# Data Quality as Predictor of Voice Anti-Spoofing Generalization

Bhusan Chettri, Rosa González Hautamäki, Md Sahidullah, Tomi Kinnunen

# Data Quality as Predictor of Voice Anti-Spoofing Generalization

*Bhusan Chettri*[1,2], *Rosa González Hautamäki*[1,4], *Md Sahidullah*[3], *Tomi Kinnunen*[1*]

[1]School of Computing, University of Eastern Finland, Finland
[2]School of EECS, Queen Mary University of London, United Kingdom
[3]Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France
[4]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

b.chettri@qmul.ac.uk, rgonza@cs.uef.fi, md.sahidullah@inria.fr, tkinnu@cs.uef.fi

## Abstract

Voice anti-spoofing aims at classifying a given utterance either as a bonafide human sample, or a spoofing attack (e.g. synthetic or replayed sample). Many anti-spoofing methods have been proposed but most of them fail to generalize across domains (corpora) — and we do not know *why*. We outline a novel interpretative framework for gauging the impact of data quality upon anti-spoofing performance. Our within- and between-domain experiments pool data from seven public corpora and three anti-spoofing methods based on Gaussian mixture and convolutive neural network models. We assess the impacts of long-term spectral information, speaker population (through x-vector speaker embeddings), signal-to-noise ratio, and selected voice quality features.

**Index Terms**: anti-spoofing, data quality, interpretative models

## 1. Introduction

In the context of biometrics, *presentation attack detection* (PAD) or *anti-spoofing* aims at classifying a given signal either as a bonafide (human) sample or a *spoofing attack*. Replay, text-to-speech, and voice conversion attacks degrade the performance of automatic speaker verification (ASV) systems. Driven by fraud prevention in call-centers and securing our identities in other applications, a new research community working on voice anti-spoofing has emerged during the past few years. In part, research has been enabled by increased number of corpora containing both bonafide and spoofed data, such as ASVspoof [1]. There are also other public (and proprietary) data such as BTAS 2016 [2], SAS [3], ReMASC [4], and PhoneSpoof [5].

Numerous speaker-independent voice anti-spoofing methods have been proposed. Many focus on designing new acoustic features [6, 7], deep neural network (DNN) architectures [8, 9] or combining different models [10, 11] through classifier fusion. Many studies report low spoofing attack detection error rates (even 0 %) though the methods are usually tested using a single corpus only. With sufficiently many architectural modifications, control parameter optimizations and experiments it may be feasible to push error rates down on a given corpus. Performance on a single corpus, however, should not be viewed as a measure of generality or to suggest a solved task. Real-world operation demands reliable operation across *many* test conditions, most of which are never encountered during system development.

Lack of generality has been noted in (limited number of) *cross-corpus* studies [12, 13, 14, 15] where the training and test data originate from disjoint collections (often compiled by different research teams). The reported error rates, sometimes close to chance level, are disturbing as they suggest overfitting on the existing corpora. With a spoiler alert, the reader is encouraged to peek our cross-corpus results reported in Table 1.

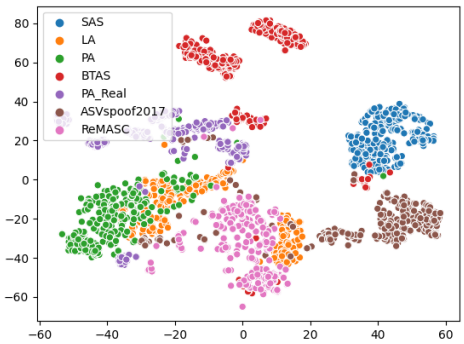


Figure 1: *Voice anti-spoofing corpora visualized with t-stochastic neighborhood embedding (t-SNE). Each point corresponds to long-term average spectrum (LTAS) of one utterance.*

But *why* voice anti-spoofing, especially across corpora, is so difficult? As an intuitive motivation, Fig. 1 visualizes spectral differences in seven different voice anti-spoofing corpora. Even if each corpus contains different spoofing attacks of varied difficulty, at the corpus level the audio files can be homogenous. This is due to shared acoustic properties that may depend on speaker population, original recording environment, choice of microphones, data processing pipelines — and perhaps even on signal scale and audio file format. Similarly, there are systematic differences across corpora due to differences in such characteristics. When a voice anti-spoofing system is trained and tested using data in a single corpus only, one conveniently sidesteps the issue of feature or representation compatibility across domains; it may not be needed as the training and test data are already homogenous in their qualities.

Our work aims at quantifying the impact of corpus-level acoustic mismatch factors upon voice anti-spoofing performance. Our work is differentiated from majority of prior work in anti-spoofing by an *explanatory* perspective. As a community, we lack *understanding* of the role of training and test data in anti-spoofing. Given the central role of data in any machine learning task (including anti-spoofing), we argue that it is useful to uncover data-related factors that contribute negatively (or positively) to performance. We approach this problem by focusing on a few carefully selected corpus level attributes, such as distribution of signal-to-noise ratio and speakers. These potentially confounding variables are then used as predictors of anti-spoofing performance in a regression analysis setting.

Our work is not the first to address the impact of factors that may influence anti-spoofing performance or bias evalua-

*All authors have equal contribution.

tion results. Prior work has addressed, for instance, the impact of waveform sample distributions [16] and biases due to presence of silence regions [10, 17]. Other work, such as [18], have provided interpretations beyond error rates for specific anti-spoofing methods. Our work is differentiated from these studies in that we propose a *unified* framework for assessing data-related quality factors, treated as predictors in a regression model setting. What follows is description of our framework and preliminary experiments that pool data from seven different anti-spoofing corpora.

# 2. Methodology

## 2.1. Re-thinking training and test sets as random data

Assume that we have a total of $M$ distinct, labeled anti-spoofing collections $\{\mathcal{D}_i\}_{i=1}^{M}$ available (here, $M = 7$). The $i^{\text{th}}$ collection contains, respectively, $N_{\text{bona}}^{(i)}$ and $N_{\text{spoof}}^{(i)}$ bonafide (human) and spoof audio files. Each file is labeled as either one of these two classes. Each collection (e.g. particular ASVspoof edition) is assumed to consist of somewhat homogenous audio material, while different collections — possibly compiled by different researchers — are assumed to be more heterogenous. Each collection can be understood as a cluster or group of audio files that share some commonalities. The reported performance gap of within-corpus vs. cross-corpus results [12], along with Table 1 and the visualization in Fig. 1 on long-term spectral characteristics provide support for these assumptions.

Typically, a voice anti-spoofing corpus contains an *evaluation protocol* that defines partitioning of the speech files into training and test portions[1]. Even if standard evaluation protocols are necessary for commensurable performance comparisons, a protocol defines only one possible data partitioning of all the available data. As a result, reported anti-spoofing results on a given corpus may be specific to that random partitioning. In stark contrast to fixed train-test protocol division, *we consider the training/test corpora as random observations*. Whenever the anti-spoofing system (and its parameters) are frozen, one obtains *one* performance number (such as equal error rate, EER) for a fixed evaluation protocol. We, instead, gather *several* repeated measurements of the selected performance measure (here, the EER) within and across data collections.

In practice, for each of the $M$ collections we designate a *single* training set $\mathcal{D}_{\text{train}}^{(i)}$ and *multiple* test sets, $\mathcal{D}_{\text{test}}^{(i,j)}, j = 1, \ldots, N_{\text{test}}^{(i)}$. In principle, this choice is arbitrary and we could have also fixed the test sets and sample random training sets instead. The choice is primarily dictated by computational reasons elaborated shortly. We sample equal number of test portions within each collection: $N_{\text{test}}^{(1)} = \cdots = N_{\text{test}}^{(M)} \equiv N_{\text{test}}$. Note that the special case $N_{\text{test}} = 1$ corresponds to conventional approach where a given corpus is equipped with a predefined evaluation protocol. In our revised set-up we train and test anti-spoofing systems across all the collections. This yields $N_{\text{test}}$ within-corpus and $(M-1) \times N_{\text{test}}$ cross-corpus experiments, *per training set*. As we have $M$ training sets (one per corpus), we have a total of $M \times N_{\text{test}}$ within-corpus results and $M \times (M-1) \times N_{\text{test}}$ cross-corpus results. In our experiments, $N_{\text{test}} = 20$ which implies 140 within- and 840 cross-corpus experiments. This is why we fix the training partition and treat only the test portions as random: despite the large number of EERs produced, we need to train only $M = 7$ anti-spoofing models (one per collection).

[1] ASVspoof challenges contain *train*, *development* and *evaluation* sets; we do not differentiate between the latter two which, really, are two different test sets. During a challenge, the labels of development set are available for detector optimization while test data that lacks labels.

Table 1: *Cross-corpus performance (EER%) of spoofing countermeasures. 2017: ASVspoof 2017 v2.0, PA: ASVspoof 2019 PA, RPA: ASVspoof 2019 Real PA, LA: ASVspoof 2019 LA. ReM: ReMASC, BT: BTAS. An EER of greater than* 50% *indicates chance level in a 2-class task.*

| | Tested on | | | | | | |
|---|---|---|---|---|---|---|---|
| | SAS | LA | 2017 | RPA | ReM | PA | BT |
| SAS | **0.99** | 62.08 | 53.76 | 70.18 | 46.85 | 52.29 | 73.51 |
| LA | 45.07 | **11.17** | 41.06 | 34.0 | 49.37 | 36.16 | 81.75 |
| 2017 | 52.97 | 39.07 | **13.02** | 41.83 | 43.52 | 47.92 | 70.6 |
| RPA | 52.01 | 53.75 | 46.16 | **39.35** | 45.46 | 46.52 | 54.77 |
| ReM | 42.27 | 24.2 | 54.08 | 58.58 | **50.3** | 48.88 | 66.02 |
| PA | 65.56 | 24.94 | 52.21 | 29.88 | 47.39 | **7.0** | 13.87 |
| BT | 61.13 | 68.59 | 17.03 | 33.49 | 43.78 | 46.44 | **0.18** |

## 2.2. Overview of multiple linear regression setting

We model the dependency of anti-spoofing performance upon data-related mismatch factors. For instance, if the training and test data consist of homogenous speakers (e.g. all have the same gender or native language) one might expect better performance compared to a situation with disjoint speaker qualities. We consider paired observations $\{(\boldsymbol{d}_t, E_t) : t = 1, \ldots, T\}$ where $E_t$ is performance metric (here, bonafide-vs-spoof EER) for training-test pair indexed by $t$ and $\boldsymbol{d}_t = (d_t^{(1)}, \ldots, d_t^{(R)})^{\mathsf{T}} \in \mathbb{R}^R$ is a set of predictors suspected to influence $E_t$. We model the assumed statistical dependency using *multiple linear regression*. Our prime interest is in the relative contribution of the individual predictors $d_t^{(1)}, \ldots, d_t^{(R)}$, each of which is a distance between the training and test sets, formalized next.

## 2.3. Defining the predictors (corpus distances)

Let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ denote training and test sets that are used, respectively, to train and score any anti-spoofing system. They could be sets within the same collection or sets taken from different collections; this distinction is not important as the procedure of distance computation is the same. Let $\mathcal{D}_{\text{train}} = \{(\mathcal{X}_j, y_j)\}_{j=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{(\mathcal{X}_m, y_m)\}_{m=1}^{N_{\text{test}}}$ denote training and test waveforms $\mathcal{X}$ paired up with their ground-truth labels, $y \in \{0 \equiv \text{spoof}, 1 \equiv \text{bonafide}\}$. The $j$th waveform, $\mathcal{X}_j$, is represented by a set of quality features, $\boldsymbol{\phi}_j^{(1)}, \ldots, \boldsymbol{\phi}_j^{(Q)}$. They may have different dimensionalities and numerical ranges. For instance, $\boldsymbol{\phi}_j^{(1)}$ might be scalar-valued signal-to-noise ratio (SNR) and $\boldsymbol{\phi}_j^{(2)}$ a 512-dimensional deep speaker embedding. Each feature set corresponds to attributes suspected to influence anti-spoofing performance but which (ideally) should be uninformative about the class label $y$. For instance, one is not supposed to detect a spoofing attack based on knowledge of the speaker (at least in speaker-independent anti-spoofing setting). At the level of the corpus, however, it is useful to gauge the potential impact of speaker population upon anti-spoofing performance.

In practice, we treat each of the $Q$ features independent of each other. We drop the feature superscript momentarily and use $\boldsymbol{\phi}_j$ to denote any of the $Q$ measurements of file $j$. The observed quality data are then $\mathcal{D}_{\text{train}} = \{(\boldsymbol{\phi}_j, y_j)\}_{j=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{(\boldsymbol{\phi}_m, y_m)\}_{m=1}^{N_{\text{test}}}$, viewed as i.i.d. samples from some underlying true distribution $p(\boldsymbol{\phi}, y)$. By conditioning the data distribution both by the class label (bonafide/spoof) and the data portion (train/test) we have four conditional data distributions in total, as illustrated in Fig. 2. For regression modeling, we

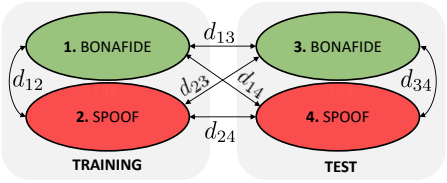


Figure 2: *For each of $Q$ quality measures, six distances are computed: within- and between-class distances of bonafide and spoof (both within and across training and test data).*

consider all the indicated six distances, for a given set of quality measurements (thus, the maximum number of predictors is $R = 6Q$, obtained by cross-combining all $Q$ quality measurements with the six different distances). The distance that we use is *Chamfer distance* (or *modified Hausdorff distance*) based on averaged Euclidean squared distances with nearest-neighbor rule. It can be computed without numerical issues for features of any dimensionality. It gives non-negative distance of two point clouds, each of which corresponds to one the four portions shown in Fig. 2. Chamfer distance is not symmetric but we compute distance to both directions and average the two values. We also normalize the distance by the dimensionality of the respective quality measurement.

The within-class distances across training and test are perhaps most easily intuitively understood. For instance, $d_{13}$ measures how much bonafide data qualities between training and test data differ (likewise for spoof, $d_{24}$). The remaining four *cross-class* distances may appear strange at first but we have a reason to include them in our models. If *both* bonafide and spoof are corrupted by similar nuisance variations (e.g. both are either clean or noisy) one may expect lower anti-spoofing EER as the classifier does not have to address the issue of noise. Similarly, if the training distributions of bonafide and spoof distributions are very different, the anti-spoofing system may learn to *cheat* (take a shortcut) by extracting information unrelated to bonafide-spoof discriminating cues — hence, potentially exhibit low generalization performance.

# 3. Experimental Setup

## 3.1. Spoofing corpora

We use seven publicly available corpora: SAS [3], ASVspoof 2017 v2.0 [19], ASVspoof 2019 (LA, PA and PA Real) [20], BTAS 2016 [2], and ReMASC [4]. The SAS corpus was created for anti-spoofing research with seven voice conversion (VC) and three speech synthesis (SS) methods. The subsequent ASVspoof 2015 corpus includes the same attacks. While the ASVspoof 2017 corpus contains real replay attack recordings, the ASVspoof 2019 PA corpus consists of simulated replay attacks. PA real is a small test set that contains real replayed audio files. ReMASC [4] is a another publicly available corpus for replay spoofing attack research in voice controlled applications. We also include thecorpus used in BTAS 2016 anti-spoofing competition. It consists of different types of replay attacks [2].

## 3.2. Random training-test protocol design

We have created multiple train-test conditions with smaller subsets. We sampled the training data to create a smaller training subset balanced according to the number of utterances and speakers. We include five speakers from each corpus, each with 10 bonafide and 50 spoofed utterances. This results to 300 training utterances per corpus. Similarly, we created 20 test sets for each of the seven corpora, each consisting of 50 bonafide and 250 spoofed utterances. The bonafide-to-spoof utterance ratio approximately corresponds to the ratio in standard evaluation protocols — there are typically far more spoofed than bonafide utterances available. We selected the speakers and the utterances from the respective pre-defined 'train' and 'evaluation' partition randomly. Due to unavailability of the speaker partitioning of training and evaluation in ReMASC and ASVspoof 2019 Real PA, we select the speakers of train and test in a disjoint manner.

## 3.3. Classifiers and performance measures

We use Gaussian mixture model (GMM) and convolutional neural network (CNN) as classifiers, due to their extensive use in anti-spoofing research [8, 7, 1, 21]. The GMM-based systems are the same as the two baseline systems used in the ASVspoof 2019 challenge. They operate on 60-dimensional linear frequency cepstral coefficients (LFCCs) and 90-dimensional constant-Q cepstral coefficients (CQCCs), respectively. Two GMMs are trained to model the distribution of bonafide and spoof data using 512 mixture components. The CNN system, in turn, uses power spectrogram inputs. It is trained discriminatively to optimise cross-entropy between bonafide and spoofed class using Adam optimiser. We use the CNN architecture, training and testing approach from [22].

We evaluate classifier performance using equal error rate (EER) as a measure of bonafide-spoof discrimination. We compute EER using the public scoring toolkit used in the ASVspoof 2019 challenge. Table 1 summarises the cross-corpus performance evaluation of CNN countermeasure. As can be seen, performance is reasonable for (some) within-corpus tasks but consistently low in cross-corpus scenarios, as expected [12].

## 3.4. Quality features

We include five types of quality features. Four of them are computed with rule-based methods available in common toolkits, while one (x-vector) uses a data-driven approach, which makes feature values dependent on the training data of the extractor.

**LTAS** represents spectral information averaged over time. We compute 257-dimensional LTAS per utterance using 512-point FFT from 32 ms Hanning-windowed frames shifted by 10 ms.

**SNR** is computed using *waveform amplitude distribution analysis* (WADA) method [23], which assumes that the amplitude of the speech can be approximated with Gamma distribution with shape parameter 0.4 and the noise by Gaussian distribution. WADA shows competitive performance compared to the DNN-based data-driven methods specially in higher SNR conditions [24], case relevant for our data.

**Noise spectrum** Besides scalar-valued SNR, we also estimate noise spectral density using optimal smoothing and minimum statistics method [25]. The method estimates noise for all frequency bins in every speech frame. We average these noise spectral densities to obtain 257 coefficients per utterances.

**X-vector** represents 512-dimensional deep speaker embedding extracted with pre-trained models trained on VoxCeleb corpus [26], processed further with length normalization. Though x-vectors depend on training data and may contain nuisance variations [27], the pre-trained model shows reasonable speaker verification EER of 3.13% on VoxCeleb1 test set. This indicates high specificity to speaker-related information.

**Acoustic descriptors** are extracted using openSMILE toolkit 2.3 [28]: *fundamental frequency* (F0), *formant frequencies* (F1

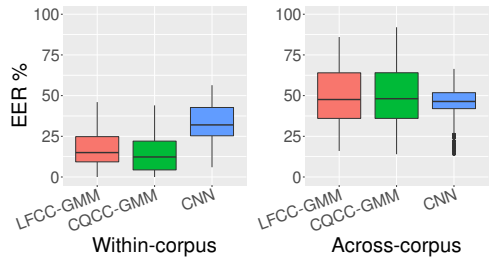


Figure 3: *EER distribution on 20 randomly created trial lists.*

to F4), and *loudness*. The feature extraction configuration corresponds to the *extended Geneva Minimalistic Standard Parameter Set* [29] summarized with the **mean** of the descriptor at the utterance level. F0 is presented in a semitone scale. Loudness is an estimate of the perceived signal energy from an auditory spectrum from perceptual linear prediction (PLP) analysis [30].

## 4. Results

We explore the collinearities between the predictive features and the performance of classifiers using the Pearson correlation. The relation was analyzed considering the data grouped in within- and across-corpus that include 140 and 840 data points respectively, each with the six distances from the predictive features (as illustrated in Fig. 2). Figure 3 shows the EER distribution for the three classifiers and describes the dependent variable variations to be explored by the regression models.

Table 2 shows, as an example, the correlation of LTAS feature distances with the EER of the CNN classifier (similar trends were observed for the other predictive features). Within-corpus correlations are stronger than across-corpus correlations. This indicates collinearity of the within-corpus distances and the performance of the classifiers. As for the distances, the four cross-class distances ($d_{12}, d_{14}, d_{23}, d_{34}$) have stronger correlations with EER (whether positive or negative) than the within-class distances ($d_{13}, d_{24}$). Note also that, apart from $d_{13}$ on across-corpus case, the within-class correlations are positive. This is as expected: the larger the domain mismatch in either bonafide or spoof class, the higher the EER.

Table 2: *Pearson correlation between LTAS distances and the equal error rate for the CNN classifier.*

| | $d_{12}$ | $d_{13}$ | $d_{23}$ | $d_{14}$ | $d_{24}$ | $d_{34}$ |
|---|---|---|---|---|---|---|
| Within-corpus | −0.605 | 0.162 | −0.584 | −0.651 | 0.367 | −0.737 |
| Across-corpus | 0.107 | −0.115 | 0.044 | −0.107 | 0.029 | 0.099 |

We now turn our focus on the predictive features. To this end, we created *multiple linear regression model* for each feature to measure how well the six distances predict the corresponding EER. The *coefficient of determination*, or $R^2$ [31], measures the proportion of the total variation of the dependent variable (EER) that is explained by the fitted model. The higher the number, the better the model fits the data. *Adjusted-*$R^2$ takes into account the number of predictors included in the model and how they contribute information. If the predictor is not significant, the adjusted-$R^2$ will compensate it by penalizing the model fit.

Table 3 presents the adjusted-$R^2$ for the feature models for each classifier separately for within- and across-corpus data. The values can be compared across the rows for each classifier to identify the data quality feature that better explain the EER variations. For instance, in the within-corpus data, for LFCC-GMM classifier all the feature distance models are good at explaining the performance, particularly LTAS distance predictors explain 68% of the EER's variation. Similar strong dependencies are noted with SNR for CQCC-GMM and with x-vector for CNN. Though the adjusted-$R^2$ are lower for across-corpus data, the same features explained the classifiers' EERs with high levels of significance. It is worth noting that our aim is to identify features that best explain the classifiers' performance, rather than searching for the best combination of different predictors. All our features explain well the variation in EER, especially for within-corpus data.

Table 3: *Adjusted-*$R^2$ *for grouped feature distances models of within- and across-corpus data of the three classifiers. Dimensionality of each feature set is indicated in parenthesis.*

| | **Within-corpus data** | | | **Across-corpus data** | | |
|---|---|---|---|---|---|---|
| | **LFCC GMM** | **CQCC GMM** | **CNN** | **LFCC GMM** | **CQCC GMM** | **CNN** |
| LTAS (257) | 0.679 | 0.543 | 0.670 | 0.289 | 0.166 | 0.038 |
| F1..F4 (4) | 0.641 | 0.497 | 0.470 | 0.131 | 0.075 | 0.142 |
| F0 (1) | 0.513 | 0.328 | 0.073 | 0.058 | 0.082 | 0.096 |
| x-vec. (512) | 0.558 | 0.642 | 0.817 | 0.067 | 0.149 | 0.202 |
| SNR (1) | 0.593 | 0.715 | 0.187 | 0.160 | 0.227 | 0.075 |
| Noise s. (257) | 0.649 | 0.439 | 0.565 | 0.141 | 0.207 | 0.010 |
| Loudness (1) | 0.455 | 0.304 | 0.448 | 0.021 | 0.050 | 0.060 |

So, what does Table 3 suggest? Due to space reasons, we arbitrarily pick the strongest and weakest individual predictors per classifier:

1. LFCC-GMM is most strongly impacted by LTAS, least by loudness;
2. CQCC-GMM is most strongly impacted by SNR, least by loudness;
3. CNN is most strongly impacted by x-vector, least by F0 (within-corpus) or noise spectrum (across-corpus).

So one may conjecture, for instance, that the CQCC-GMM system is potentially sensitive to noise (suggested earlier through simulated additive noise experiments [32]) and the CNN system potentially more strongly impacted by the choice of speaker population. The authors emphasize *potentially*: it is acknowledged that, despite speaker-discriminative training objective, x-vectors are not 'pure' speaker representations [27]. Their quality depends on several factors (including the choice of training data).

## 5. Conclusions

We addressed the role of data quality in voice anti-spoofing generalization. The framework can be used to address statistical dependency between selected quality features and anti-spoofing performance. Pinpointing the potential issues can be used to design better anti-spoofing systems where the unwanted variations are suppressed in explicit ways. Our future plans include addressing further quality features and distance measures, *mixed effects* regression modeling, adding more powerful classifiers, and using the acquired knowledge to improve selected classifiers. In many classification tasks, performance can be improved by using additional training or adaptation data from the target domain. Our assumption, however, is that only a single training domain is available. The intention was to address 'the truly unknown' in terms of domain variation. Our findings indicate that substantial further research remains in this area.

## 6. Acknowledgements

This work was supported in part by the Academy of Finland (Proj. No. 309629) and Nokia Foundation.

# 7. References


[1] X. W. et al., "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech and Language*, vol. 64, p. 101114, 2020. [Online]. Available: https://doi.org/10.1016/j.csl.2020.101114

[2] P. K. et al., "Overview of BTAS 2016 speaker anti-spoofing competition," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1–6.

[3] Z. W. et al., "SAS: A speaker verification spoofing database containing diverse attacks," in *Proc. IEEE ICASSP*, 2015, pp. 4440–4444.

[4] Y. Gong, J. Yang, J. Huber, M. MacKnight, and C. Poellabauer, "ReMASC: Realistic replay attack corpus for voice controlled systems," in *Proc. Interspeech*, 2019, pp. 2355–2359. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1541

[5] G. L. et al., "PHONESPOOF: A new dataset for spoofing attack detection in telephone channel," in *Proc. IEEE ICASSP*, 2019, pp. 2572–2576.

[6] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech and Language*, vol. 45, pp. 516 – 535, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230816303114

[7] G. Suthokumar, V. Sethu, C. Wijenayake, and E. Ambikairajah, "Modulation dynamic features for the detection of replay attacks," in *Proc. Interspeech*, 2018, pp. 691–695.

[8] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," in *Proc. Interspeech*, 2017, pp. 82–86. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0360.html

[9] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A gated recurrent convolutional neural network for robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1985–1999, 2019.

[10] B. C. et al., "Ensemble models for spoofing detection in automatic speaker verification," in *Proc. Interspeech*, 2019, pp. 1018–1022. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2505

[11] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "ResNet and model fusion for automatic spoofing detection," in *Proc. Interspeech*, 2017, pp. 102–106. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1085

[12] P. Korshunov and S. Marcel, "A Cross-Database Study of Voice Presentation Attack Detection," in *Handbook of Biometric Anti-Spoofing - Presentation Attack Detection, 2nd Ed.*, ser. Advances in Computer Vision and Pattern Recognition, S. M. et al, Ed. Springer, 2019, pp. 363–389. [Online]. Available: https://doi.org/10.1007/978-3-319-92627-8_16

[13] D. Paul, M. Sahidullah, and G. Saha, "Generalization of spoofing countermeasures: A case study with asvspoof 2015 and btas 2016 corpora," in *Proc. IEEE ICASSP*, 2017, pp. 2047–2051.

[14] R. K. Das, J. Yang, and H. Li, "Assessing the scope of generalized countermeasures for anti-spoofing," in *Proc. IEEE ICASSP*, 2020, pp. 6589–6593.

[15] P. Parasu, J. Epps, K. Sriskandaraja, and G. Suthokumar, "Investigating light-ResNet architecture for spoofing detection under mismatched conditions," in *Proc. Interspeech*, 2020.

[16] I. Lapidot and J.-F. Bonastre, "Effects of waveform PMF on anti-spoofing detection for replay data - ASVspoof 2019," in *Proc. Speaker Odyssey*, 2020, pp. 312–318. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-44

[17] B. Chettri, E. Benetos, and B. L. Sturm, "Dataset artefacts in anti-spoofing systems: a case study on the ASVspoof 2017 benchmark," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[18] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification," in *Proc. Speaker Odyssey*, 2020, pp. 333–340. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-47

[19] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi, "Asvspoof 2017 version 2.0: meta-data analysis and baseline enhancements," in *Proc. Speaker Odyssey*, 2018, pp. 296–303. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2018-42

[20] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. Interspeech*, 2019, pp. 1008–1012. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2249

[21] M. Sahidullah, H. Delgado, M. Todisco, T. Kinnunen, N. Evans, J. Yamagishi, and K.-A. Lee, "Introduction to voice presentation attack detection and recent advances," in *Handbook of Biometric Anti-Spoofing*. Springer, 2019, pp. 321–361.

[22] B. Chettri, T. Kinnunen, and E. Benetos, "Subband modeling for spoofing detection in automatic speaker verification," in *Proc. Speaker Odyssey*, 2020, pp. 341–348. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2020-48

[23] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Proc. Interspeech*, 2008, pp. 2598–2601.

[24] H. Li, D. Wang, X. Zhang, and G. Gao, "Frame-level signal-to-noise ratio estimation using deep learning," *Proc. INTERSPEECH*, pp. 4626–4630, 2020.

[25] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[26] "VoxCeleb Xvector models system 1a," https://kaldi-asr.org/models/m7, accessed: 2021-03-20.

[27] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," in *Proc. IEEE ASRU*. IEEE, 2019, pp. 726–733.

[28] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the munich open-source multimedia feature extractor," in *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference*, 10 2013, pp. 835–838.

[29] F. E. et al., "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, 2016.

[30] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[31] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2.

[32] C. Hanilçi, T. Kinnunen, M. Sahidullah, and A. Sizov, "Spoofing detection goes noisy: An analysis of synthetic speech detection in the presence of additive noise," *Speech Communication*, vol. 85, pp. 83–97, 2016.