CNN with the following changes. First, we use a 300x1025 (time x frequency) log-power-normalized magnitude spectrogram as input. Second, we use a convolutional layer in place of a network-in-network layer. Third, we use 64 neurons in the fully connected layer. Fourth, we replace the max-feature-map by an exponential linear unit [26] activation and train our network. The trained network extracts 64-dimensional feature vectors from an audio file. We train genuine and spoofed GMM models using 8 mixtures. It should be noted that our input pipeline uses a preprocessing step that ensures the smallest value in the spectrogram is no less than 1e-7. Thus the network will implicitly take care of the genuine signature.

We use 40-dimensional delta-acceleration SCMC features to train a 256 mixture universal background model and total variability matrix with 200 factors on pooled data. We extract 200-dimensional i-vectors [25] for the entire dataset. We then use the training set i-vectors to train a linear support vector machine (SVM) using scikit-learn [27] with its default parameters.

Table 5. EER of utterance-based anti-spoofing systems before and after injecting the genuine signature to all the test files in the development and evaluation set.

System	**CNN features + GMM**		**i-vectors+SVM**	
	dev	**eval**	**dev**	**eval**
before	9.06	32.65	21.88	20.9
after	9.24	32.69	21.81	20.5

Table 5 shows the performance of these two utterance-level systems before and after we add the "genuine signature" to the test files. As i-vector extraction involves stacking mean vectors from the mixture components, the effect of the zero valued samples is taken care of automatically and thus we do not see any impact on performance after adding the genuine signature. Similarly, CNN has a max-pooling layer that choses a maximum from a given block of convolved input, thus the artefacts are taken care of in the first convolutional layer, thereby eliminating the impact of genuine signature on the predictions. As expected, the experimental results in Table 5 clearly indicate that systems trained on utterance-based fixed length feature representations in the 2017 ASVspoof Challenge dataset are resilient against such frame-level presentation attacks.

6. DISCUSSION

In our work, the SCMC-feature based GMM system showed the best performance on ASVspoof 2017 challenge dataset. Deeper analysis of this system led us to interesting observations. We find the presence of recording artefacts (initial silence frames contain zeros) in some genuine audio files in the dataset that is missing from the replayed version. As a consequence spoofed models assign a very low likelihood to such frames during testing. We demonstrate how knowledge of such cues can compromise system predictions. Though such data-intrinsic behavior may not appear in real-world scenarios our work shows how severe impact it can have on the EER for frame-level GMM systems. We investigated two intervention approaches to help mitigate against such manipulation attacks. Comparing Table 3 and Table 4 we see that our proposed approaches helped reduce the error-rate of all the systems. Section 5 shows two utterance-level anti-spoofing systems that do not suffer from such manipulation. A bigger question we have yet to answer is what is causing the large difference between the EER on the development and evaluation datasets.

7. CONCLUSION

In this paper, we investigated the generalizability of different features for the automatic detection of replay spoofing on the ASVspoof 2017 challenge dataset. Our frame-level analysis shows how class-dependent cues in the dataset can lead to the manipulation of class predictions. We find that frame-level systems are highly vulnerable against such manipulation attacks except the CQCC. Our proposed solutions help mitigate the problem effectively. Further, as a proof-of-concept we showed that utterance-level feature-based systems are resilient to such manipulations. Our future work aims to perform more in-depth analysis on the dataset and investigate neural network architectures for learning robust features for replay detection.

8. 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., "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *APSIPA*. IEEE, 2014, pp. 1–5.

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

[5] Z. Wu et al., "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE JSTSP Special Issue on Spoofing and Countermeasures for Automatic Speaker Verification*, 2017.

[6] 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.

[7] J. Villalba and E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems," in *BioID*. Springer Berlin Heidelberg, 2011, pp. 274–285.

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

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

[10] J. Gaka, M. Grzywacz, and R. Samborski, "Playback attack detection for text-dependent speaker verification over telephone channels," *Speech Communication*, vol. 67, pp. 143 – 153, 2015.

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

[12] K. A.and others Lee, "The RedDots data collection for speaker recognition," in *INTERSPEECH*, 2015.

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

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

[15] G. Lavrentyeva et al., "Audio replay attack detection with deep learning frameworks," *Proc. Interspeech 2017*, pp. 82–86, 2017.

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

[17] 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.

[18] T. Hasan et al., "CRSS systems for 2012 nist speaker recognition evaluation," in *ICASSP*, 2013, pp. 6783–6787.

[19] F. Alegre, A. Amehraye, and N. Evans, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," in *BTAS*, Sept 2013, pp. 1–8.

[20] J. M. K. Kua, T. Thiruvaran, M. Nosratighods, E. Ambikairajah, and J. Epps, "Investigation of spectral centroid magnitude and frequency for speaker recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2010, pp. 34–39.

[21] M Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Interspeech 2015*, 2015.

[22] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q Cepstral Coefficients," 2016.

[23] C. M. Bishop, "Pattern recognition and machine learning," 2006.

[24] 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.

[25] 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.

[26] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.

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