# Visualizing Classifier Adjacency Relations: A Case Study in Speaker Verification and Voice Anti-Spoofing

Tomi Kinnunen, Andreas Nautsch, Md Sahidullah, Nicholas Evans, Xin Wang, Massimiliano Todisco, Héctor Delgado, Junichi Yamagishi, Lee Kong Aik

# Visualizing Classifier Adjacency Relations: A Case Study in Speaker Verification and Voice Anti-Spoofing

*Tomi Kinnunen*[1], *Andreas Nautsch*[2], *Md Sahidullah*[3], *Nicholas Evans*[2], *Xin Wang*[4], *Massimiliano Todisco*[2], *Héctor Delgado*[5], *Junichi Yamagishi*[4], *Kong Aik Lee*[6]

[1]University of Eastern Finland, Finland
[2]Eurecom, France
[3]Inria, France
[4]National Institute of Informatics, Japan
[5]Nuance Communications, Spain
[6]Institute for Infocomm Research, Singapore

tomi.kinnunen@uef.fi,andreas.nautsch@eurecom.fr,md.sahidullah@inria.fr

## Abstract

Whether it be for results summarization, or the analysis of classifier fusion, some means to compare different classifiers can often provide illuminating insight into their behaviour, (dis)similarity or complementarity. We propose a simple method to derive 2D representation from detection scores produced by an arbitrary set of binary classifiers in response to a common dataset. Based upon rank correlations, our method facilitates a visual comparison of classifiers with arbitrary scores and with close relation to receiver operating characteristic (ROC) and detection error trade-off (DET) analyses. While the approach is fully versatile and can be applied to any detection task, we demonstrate the method using scores produced by automatic speaker verification and voice anti-spoofing systems. The former are produced by a Gaussian mixture model system trained with VoxCeleb data whereas the latter stem from submissions to the ASVspoof 2019 challenge.
**Index Terms**: classifier, multi-dimensional scaling

## 1. Introduction

Whether it be for challenge results summarization [1, 2, 3, 4] or the analysis of classifier fusion, some means to compare alternative classifiers can often provide illuminating insight into robustness, coherence, and generalisation [5]; (dis)similarity, and complementarity. Depending on prior knowledge (and data) available, one could compare

- the underlying working principles, features, architectures, training data properties — **classifier metadata**,
- **empirical performance** (e.g. accuracy) on a common set of evaluation data,

to name two possibilities. Often we want to understand how the two are related — what is the impact of specific modeling choices upon performance. A common approach is to fix some parameters while varying others to find out their impact on performance. Whenever one implements the classifier and has full control over the experiment, this *white-box* approach is relatively straightforward. There are, however, situations where one may not have access to exhaustive classifier details and configuration settings but would still like to learn about similarities between alternative solutions. One such *black-box* scenario are public machine learning challenges and technology benchmarks. Whether it be NIST speaker recognition evaluations [2, 3], VoxCeleb [4], ASVspoof [1], or indeed any other competitive campaigns, participants typically run their in-house systems on a common evaluation set and submit predictions (e.g. scores) for unlabeled data along with a system description. Since source codes or models are often not required, there can be uncertainty as to why specific challenge entries outperform others. The authors' personal motivation for the presented work stems partially from difficulties encountered in our efforts to link classifier properties to their performance in a recent challenge [1]. We wanted to find out what can be learnt about classifier differences based on detection scores.

To this end, we propose a method for computing the distance between binary classifiers based on the scores produced for common evaluation data (Fig. 1). We impose a stronger notion of classifier (dis)similarity than the established methods of *detection error trade-off* (DET) [6] and *receiver operating characteristics* (ROC) analyses: to be considered identical, a pair of classifiers must agree not only in their DET profiles but relative ordering of individual trials. Similar principles are the basis for a number of statistical significance tests but our perspective is in visualizing classifiers beyond DET/ROC plots.

As Fig. 1 suggests, we conjecture that distances derived from scores may provide information on (possibly unknown) classifier metadata differences. After describing the background and methodology, we report an example application of our methodology to automatic speaker verification (ASV) and voice anti-spoofing. The former includes classifiers constructed by the authors (known classifier metadata) while the latter represents the less controlled case of submissions to the ASVspoof 2019 challenge. Note, however, that the proposed methods are not in any way specific or limited to the speech domain. To this end, we provide an open-source reference implementation[1].

## 2. Trade-offs in Characterizing Classifiers

There are various ways to describe a classifier. The coarsest description might include the general model class — such as 'a linear classifier' or 'a deep neural network.' A more refined description might detail the features and their dimensionality, the type and number of layers, the software toolkit, loss function,

[1] https://github.com/asvspoof-challenge/classifier-adjacency (referenced June 10, 2021).

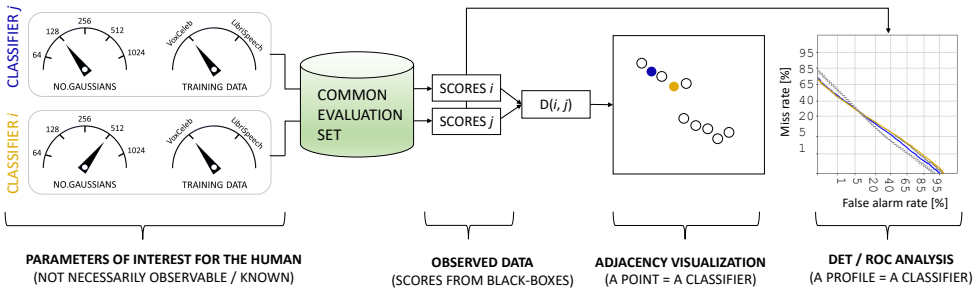



Figure 1: *The behavior of classifiers can be controlled using a limited number of parameters meaningful to the experimenter. Not all the details, however, may be available to an evaluator. We propose to visualize classifier adjacencies based on their scores to complement other traditional analyses. The adjacencies can be informative of the adjacency of the underlying control parameters.*

or training corpus to name a few possibilities.

Depending on the target group (e.g. general public, scientists, students, software engineers) one may prefer different levels of detail. What is in common is that the descriptions are designed for *communicating* key principles rather than enabling perfect reproducibility. The description is a human-readable tuple, $\mathbf{h}$. For instance, our automatic speaker verification (ASV) systems described below are characterized by triplets $\mathbf{h} = (\texttt{TrData}, F, G)$ where $\texttt{TrData} \in \{\texttt{VoxCeleb}, \texttt{LibriSpeech}\}$ indicates the training corpus while $F$ and $G$ are integers that indicate the number of mel filters and Gaussians, respectively. Each choice (attribute value in $\mathbf{h}$) impacts results but there is no expectation for such triplets to be a complete description; it is a description intended for a particular audience (here, Interspeech attendee).

At the other extreme, the most precise description is a packaged/trained model — a list of all parameters, $\boldsymbol{\theta} \in \mathbb{R}^D$. For instance, $\boldsymbol{\theta}$ could represent all the weights of a deep neural network (DNN), $D$ being potentially in the order of millions. If scoring is deterministic (as is usually the case), $\boldsymbol{\theta}$ and the functional form $g(X;\boldsymbol{\theta})$ completely specify how an arbitrary input $X$ maps to an output (predicted class label or score), $X \mapsto \hat{Y} = g(X;\boldsymbol{\theta})$. This leads to perfect reproducibility — one obtains the same scores, and hence also results, every time. Unlike $\mathbf{h}$, however, $\boldsymbol{\theta}$ is not intended to be comprehensible to the human: eyeballing the numerical values of DNN weights (an exercise which could take a while) is not particularly helpful in understanding *how* the system works or *why* it may work/fail in specific cases. Another problem is incomparability of the parameter vectors of different models. They could have different dimensionalities and there may be no meaningful correspondence between dimensions.

One more level of description are the scores that can be used to derive detection error trade-off (DET) profiles [6] and metrics such as the equal error rate (EER). Unfortunately, being based on score *distributions*, DETs or EERs are not informative of classifier responses to individual trials. A common compromise is to define a limited number of *evaluation conditions* — subsets of trials with characteristics useful for diagnostics (e.g. 'telephony', 'short-duration', 'attack S10', 'females'). Performance breakdown by condition is useful in analysing trends.

In summary, there are different abstraction levels to characterize classifiers. We aim at learning about their differences in terms of human-targeted descriptions *without the precise knowledge of the attribute values themselves*. We conjecture that classifiers that are similar in their $\mathbf{h}$ descriptions may produce similar outputs when executed on common data. In this case, one can learn about classifier design differences from their scores. To this end, we propose a method to compute pairwise classifier distances for common evaluation data. The distances can then be used for different purposes, such as agglomerative clustering or visualisation, to help in demonstrating trends or diagnosing classifier complementarities.

## 3. Proposed Classifier Adjacency Visualizer

### 3.1. Assumed input data

Let $X$ and $Y$ denote some input space and (binary) labels, respectively. We use $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N} \sim_{\text{i.i.d.}} p(X, Y)$ to denote a labeled (supervised) evaluation set of $N$ *trials*. In an ASV set-up, each $X_i$ corresponds to an enrollment-test pair and $Y_i \in \{0, 1\}$ indicates whether the speaker identities in the two utterances are the same (target trial) or different (nontarget trial). In voice anti-spoofing, in turn, each $X_i$ corresponds to a test file and $Y_i$ indicates whether $X_i$ represents bonafide audio or a spoofing attack (such as computer-generated speech).

We assume a set $\mathcal{G} = \{g_1, \ldots, g_M\}$ of $M$ *binary classifiers* (detectors), $g_j : X \to \mathbb{R}$, each of which assigns a numeric membership value (*detection score*) to each trial, with the convention that high scores indicate support towards the positive class (e.g. target speaker). Unless we construct the classifiers ourselves, we do not have access to the scoring functions $\{g_j\}$ but only to their *outputs* for the shared evaluation data $\mathcal{D}$ (in turn, in a challenge setting, only the organizer knows the ground truth values $\{Y_i\}$ while the participants produce predictions on the test trials $\{X_i\}$ blindly). We use $s_{i,j} = g_j(X_i)$ to denote the response of the $j$th classifier to the $i$th trial. The data used for our modeling consists of the responses of all classifiers to all the test trials — an $N \times M$ matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_M]$ where $\mathbf{s}_j = [s_{1,j}, \ldots, s_{N,j}]^\mathsf{T}$ is a column vector of size $N \times 1$.

Optionally, one may have additional trial metadata. In this case, we assume that each trial can be uniquely assigned to one member of a mutually exclusive set of *groups* (conditions), $c \in \{1, 2, \ldots, C\}$. In our experiments, for instance, groups in the ASV application correspond to the specific pair of speaker identities in the trial. We denote the number of trials in group $c$ by $N_c$. As the groups are mutually exclusive, $N = \sum_{c=1}^{C} N_c$. We use either the original $N \times M$ score matrix or the smaller $C \times M$ matrix $\mathbf{S}' = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M]$ where each group is represented using its average score. Here, $\boldsymbol{\mu}_j = [\mu_{1,j}, \ldots, \mu_{C,j}]^\mathsf{T}$ is a column vector of size $C \times 1$ for the $j$th classifier.

### 3.2. Classifier Adjacency

To identify similar (and dissimilar) systems, we define a distance function between classifiers, indicated by $D(i, j)$, where $i, j \in \{1, \ldots, M\}$. It is computed either from $\mathbf{S}$ or $\mathbf{S}'$. We

first note that, since the numerical range of detection scores can be arbitrary, a direct comparison of scores (e.g. using Euclidean distance) is usually not meaningful. In the ASV field, 'correction' for the range of scores is typically addressed through *calibration* (e.g. [7]) that converts arbitrary scores into calibrated log-likelihood ratios (LLRs). Here, however, since score calibration is not our main concern, we opt for more straightforward scale- and translation-invariant distance computation. To this end, we define classifier distance through *order statistics* as opposed to raw scores. This choice stems from the knowledge that any two classifiers which yield the same order of trial scores on common data will share the same detection error trade-off (DET) curve and equal error rate (EER) [8, p.81][2].

We use *Kendall's* $\tau$ [9, 10, 11] as the rank correlation measure between classifiers $i$ and $j$:

$$\tau(i,j) = \frac{N_{\text{con}} - N_{\text{dis}}}{\sqrt{(N_{\text{con}} + N_{\text{dis}} + T_i) \times (N_{\text{con}} + N_{\text{dis}} + T_j)}}, \quad (2)$$

where $N_{\text{con}}$ and $N_{\text{dis}}$ are, respectively, the number of concordant and discordant pairs between $i$ and $j$. Further, $T_i$ is the number of ties only in $i$, and $T_j$ is the number of ties only in $j$. *Concordance* of a pair of trials means that the sort order agrees across the classifiers. For instance, if the $a^{\text{th}}$ and $b^{\text{th}}$ trial scores are ordered $s_{a,i} < s_{b,i}$ in classifier $i$, the trial pair $(a, b)$ is cordordant with classifier $j$, if similarly $s_{a,j} < s_{b,j}$. Likewise, if $(s_{a,i} > s_{b,i})$ AND $(s_{a,j} > s_{b,j})$, pair $(a, b)$ is again corcondant. If concordance is not satisfied, the pair $(a, b)$ is *discordant*. A *tie* within either classifier means that there identical scores produced for different trials. With real-valued scores stored in float or double precision this is generally a rare case.

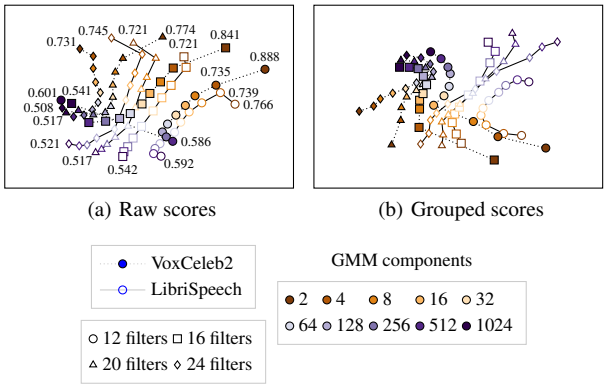
An intuitive feeling of (2) can be developed by first assuming there are no ties ($T_i = T_j = 0$). Now, if the two classifiers place all trials into the same sort order, $N_{\text{dis}} = 0$ and $\tau(i,j)$ reaches the maximum value of 1. The other extreme is obtained when the sort order of trials of one classifier is the reverse of the other. This corresponds to $N_{\text{con}} = 0$ and yields the minimum value of $-1$. Finally, $\tau = 0$ indicates lack of statistical association between the two. To sum up, $\tau(i,j)$ takes values in $[-1, 1]$ (similar to Pearson correlation) and can be considered as a degree of agreement in the sort order of trials. Whenever $\tau(i,j) = 1$, the classifiers have identical DET or ROC profiles.

For the purposes of visualization, we map $\tau$ into distances as $D(i,j) = \frac{1}{2}(1 - \tau(i,j))$. Hence, identical systems (in the sense of their Kendall $\tau$) map to a distance of 0 while reverse ordering yields a maximum distance of 1. Finally, visualizations of classifier adjacency relations are obtained using classical (non-metric) *multidimensional scaling* (MDS). Each classifier is represented by a point in 2D space so that between-classifier distances approximate those in the given distance matrix (here, the $M \times M$ matrix containing all pairwise distances).

# 4. Experimental set-up

We demonstrate the proposed methods with two different tasks: automatic speaker verification (ASV) and voice anti-spoofing. The former consists of classifiers constructed by the authors with controlled parameters. The latter consists of countermeasures submitted to the ASVspoof 2019 challenge. Even if the system descriptions are known, we cannot interact with these systems nor do we know all their implementation nuances.

[2]The converse does not necessarily hold: having the same DET curve or EER does not imply that the trial rankings must be the same.




(a) Raw scores (b) Grouped scores

Figure 2: *Visualisation of ASV similarity on VoxCeleb1 test set. The ASV systems are toy classifiers to show case how the proposed visualisation method behaves in a controlled setting. Shown* $C_{llr}^{min}$ *values serve nominal orientation.*

### 4.1. Automatic speaker verification

The analysis of ASV methods uses the publicly available VoxCeleb dataset. We use the standard trial list of 40 test speakers in VoxCeleb1 [12]. It consists of 18860 target trials and the same number of nontarget trials. We form the groups (see Section 3) from the cross-product of speaker identities in the trials. With 40 speakers, this yields $(40 \cdot 41)/2 = 820$ groups.

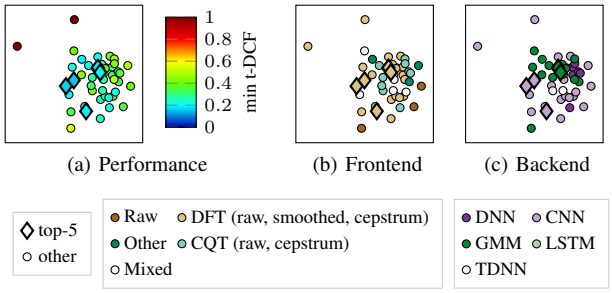
We use Gaussian mixture model-universal background model (GMM-UBM) based ASV system with an MFCCs front-end. While there are better ASV methods, we opt for the computationally light GMM-UBM as we want to generate a large number of classifiers. The focus of this study is in classifier adjacency rather than performance.

Following standard practice, extracted MFCCs are augmented with delta features and processed with RASTA and utterance-level *cepstral mean and variance normalization* (CMVN). For MFCCs, we include filterbanks comprising 12, 16, 20 and 24 filters. For UBMs, we vary the number of components in the powers of 2 between 2 and 1024. For UBM training, we select two different datasets: VoxCeleb2 [4] and LibriSpeech [13]. In total, we have 80 ASV systems.

### 4.2. Anti-spoofing

The analysis of voice anti-spoofing methods, or *spoofing countermeasures* (CMs), uses ASVspoof 2019 challenge submission entries. Details of the dataset [14] and challenge results [1] are provided elsewhere. Here we focus on aspects relevant for the novel classifier adjacency analysis.

The challenge data consists of two different scenarios. The evaluation set of the *logical attack* (LA) scenario contains 13 different text-to-speech or voice conversion attacks (labeled A07...A19). The physical attack (PA) scenario, in turn, consists of simulated replay attacks from 27 different environments and 9 replay configurations. Both sets also contain additional *bona fide* (human speech) utterances. The participants processed the corresponding audio files blindly to obtain detection scores. In the LA scenario, the 3-class evaluation set consists of 5 370 bonafide/target, 1 985 bonafide/non-target, and 63 882 spoofed trials. For the PA scenario, the corresponding numbers are 12 960 bonafide/target, 5 130 bonafide/non-target, and 116 640 spoofed trials. For the LA scenario, the groups (see Section 3) are formed from the 13 attacks plus the bonafide class




(a) Performance (b) Frontend (c) Backend

Figure 3: *CM similarity on ASVspoof 2019 LA test set. The CM systems are single systems (no fusions) submitted by participants; an uncontrolled setting is showcased (raw scores). The ASVspoof 2019 primary metric is shown, the so-called* min t-DCF *[16]. Colours show different frontend transforms & feature types and backend classifiers.*

(a total of 14 groups). For the PA scenario, we used all cross-combinations of 27 environments × (9 replay configurations + 1 bona fide) to define 270 groups.

# 5. Results

The proposed method is demonstrated in a controlled-classifier setting (ASV) and in a setting of unconstrained classifiers (CMs).

## 5.1. Speaker Verification

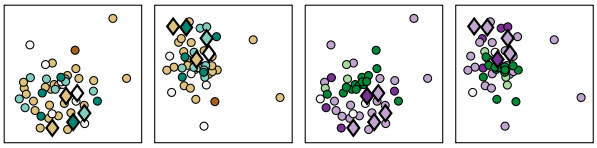
Fig. 2 illustrates ASV results with varying number of filters, mixtures, and training data. The indicated $C_{\text{llr}}^{\text{min}}$ performance [7, 15] is demonstrated for the lowest and highest number of mixture components to provide nominal guidance only. For the top-5 systems, $C_{\text{llr}}^{\text{min}}$ varies between 0.466 and 0.481 (for visual clarity, we do not show all the $C_{\text{llr}}^{\text{min}}$ values in Fig. 2).

Differences in classifier parameters expose observable trends. For 12 filters, decision boundaries resulting from different training data become closer for as few as 4 components. Similar trajectories are followed as the number of components increases (as does the discrimination performance). Then, while the LibriSpeech trajectory continues, the VoxCeleb trajectory makes a sudden jump into the area of the top classifiers (at 512/1024 components and 20/24 filters). Adjacent classifiers (to the top ones) have less filters but more components; the top classifiers use more filters but less components. Our visuals supplement conventional DET plots: they provide insights into how decision boundaries are affected by data and parameters.

## 5.2. Anti-spoofing

Figs. 3 and 4 show visualisations for the LA and PA scenarios, respectively. Note that the location of points in each panel for each dataset (left-to-right) are identical. In each panel, circles indicate a CM, whereas diamonds signify top-5 CMs. Colours indicate performance and frontend/backend meta-data.

The plots give an indication of the diversity among systems. Differences between the top-5 systems are reasonably representative of the full set of systems. The diversity may imply that the top-performing single systems are complementary. This hypothesis is supported by fusion results discussed in [1] which shows that the combination of top-performing single LA and PA systems leads to improved performance. One can presume that the reason for the ineffective fusion strategies for PA primary



(a) Raw/FE (b) Grouped/FE (c) Raw/BE (d) Grouped/BE

Figure 4: *Visualisation of CMs on ASVspoof 2019 PA task; single systems. Frontends (FEs) and backends (BEs) are compared for raw and grouped scores. Legend as in Fig. 3.*

system submissions might be due to lack of complementary of the systems involved in the ensemble.

## 5.3. Adjacency from raw vs. grouped scores

Finally, we address classifier adjacency through score averaging by groups and subsequent use of the proposed method. Figs. 2 and 4 illustrate a comparison of the proposed method for raw and grouped scores. Classifier adjacency is viewed through an alternative lens. In the ASV task, a clear separation between the two training is apparent. The VoxCeleb-trained classifiers (comparatively in-domain) become more alike. By using raw scores, trajectories by number of components (design parameters) are more alike. Feature resolution (the number of filters) corresponds to clear trajectories. In the uncontrolled CM setting, frontends as well as backends are more alike after grouping: the separation of non-GMM and GMM classifiers becomes oblivious.

Originally, we had two motivations for the grouped approach. First, the highly-compressed per-group average was hypothesized to give potentially a stable high-level classifier signature. Second, there is a computational advantage for very large trial sets. However, one needs additional metadata (and domain knowledge) to define the trial grouping. As we see, raw vs. grouped scores produce different findings. We prefer not to recommend either variant without a further study. A related approach, convex optimisation of error trade-offs[3], resonates with the Bayesian decision framework; see use of the ROC convex hull instead of the corresponding 'steppy' profile [7].

# 6. Conclusions

We proposed a simple approach to visualise classifier adjacency on common data. An example application under controlled conditions show that the tool renders classifiers adjacent, when they are similar in metadata. In the absence of unified metadata, analyses for uncontrolled, challenge conditions prove more challenging. The proposed tool nonetheless reveals intriguing, new visual insights into classifier adjacency not provided by any existing tools.

# 7. Acknowledgements

This work was supported by a number of projects and funding sources: VoicePersonae, supported by the French Agence Nationale de la Recherche (ANR) and the Japan Science and Technology Agency (JST) with grant No. JPMJCR18A6; Academy of Finland (proj. 309629); Region Grand Est, France.

[3] In preliminary experiments, we used the proposed method in combination with the *pool adjacent violators* algorithm [17]. System dependently, 37720 ASV scores are reduced to 61 to 98 groups of same rank: visual differences are minimal; we spare comparisons.

# 8. References


[1] A. Nautsch, X. Wang, N. Evans, T. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "Asvspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.

[2] K. A. Lee, O. Sadjadi, H. Li, and D. Reynolds, "Two decades into speaker recognition evaluation - are we there yet?" *Computer Speech & Language*, vol. 61, p. 101058, 2020.

[3] C. S. Greenberg, L. P. Mason, S. O. Sadjadi, and D. A. Reynolds, "Two decades of speaker recognition evaluation at the National Institute of Standards and Technology," *Computer Speech & Language*, vol. 60, p. 101032, 2020.

[4] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[5] D. Ramos, R. Haraksim, and D. Meuwly, "Likelihood ratio data to report the validation of a forensic fingerprint evaluation method," *Data in Brief*, vol. 10, pp. 75–92, 2 2017.

[6] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, and M. A. Przybocki, "The DET curve in assessment of detection task performance," in *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*, G. Kokkinakis, N. Fakotakis, and E. Dermatas, Eds. ISCA, 1997. [Online]. Available: http://www.isca-speech.org/archive/eurospeech_1997/e97_1895.html

[7] N. Brümmer and E. de Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing," [Online] https://sites.google.com/site/bosaristoolkit, AGNITIO Research, South Africa, Tech. Rep., 12 2011.

[8] N. Brümmer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D. dissertation, Faculty of Engineering, Department of Electrical & Electronic Engineering, University of Stellenbosch, 2010.

[9] M. G. Kendall, "A new measure of rank correlation," *JSTOR: Biometrika*, vol. 30, no. 1/2, pp. 81–93, 6 1938.

[10] ——, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945. [Online]. Available: http://www.jstor.org/stable/2332303

[11] W. R. Knight, "A computer method for calculating Kendall's tau with ungrouped data," *Journal of the American Statistical Association*, vol. 61, no. 314, pp. 436–439, June 1966.

[12] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[14] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. W. D. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y. Peng, H. Hwang, Y. Tsao, H. Wang, S. L. Maguer, M. Becker, and Z. Ling, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Comput. Speech Lang.*, vol. 64, p. 101114, 2020. [Online]. Available: https://doi.org/10.1016/j.csl.2020.101114

[15] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[16] T. Kinnunen, H. Delgado, N. Evans, K. A. Lee, V. Vestman, A. Nautsch, M. Todisco, X. Wang, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2195–2210, 2020.

[17] N. Brümmer and J. du Preez, "The PAV algorithm optimizes binary proper scoring rules," 2009.