INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Transfer-Representation Learning for Detecting Spoofing Attacks with Converted and Synthesized Speech in Automatic Speaker Verification System

Su-Yu Chang, Kai-Cheng Wu, Chia-Ping Chen

National Sun Yat-sen University, Taiwan

cpchen@mail.cse.nsysu.edu.tw, suyuzhang@g-mail.nsysu.edu.tw

Abstract

In this paper, we study a countermeasure module to detect spoofing attacks with converted or synthesized speech in tandem automatic speaker verification (ASV). Our approach integrates representation learning and transfer learning methods. For representation learning, good embedding network functions are learned from audio signals with the goal to distinguish different types of spoofing attacks. For transfer learning, the embedding network functions are used to initialize fine-tuning networks. We experiment well-known neural network architectures and front-end raw features to diversify and strengthen the information source for embedding. We participate in the 2019 Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2019) and evaluate the proposed methods with the logical access condition tasks for detecting converted speech and synthesized speech. On the ASVspoof 2019 development set, our best single system achieves a minimum tandem decision cost function of nearly 0 during system development. On the ASVspoof 2019 evaluation set, our primary system achieves a minimum tandem decision cost of 0.1791, and an equal error rate (EER) of 9.08%. Our system does not have over-training issue as it achieves decent performance with unseen test data of the types presented in training, yet the generalization gap is not small with mismatched test data types.

Index Terms: automatic speaker verification, spoofing detection, logical access, representation learning

1. Introduction

An ideal automatic speaker verification system (ASV) should accept the claim from a true speaker and reject the claim from an imposter. An ASV system could achieve EER of 6% for text-independent [1] tasks and 2% for text-dependent [2] tasks. However, ASV systems may be attacked by spoofing data which leads to false accept of imposters. For example, there are text-to-speech [3, 4] (TTS), voice conversion [5, 6] (VC), cut-and-paste [7] and replay attacks. Today, the fast progress of TTS and VC technology really posts genuine threats to ASV systems.

To deal with the issue of spoofing attacks, ASVspoof evaluations have been held in recent years. ASVspoof 2013 [8] was intended to raise awareness of the spoofing problem. ASVspoof 2015 [9] focused on the design of countermeasure solutions capable of discriminating between bona fide speech and spoofing speech produced using either TTS or VC systems. ASVspoof 2017 [10] added a new perspective in this challenge, i.e. audio replay attacks. ASVspoof 2019 includes all three major attack types, namely those stemming from the up-to-date TTS, VC and replay spoofing attacks. Specifically, ASVspoof 2019 divides the attacks into logical access (LA) and physical access (PA). The LA task focuses on the detection of TTS and VC attacks, and the PA task focuses on the detection of replay attacks. In this study, we focus on the LA task.

Our approach to ASVspoof 2019 LA task is to develop embedding-based system for spoofing detection. Embedding-based methods have been applied in classification and recognition tasks, e.g. language models [11], face recognition [12], image classification [13] and speaker recognition [14]. Specifically, the Inception model [15] based on convolutional neural networks (CNN) has been proposed to learn good representation of images for face recognition with great success. The X-vector embedding method based on time-delayed neural network (TDNN) structure [16] has been successful in the NIST speaker recognition evaluation [17]. If a good frontend embedding vector extractor is learned, the job of a backend classifier becomes easy. This is why we choose embedding approach to spoofing detection.

We develop systems with a front-end embedding extractor and a backend classifier, both of them need to be learned from data. The embedding extractor extracts embedding vector from raw speech features through a neural network. For the raw speech features, we assess the constant-Q cepstral coefficients [18] (CQCCs) and the linear-frequency cepstral coefficients (LFCCs) to analyze speech from multiple aspects. For the embedding networks, we investigate three different network architectures. More details will be provided in later sections.

The remainder of this paper is organized as follows. In Section 2, we introduce our system including acoustic feature extraction, representation learning and back-end classifier. In Section 3, we describe experimental setup of ASVspoof 2019 evaluation. In Section 4, we present the results on ASVspoof 2019 evaluation and our observations. In Section 5, we draw conclusion to this work.

2. System Description

A block diagram of the proposed approach to counter spoofing attacks is shown in Figure 1. A feature extraction module extracts acoustic features from input audio signal. A representation learning module learns neural network-based embedding functions via a 7-class classification task. A fine-tuning network fine-tunes the embedding functions via a 2-class classification (detection) task. A back-end classifier output a detection score regarding whether the speech is a spoofing attack or not.

2.1. Acoustic Feature Extraction

We use constant-Q cepstral coefficients (CQCCs) and linear-frequency cepstral coefficients (LFCCs) as raw speech features. CQCCs are features based on constant-Q transform (CQT). It ensures that all frequency domain information in the spectrum can be effectively preserved. We use 30 CQCCs and their first and second derivatives, resulting in 90-dimensional feature vectors. LFCCs are features based on filters equally spaced in the linear frequency domain. We use 30 LFCCs and their first and second derivatives, also resulting in 90-dimensional feature vec-

http://dx.doi.org/10.21437/Interspeech.2019-2014

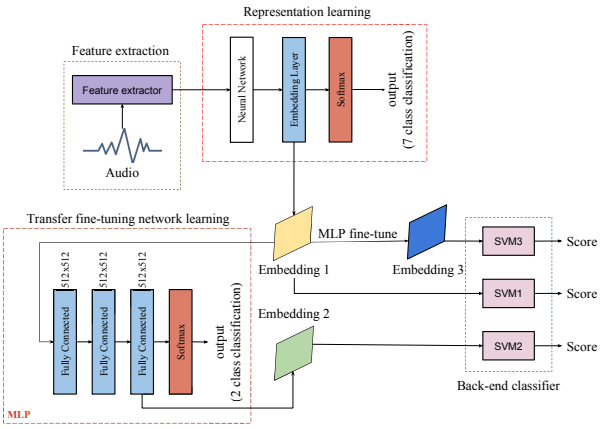
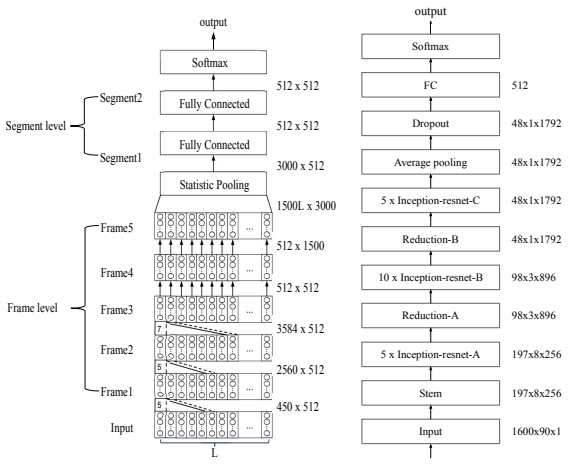


Figure 1: *Block diagram of our spoofing detection system.*



(a) X-vector (b) Inception-ResNet-v1

Figure 2: *Network architectures of (a) TDNN-based X-vector and (b) CNN-based Inception-ResNet-v1.*

tors. The audio samples are coded with a 25-ms window and 10-ms frame shift. We convert every utterance to a fixed-length utterance with 1600 frames by cropping or padding zeros.

2.2. Representation Learning

In this subsection, the neural network structures that we use for learning representation for detecting spoofing audios are described.

2.2.1. TDNN-based X-vector

We use X-vector structure based on time delay neural network (TDNN). The block diagram for X-vector is shown in Figure 2(a). The first 4 hidden layers operate at frame level, while the last 2 layers operate at segment level. At the end of frame level, a statistic pooling layer computes the mean and variance from frame-level information to segment-level information. The mean and variance are then concatenated together and propagated through segment-level layers to the softmax output layer. Each hidden layer is followed by batch normalization and ReLU activation function. The embedding is extracted from the 512-dimensional components of the first segment-level layer.

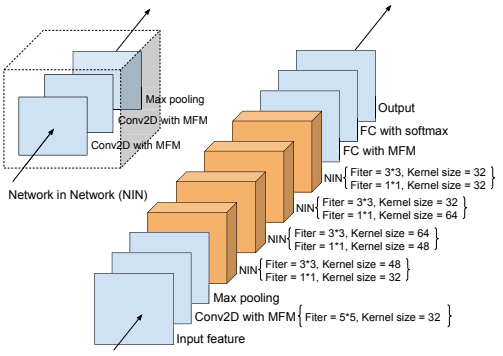


Figure 3: *There are four Networks in Network (NIN) in our implemented LCNN architecture. An NIN includes two convolution layers with MFM and one max pooling layer.*

2.2.2. CNN-based Embedding

We use Inception-ResNet-v1 [15] based on convolutional neural networks (CNN). Inception-ResNet-v1 is the state-of-the-art structure for image classification and face recognition tasks. Figure 2(b) is a simplified block diagram of Inception-ResNet-v1. We only tune one hyper-parameter, which is the embedding size controlled by the last fully connected layer. 512-dimensional embedding vectors are extracted at the final fully connected layer.

We also use light CNN (LCNN) which achieves good performance in the ASVspoof 2017 Challenge [19]. LCNN architecture is shown in Figure 3. It contains 5 convolution layers, 4 network-in-network (NIN) layers, max-feature-map (MFM) layers and 4 max-pooling layers. LCNN uses MFM as a variation of max-out activation into each convolutional layer of CNN. The NIN can do feature selection between convolution layers with MFM, and reduce the number of parameters by using small convolution kernels. The overall representation is thus regularized by NIN and tends to be more robust and effective. The vector formed by flattening the last NIN output is extracted as the embedding.

2.3. Transfer Fine-tuning Network Learning

We apply transfer learning in our spoofing detection system. We implement transfer learning by replacing and retraining the last few layers of the network. This allows learned hidden units in the bottom layers to be reused in a refined network. Specifically, we replace the last few layers by 3 hidden layers of multilayer perceptron [20] (MLP). As shown in Figure 1, each hidden layer in the MLP network structure has 512 hidden units.

2.4. Classifier

For final detection, we use support vector machines [21] (SVM) with linear kernels. When the proposed embedding learning process is completed, the embedding vectors are used in training classifiers to detect bona fide speech and spoofing attacks.

3. Experiments

We use the data officially released for ASVspoof 2019 Challenge. The train and development sets contain 6 spoofing attacks of types. Subsets A1–A4 are speech synthesis methods, while subsets A5–A6 are voice conversion algorithms.

Figure 4: *The t-SNE visualization of the embeddings extracted from the X-vector embedding layer on 2019 ASV development dataset.*

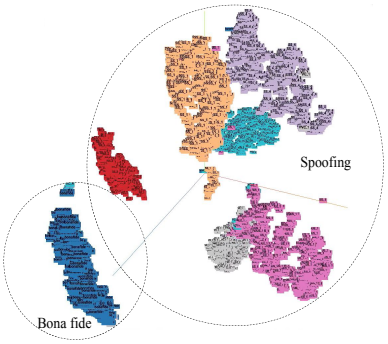


Figure 5: *The t-SNE visualization of the embeddings extracted from the MLP embedding layer on 2019 ASV development dataset.*

In the representation-learning stage, the neural network is trained to predict for 7 classes of A1–A6 spoofing data and bona fide speech. We use the t-distributed stochastic neighbor embedding [22] (t-SNE) to visualize the high-dimensional embeddings learned from this stage, as shown in Figure 4. We can see that embeddings are divided into 7 groups according to different data conditions.

In the transfer-learning stage, we redefined the neural network to predict for just 2 classes, namely the bona fide speech class and spoofing data class. We also apply the t-SNE to visualize the high-dimensional embeddings learned from this stage, as shown in Figure 5. We can see that embeddings are divided into 2 groups according to bona fide speech and spoofing speech. Therefore we can efficiently use simple SVM to distinguish between bona fide speech and spoofing speech.

4. Results of Evaluation

The results evaluated on the 2019 ASV development set and evaluation set are reported in terms of equal error rate (EER) and the minimum tandem detection cost function (t-DCF). According to ASVspoof 2019 Evaluation plan [23], the prior probabilities in tandem ASV system are

$$p_{\text{target}} = 0.9405,\ p_{\text{non-target}} = 0.0095,\ p_{\text{spoof}} = 0.05$$

The baseline systems have front-end features of LFCCs and CQCCs, and backend classifiers based on Gaussian mixture models (GMM).

There are 13 spoofing data types in ASVspoof 2019 evaluation set. Subsets A7–A12 and A16 are speech synthesis methods, subsets A17–A19 are voice conversion algorithms, and

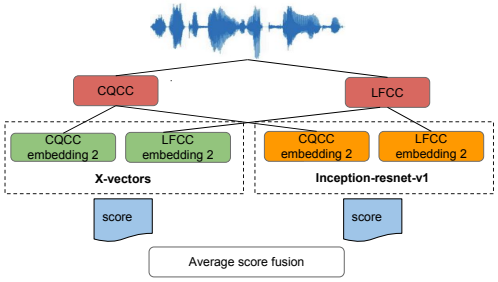


Figure 6: *The average score fusion in primary system.*

subsets A13–A15 are mixed speech synthesis and voice conversion spoofing attacks.

4.1. Overall Results

We evaluate three embedding-based systems on the evaluation data. They are the primary system, single system and contrastive system. The evaluation results are shown in Table 1.

The single system combines X-vector embeddings from CQCCs and LFCCs front-end features. They are both extracted from the MLP last fully connected layer in the transfer fine-tuning network learning stage. This single system can achieve close to 0 t-DCF on the development set.

The contrastive system adopts LCNN to extract embeddings from CQCCs and LFCCs front-end features to train the MLP classifier and fine-tune the whole network simultaneously. On the evaluation data, this system has higher EER but lower t-DCF than other systems.

The TDNN-based X-vector and CNN-based LCNN achieves similar performance with baseline systems, but there is still room for improvement. We have applied early stopping in training to avoid overfitting. However, the same dataset has been used in all of the training stages, including representation learning, transfer fine-tuning network learning, and the classification. To some degree, this may have caused the system to overfit to the used dataset. The same data appeared too many times in the construction of the detection system. If the training data is split into different parts during training, it may be possible to reduce the generalization gap.

Our primary system uses simple average score fusion. It averages the above single system score and the Inception-ResNet-v1 score, as shown in Figure 6. We consider use the characteristics of CNN with space expansion which is good at extracting position invariant features, and characteristics of RNN with time expansion which can describes the output of continuous state over time with memory function. These features can help us estimate the scores from different aspects to improve the performance. The idea achieves a t-DCF of nearly 0 on system development set and has lower t-DCF than baseline systems on the evaluation set.

4.2. Detailed Results on Evaluation Data

The detailed results with evaluation data created by different spoofing algorithms are shown in Table 2. We achieve good performance on the mixed spoofing attack methods (A13–A15). This attack type is clearly unseen in both the training and development dataset. Figure 7 shows the minimum t-DCF results of all participated primary systems on the mixed spoofing attacks. Here, the performance of our primary system is relatively good. Specifically, it achieves 0.0282 t-DCF for A14 attack type. This

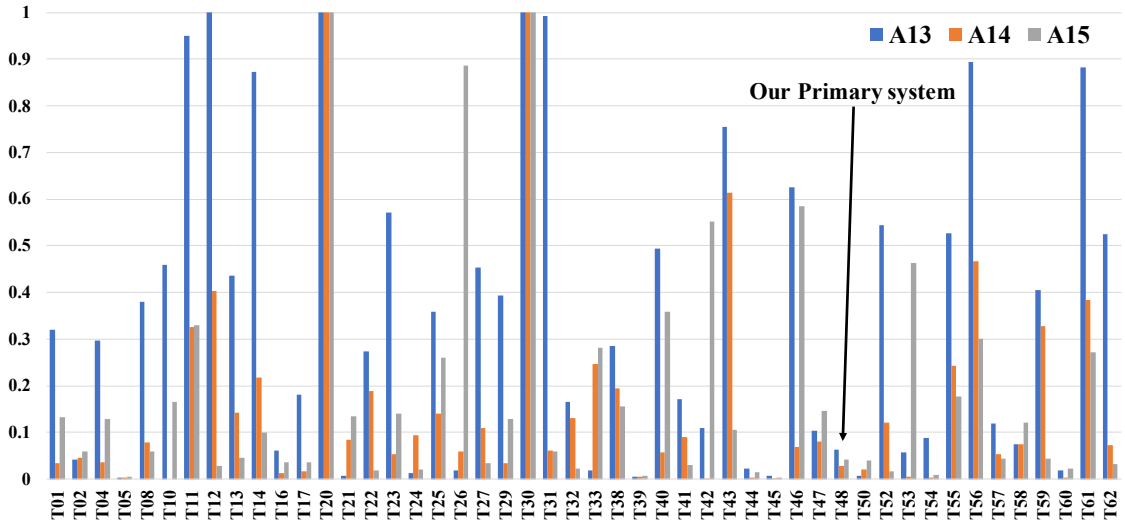


Figure 7: *Minimum t-DCF (primary evaluation metric) on the A13–A15 attack types in ASVspoof 2019 evaluation set.*

Table 1: *Minimum t-DCF and EER (secondary evaluation metric) of the ASVspoof 2019 evaluation set.*

Systems	**min. t-DCF**	**EER [%]**
Baseline-CQCC	0.2366	9.57
Baseline-LFCC	0.2116	8.09
Primary	0.1791	9.08
Single	0.2251	8.44
Contrastive	0.2143	13.87

provides some evidence that our models are not over-fitting the training data, as the generalization gap with clearly unseen data appears to be reasonable.

However, our systems have very poor results for A17 and A18 subsets, which are VC data. The performance on A17 and A18 are clearly *outliers* for the evaluation of our systems. The A17 spoofing speech is generated by waveform filtering and A18 spoofing speech is generated by vocoder. These waveform generation methods are used in the development set, so their *catastrophic performance* is somewhat difficult to explain. It almost looks like a systematic *bug*, rather than just very poor results. More inspection are needed to make more sense of these results.

5. Conclusion

In participating ASVspoof 2019, we build a spoof detection module based on learning embedding (distributed representation) functions. Initially, the embedding is learned by 7-class classification to discriminate different data types. Subsequently, it is fine-tuned by 2-class detection. We investigate TDNN-based and CNN-based embedding functions, as well as embeddings extracted from different front-end features. The best performance on the development set is achieved by combining X-vectors extracted from CQCC and LFCC features. We use average score fusion of TDNN-based X-vector and CNN-based Inception-ResNet-v1 models to detecting the spoofing speech, which eventually achieves a minimum t-DCF of 0.1791 and an EER of 9.08 in ASVspoof 2019 evaluation set.

Table 2: *Minimum t-DCF and EER for the different types of spoofing algorithms on the ASVspoof 2019 evaluation set.*

	Primary		Single		Contrastive	
	min. t-DCF	*EER*	*min. t-DCF*	*EER*	*min. t-DCF*	*EER*
Pool	0.1791	9.08	0.2251	8.44	0.2143	13.87
A07	0.0010	0.06	0.0003	0.02	0.0003	0.02
A08	0.0016	0.06	0.0018	0.06	0.0131	0.49
A09	0.0024	0.08	0.0153	0.31	0.0363	0.49
A10	0.0672	2.36	0.1503	5.18	0.0719	2.58
A11	0.0171	0.61	0.0658	2.32	0.0265	0.94
A12	0.0356	1.25	0.0843	2.93	0.0726	2.61
A13	0.0632	2.30	0.1251	4.40	0.1665	6.12
A14	0.0282	1.04	0.1118	3.99	0.0297	1.12
A15	0.0414	1.47	0.1646	5.82	0.0298	1.16
A16	0.0036	0.14	0.0003	0.02	0.0018	0.06
A17	0.9862	**25.54**	0.9484	**24.61**	0.9982	**81.77**
A18	0.9992	**29.99**	0.9984	**22.99**	1.0000	**92.09**
A19	0.0228	0.55	0.0025	0.06	0.0018	0.06

A close look at the detailed results reveals that our systems have very serious issues with A17 and A18 test subsets, and work very well with other subsets. This is certainly an unusual experience we want to figure out what has happened and we can learn from. In the future, we certainly want to understand why that occurs, and to reduce the worst-case generalization gap.

6. References

[1] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017, pp. 999–1003.

[2] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[3] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[6] Yannis Stylianou, "Voice transformation: a survey," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3585–3588.

[7] Jesús Villalba and Eduardo Lleida, "Speaker verification performance degradation against spoofing and tampering attacks," in *FALA workshop*, 2010, pp. 131–134.

[8] Nicholas WD Evans, Tomi Kinnunen, and Junichi Yamagishi, "Spoofing and countermeasures for automatic speaker verification.," in *Interspeech*, 2013, pp. 925–929.

[9] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[10] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[13] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid, "Label-embedding for image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

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

[15] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig S Greenberg, Douglas A Reynolds, Elliot Singer, Lisa P Mason, and Jaime Hernandez-Cordero, "The 2016 nist speaker recognition evaluation.," in *Interspeech*, 2017, pp. 1353–1357.

[18] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," .

[19] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin, "Audio replay attack detection with deep learning frameworks.," in *Interspeech*, 2017, pp. 82–86.

[20] Matt W Gardner and SR Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[21] Thorsten Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[22] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[23] "Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," `http://www.asvspoof.org/asvspoof2019/asvspoof2019_evaluation_plan.pdf`.