Audio Replay Attack Detection in Automated Speaker Verification

Pooja Anjee
Rashtreeya Vidyalaya College of Engineering
Karnataka, India

Shubham Ghosh
Rashtreeya Vidyalaya College of Engineering
Karnataka, India

Shrirag Kodoor
Rashtreeya Vidyalaya College of Engineering
Karnataka, India

Rajashree Shettar
PhD, Rashtreeya Vidyalaya College of Engineering
Karnataka, India

ABSTRACT
Automated Speaker Verification (ASV) systems are extensively used for authentication and verification measures. Countermeasures are developed for ASV systems to protect it from audio replay attacks. This paper describes the ASVspoof2017 database, conceptual analysis of various algorithms and their classification followed by prediction of results. Feature extraction is based on the recently introduced Constant Q Transform (CQT), a perceptually mapped frequency-time analysis tool mainly used with audio samples. The training dataset comprises of 1508 genuine samples and 1508 spoof samples. A training accuracy of 84.4% is achieved for variations of boosted decision tree. Parameters such as learning rate, number of learners and splits were empirically optimized. LogitBoost was found to have outperformed AdaBoost in all metrics. Furthermore, an implementation of a single hidden layer neural network achieved a training accuracy of 92.1%. A comparison of the algorithms revealed that while the neural network achieved a higher overall training accuracy, it had a lower True Negative Rate than LogitBoost. Overall, the paper describes a generalized system capable to detection of replay attacks in known and unknown conditions.

General Terms
Automated Speaker Verification, Neural networks, AdaBoost and LogitBoost, Constant Q Transform, Feature Extraction

Keywords
Replay attack detection, Automated speaker verification, Classification of Speech Samples

1. INTRODUCTION
Automatic Speaker Verification (ASV) systems [1] are widely used as authentication measures. However they show vulnerability to spoofing attacks (impersonation of a system user). There exist a variety of different spoofing methods which may be used to exploit these vulnerabilities. Finding countermeasures to their exploits is vital for their continued use in commercial systems.

When implementing automatic speech verification in real time applications robustness, security plays an important role. These include major scenarios which are likely to affect major applications such as mobile transaction system, phone authentication process and 911 emergency calls [2]. Hence it is essential to know the robustness of the automatic speaker verification (ASV) against spoofed attacks.

ASVspoof 2017 is a challenge for the purpose of developing countermeasures to these spoofing attacks. This edition of the challenge places emphasis on replay attacks**[**3],which are a core threat and can be easily performed. Replay attacks are conducted by using a recording of a speaker's voice which are replayed to the ASV system. An example of such an attack would be using a device to replay a recording of a speaker's voice to unlock a Smartphone which uses ASV for access control.

Countermeasures have been developed to protect ASV systems from replay attacks. There exist three general strategies [2]. *Prompted-phrase ASV*, e.g. randomised digit sequences, and utterance verification offers some protection, but are vulnerable to replay attacks produced by remixing several recordings. *Copy detection*, or *audio fingerprinting* can also be used to detect recordings of genuine enrolment utterances or previous access attempts although this approach calls for the maintenance of a dynamically growing database. The challenge seeks to address a third strategy involving detection of replay attacks using only the acoustic characteristics of the given speech audio [2].

The difficulty of identification by using the acoustic characteristics is caused by variation in the quality of a replay attack. Certain low quality recordings may contain significant noise. The problem of detection may then boil down to noise classification. On the other hand, high quality recordings may be such that they are nearly identical to genuine speech samples owing to negligible amounts of noise or distortion. As a result, it may be nearly impossible to distinguish from genuine speech in case of high quality recordings.

Existing studies attempting to assess replay attacks and their countermeasures generally involve a modest number of evaluation conditions[4]. Studies have reported nearly perfect accuracy in cases relatively homogeneous acoustic conditions. Some other works suggest that performance may depend on the acoustic conditions of the replay attack and is likely to degrade in real-world situations [4].

The primary goals of the challenge are therefore twofold: (i) to assess the practical limitations of replay attack detection and (ii) to promote the development of countermeasures.

This paper will describe a system developed for the purpose of this challenge. The system uses Constant Q Cepstral Coefficients [3] and Constant Q Transform spectrograms as features [5]. A comparison of the use of variations of Boosted Decision Trees and neural networks as classifiers is performed. Obtained results are tabulated and discussed. Finally, experimental results and conclusions are presented.

2. PRIOR/RELEVANT WORK

This section will describe the countermeasures to attacks on ASV system such as impersonating the owner.

2.1 ASVspoof 2015 Challenge

In systems developed prior to the commencement of the challenge, both countermeasures and spoofing attacks were developed with detailed information regarding particular speaker verification systems. Specific spoofing attack that are meant to be detected are developed in such a way the countermeasures to the attacks can be attained. In reality, it is highly unlikely that prior information of spoofing attacks and verification systems is available [6]. The ASVspoof challenge is intended as a call for development of generalised countermeasures against spoofing attacks.

3. ASVSPOOF 2017 CORPUS

This is a dataset corpus that is provided by the ASV spoof 2017 challenge. The ASV spoof 2017 Corpus is developed using the Red Dots corpus which was collected by representatives (ASV researchers) in FUB (Italy),UEF (Finland),EUR (France)and AAU (Denmark). It is a publicly available corpus inclusive of replay attacks [7].

The system consists of multiple speakers recorded for the first time using two android smart phones and a portable device such as a laptop. The replay utterances were then captured by smart-phones. Replayed signals are generated using high quality loudspeaker and the built-in speaker of laptop.

The genuine non-spoofed utterances are a portion of the underived RedDots [7] recordings, while on the contrary spoofed recordings are replayed versions. Spoofed samples which are replayed versions are representatives of a replay attack scenario. Here, the attacker would have an access to a digital copy of an original target speaker utterance which is then replayed using high quality transducers which can be used to authenticate through devices.

It comprises of the training set, development set and evaluation set. Speaker ID, phrase ID, and replay configuration details were provided for the training and development subsets. Only audio data and phrase ID were provided for the evaluation set for which participants were required to submit scores.

4. METHODOLOGY

This section describes the methodology of the system, including features extracted and classifiers used Figure 1 shows the system architecture comprising of pre-processing, feature extraction, classification and prediction.

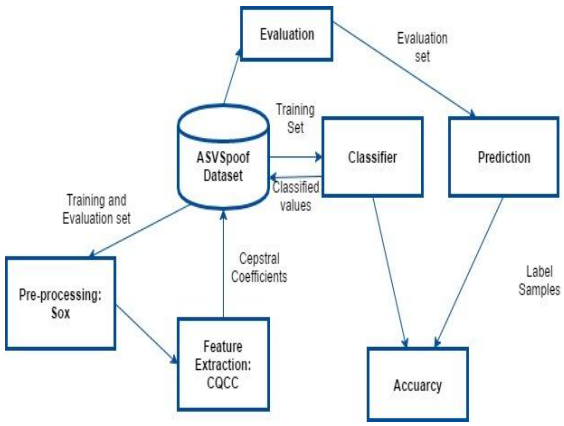

Figure 1: System Architecture depicting flow of methods used.

Pre-processing is the first step to ASV systems and the next step is feature extraction. The features obtained are classified using various classifiers to genuine and spoofed samples. Training and Evaluation accuracy are computed.

4.1. Pre-Processing

Pre-processing of audio files is regarded as a significant step in Automated Speaker Verification Systems. It is performed so that no time is wasted in processing non-speech intervals and to avoid processing very long audio chunks. All the audio files have varied lengths ranging from 0.4 seconds to 10 seconds. These files are manipulated to attain a uniform length of 2 seconds. Samples with length less than 2 seconds are retained. Samples with length greater than 2 seconds are trimmed (using a manipulation tool called SoX [8]). Uniform length samples produce efficient results for feature extraction.

4.2. Feature Extraction

The feature is defined and is in view of constant Q transform algorithm. It deals with the extraction of cepstral coefficients and cepstral analysis. [3] The cepstral coefficients (CQCC) allows provides a time frequency representation of the spectrum. This allows for the capture of characteristics which are missed by other approaches. This feature has been shown to have a marked improvement over existing methods of spoofing attack detection.

This proposed system accepts CQCC feature along with acceleration coefficients 'A' and zeroth order coefficients of the audio file. A maximum frequency of F_{max}= 8kHz and minimum frequency of $F_{min} = F_{max}/2^{10}$is applied on the CQT and the number of bins per octave B is taken as 96 which is considered as a standard value and it generates the CQCC feature values accordingly [5].

Equation (1) computes the CQT values by changing few parameters below

$$X^n = \sum_{j=n-\left|\frac{n_k}{2}\right|}^{n+\left|\frac{n_k}{2}\right|} x(j)a_k(j - n + \frac{N_k}{2}) \quad (1)$$

Where, a $*$ k(n) is the conjugate of a_k(n) and N_k is known as window length. The value k ranges from 1, 2,... till k.

Equation (2) computes the time frequency of atoms and gives corresponding values. The a_k(n) is defined according to:

$$a_k(n) = \frac{1}{C}\left(\frac{n}{N_k}\right)\exp\left[i\left(2\pi n\frac{f_k}{f_s} + \varphi_k\right)\right] \quad (2)$$

Where the bin k centre frequency is f_k, f_s is known as the sampling rate, and function window as w_k. Φk is a phase offset. Equation(3) gives the scaling factor C:

$$C = \sum_{l=-\left|\frac{N_k}{2}\right|}^{\left|\frac{N_k}{2}\right|} w\left(\frac{1+\frac{N_k}{2}}{N_k}\right) \quad (3)$$

The centre frequencies f_k given by equation (4):

$$f_k = f_1 2^{\frac{k-1}{B}} \quad (4)$$

Where the centre frequency of the lowest-frequency bin is $f_1$and number of bins per octave is denoted by B.

The Q factor is then given by equation (5):

$$Q = \frac{f_k}{f_{k+1} - f_k} = \left(2^{\frac{1}{B}} - 1\right)^{-1} \quad (5)$$

The window lengths $N_k \in$ R in equations 1 and 2 are real-valued and inversely proportional to f_k, and Q is constant for all frequency bins k, is as given in equation (6):

$$N_k = \frac{f_s}{f_k} \quad (6)$$

4.3 Gradient Boosting

It is a method, which delivers a prediction model as a group of weak prediction models. Earlier work has introduced the dynamic perspective of boosting algorithms as iterative useful gradient descent algorithms i.e. it optimizes a cost function by iteratively picking a weak hypothesis that focuses on the negative gradient direction. This perspective of boosting has led to the development of boosting algorithms in numerous zones of machine learning and statistics.

4.3.1 AdaBoost: AdaBoost [9] is a meta-algorithm introduced by Freund and Schapire and alludes to a specific strategy for training a classifier. It is used together with weak learners to enhance their execution. The result of the classifier is taken from the results given by the weak learners which was together is taken as a weighted sum with ensuing weak learners being tweaked in favour of those instances incorrectly predicted by previous classifiers. For certain classification tasks, it may be more resistant to over fitting other classifiers. Even though AdaBoost uses individual weak learners, if the performance of each weak learner is more than that of random guessing, the model can be proven to focalize to a strong learner. The system uses decision trees as the weak learners.

4.3.2 LogitBoost: LogitBoost [10] is a combination of logistic regression techniques and the AdaBoost algorithm. LogitBoost and AdaBoost are similar in the sense that both perform an additive logistic regression but differ in that AdaBoost minimizes the exponential loss, whereas LogitBoost minimizes the logistic loss. The system uses the same parameters as AdaBoost for LogitBoost.

4.3.2 Neural Networks: A feed-forward two-layer network[11] with sigmoid linear output neurons and hidden neurons is used as shown in figure 2. Multi-dimensional mapping problems are arbitrarily well fit, given steady information(data) and sufficient neurons in its hidden layer. The system uses 10 neurons (empirically obtained) in the hidden layer. The algorithm used to train the network is Levenberg-Marquardt back propagation algorithm. Scaled conjugate gradient back propagation will be used if there is not enough memory (in cases of different, larger datasets).

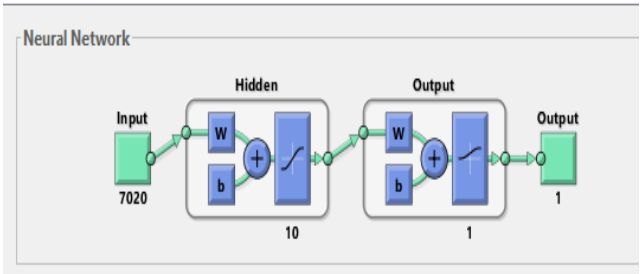

Figure 2: A two layer feed forward network with one hidden layer

5. EXPERIMENTAL RESULTS

The classification process goes through two stages. Classifier is trained and validated using a dataset comprising of 1508 genuine samples and 1508 spoof samples. The training accuracy for variations of boosted decision trees is tabulated. Input parameters such as learning rate, number of learners and splits are changed and the best result is recorded. A fivefold cross validation is used. Table 1 depicts training accuracy obtained after executing the AdaBoost, LogitBoost and Levenberg-Marquardt algorithm.

A good classifier is obtained when the input parameters are correctly set. On varying the parameters , learning rate of 0.05 and number of splits as 20 produced the best results. On increasing the learning rate, overfitting of the curve took place, hence it resulted in lower accuracy results.

Table 1: Observations for training dataset(genuine taken as positive class and spoof as negative class)

Algorithm	Training Accuracy	TPR	FPR	TNR
AdaBoost	82.6%	0.93	0.25	0.925
LogitBoost	84.4%	0.93	0.24	0.927
Neural Network	92.1%	0.973	0.131	0.88

It can be noted from table 1 that LogitBoost has a higher training accuracy than AdaBoost. Neural Network has a higher training accuracy than both variations of boosted trees.

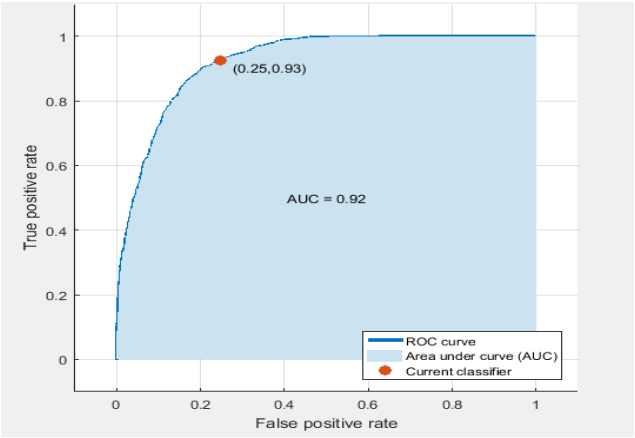

Figure 3: ROC curve of training for AdaBoost

Receiver Operating Characteristic (ROC)curve for AdaBoost is shown in Figure 3, and can be inferred that Sensitivity is 0.93 and Specificity is 0.925 for training dataset. An AUC (Area Under Curve) of 0.93 suggests the training accuracy of AdaBoost is considered to be excellent.

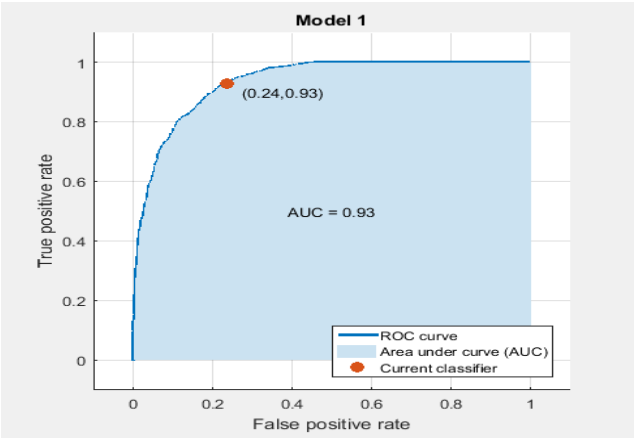

Figure 4: ROC curve of training for LogitBoost

Figure 4 shows ROC curve for LogitBoost can be inferred that Sensitivity is 0.93 and Specificity is 0.927 for training dataset. An AUC (Area Under Curve) of 0.93 suggests the training accuracy of AdaBoost is considered to be excellent.

Table 2: Observations for evaluation dataset

Algorithm	Evaluation Accuracy	TPR	FPR	TNR
AdaBoost	70.58%	0.50	0.129	0.865
LogitBoost	72.11%	0.532	0.128	0.871
Neural Network	75.9 %	0.852	0.34	0.725

It can be noted from table 2 that LogitBoost has a TNR of 0.871. This implies 87.1% of the spoof samples have been correctly identified. AdaBoost has similar statistics when compared to LogitBoost. On the other hand Neural Network has a lower true negative rate with a higher true positive rate. Which means 85.2% of the genuine samples have been identified as genuine samples. Hence, a combination of both algorithms would result in an optimized speaker verification system.

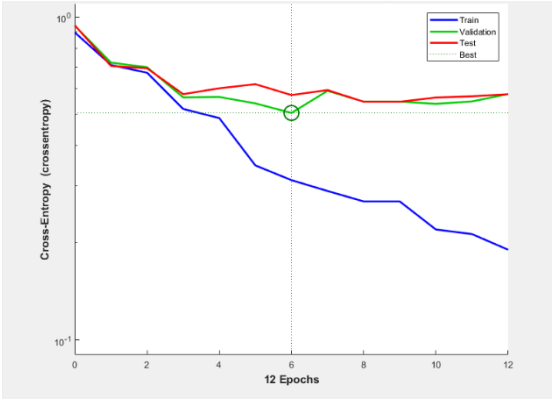

Figure 5: Performance graph of neural network

Figure 5 shows the performance graph of training, validation and evaluation of pattern recognition using neural networks. The algorithm used is Levengerb-Marquardt back propagation with a single hidden layer.

As the number of epochs increases, the loss function can be seen to decrease. This indicates that the performance of the classifier improves with each iteration. As seen in the graph, the cross-entropy loss function begins to converge at 12 epochs. At this point, there is no significant performance increase for further iterations and so the training process is halted.

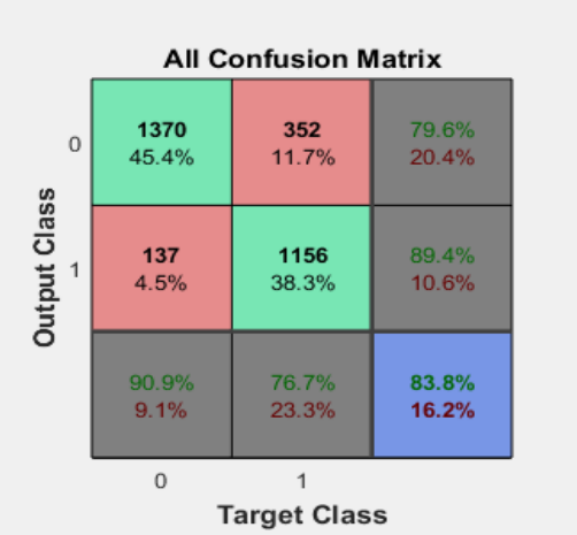

Figure 6: Confusion Matrix for Training, Validation and Test dataset in neural networks.

From the above figure we can see the overall metrics for training, validation and test data set in neural networks. It can be inferred that 2527 samples have been correctly identified as genuine or spoof.

From the training and evaluation results obtained, neural networks show a higher overall training accuracy. However, LogitBoost provides a higher TNR. In a spoof detection system for an ASV, a spoofed sample classified as genuine would be a critical security issue but a genuine sample classified as spoofed would merely be an annoyance. Therefore, it can be argued that a greater TNR is of greater importance for a spoof detection system.

6. CONCLUSION

In this paper, countermeasure against audio replay attack by classification of speech samples has been presented. A new feature for the detection of spoofing attacks termed as CQCC is introduced. Most classical feature extraction approaches miss certain detailed characteristics which are captured by CQCC. The features obtained are classified using two variations of boosted decision trees, AdaBoost and LogitBoostas well as a shallow neural network comprising of one hidden layer with ten neurons.Various input parameters such as number of learners, number of splits and learning rate are changed for the optimization of decision trees. Results are tabulated for the two classifiers. It can be observed that LogitBoost provides better accuracy over AdaBoost and Neural Network for the evaluation dataset.A significant amount of future enhancements is possible. Primarily, a better performed algorithm may be used and optimized, such as the use of Deep Neural Networks. With respect to the dataset, the classifiers may be trained on audio samples recorded from a greater number of sources with different environmental conditions. Different feature extraction methods may be used in addition to use of more classifiers, to improve classification accuracy. The spoof detection module may be integrating all the web interfaces together into a single unified interface for result visualization.

7. REFERENCES

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," Speech Communication, vol. 66, no. 0, pp. 130 – 153, 2015.

[2] D. Paul, M. Sahidullah, and G. Saha, "Generalization of spoofing countermeasures: A case study with ASVspoof 2015 and BTAS 2016 corpora," in Proc. ICASSP, 2016.

[3] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilci, M. Sahidullah, A. Sizov, K. A. Lee, M. Lee, H. Delgado: "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in INTERSPEECH, Sweden, 2017 (pending).

[4] Md. Sahidullah, T. Kinnunen, and C. Hanilci, "A comparison of features for synthetic speech detection," in INTERSPEECH, Sweden, 2015, pp. 2087–2091.

[5] H. Delgado, M. Todisco, M. Sahidullah, A. Sarkar, N. Evans, T. Kinnunen, and Z.-H. Tan, "Further optimisations of constant Q cepstral processing for integrated utterance verification and text- dependent speaker verification," in Proc. IEEE Spoken Language Technology Workshop (SLT), 2016, pp. 179–185.

[6] X. Xiao, X. Tian, S. Du, H. Xu, E. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: the NTU approach for ASVspoof 2015 challenge," in INTERSPEECH, Sweden, 2015

[7] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in Speaker Odyssey Workshop, Bilbao, Spain, 2016.

[8] SoX, audio manipulation tool, (accessed Jan 25, 2015). [Online]. Available: http://sox.sourceforge.net/

[9] Jerome Friedman, Trevor Hastie and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 2000. 337–407.

[10] Freund., Schapire.:"A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences. 55: 119

[11] Santaji Ghorpade, Jayshree Ghorpade and Shamla Mantri:"Pattern recognition using neural networks".International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010.

[12] F. Alegre, A. Amehraye, and N. Evans, "Spoofing countermeasures to protect automatic speaker verification from voice conversion," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 3068–3072.

[13] F. Alegre, A. Janicki, and N. Evans, "Re-assessing the threat of replay spoofing attacks against automatic speaker verification," in Proc. Int. Conf. of the Biometrics Special Interest Group (BIOSIG), 2014.

[14] J. Brown, "Calculation of a constant Q spectral transform," The Journal of the Acoustical Society of America, vol. 89, no. 1, pp. 425–434, 1991.

[15] N. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in Proc. INTERSPEECH, Lyon, France, 2013.

[16] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," Speech Communication, vol. 85, pp. 43–52, 2016.

[17] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin, "STC anti-spoofing systems for the asvspoof 2015 challenge," in INTERSPEECH, 2015.

[18] 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.

[19] Villalba E., Lleida E., "Speaker verification performance degradation against spoofing and tampering attacks", in Proc. of the FALA 2010 Workshop, pp. 131–134, 2010.

[20] Z. Wu, S. Gao, E. Chng, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in Proc. APSIPA, 2014, pp. 1–5.

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