CHECK YOUR AUDIO DATA: NKULULEKO FOR BIAS DETECTION

Felix Burkhardt[1,2], *Bagus Tris Atmaja*[3], *Anna Derington*[1], *Florian Eyben*[1], *Björn Schuller*[1,4,5]

[1]audEERING GmbH, Germany
[2]Technical University of Berlin, Germany
[3]National Institute of Advanced Industrial Science and Technology, Japan
[4]Chair EIHW, Universität Augsburg, Germany
[5]GLAM, Imperial College London, UK

ABSTRACT

We present a new release of the software tool Nkululeko. New additions enable users to automatically perform sanity checks, data cleaning, and bias detection in the data based on machine learning predictions. Two open-source databases from the medical domain are investigated: the Androids depression corpus and the UASpeech dysarthria corpus. Results show that both databases have some bias, but not in a severe manner.

Index Terms— open-source tool, machine learning, bias detection, speaker characteristics

1. INTRODUCTION AND RELATED WORK

Nkululeko is open-source software written in Python and hosted on GitHub.[1] It is predominantly a framework for audio-based machine learning explorations without the need to write Python code. The main features are: training and evaluation of labelled speech databases with state-of-the-art machine learning approach and acoustic feature extractors, a live demonstration interface, and the possibility to store databases with predicted labels. Based on this, the framework can be used to check on bias in databases by exploring correlations of target labels, like, e. g. *depression* or *diagnosis*, with predicted, or additionally given, labels like age, gender, Short-Time Objective Intelligibility (STOI), Signal-to-distortion ratio (SDR), or mean opinion score (MOS).

Open-source tools are believed to be one of the reasons for accelerated science and technology. They are more secure, easy to customise and transparent. There are several open-source tools that exist for acoustic, sound, and audio analysis, such as librosa [1], TorchAudio [2], pyAudioAnalysis [3], ESPNET [4], and SpeechBrain [5]. However, none of them are specialised in speech analysis with high-level interfaces for novices in the speech processing area.

One exception is Spotlight [6], an open-source tool that visualises metadata distributions in audio data. An existing interface between Nkululeko and Spotlight can be used to combine the visualisations of Spotlight with the functionalities of Nkululeko.

Speech is one of the important modalities of conveying messages. The message contains specific characteristics that the speaker wants to deliver. If there is bias in a database used to train speaker characteristic analysis, this could cause performance degradation of the model built with that database or an unsuitable use case of the database. Bias in data can be manifested through sensitive (possibly seemingly neutral) features that correlate with target variables or through underrepresentation of certain groups [7]. As an additional problem, a machine learning model may use such a bias as a shortcut in its decision-making [8]. A famous example of such a shortcut is a model designed to predict pandemics based on Google searches, which was later found to instead predict seasonality and failed to predict non-seasonal influenza [9]. Ref. [10] showed that a model that distinguishes between huskies and wolves merely based on the presence of snow in an image was trusted by participants before they learnt about the model's shortcut. Ref. [11] examined the gender bias present in the Distress Analysis Interview Corpus – Wizard of Oz (DAIC-WOZ) [12] database and ways to mitigate it.

Contributions of this paper are as follows:

- We report on a tool with novel functionality to investigate bias in audio databases;
- We explore two health-related speech databases as a proof of concept for the approach.

The following Section 2 introduces the Nkululeko framework. Section 3 describes the two databases that have been selected for this paper. In Section 4 we introduce the deep learning models that have been used to predict bias in the data. We follow with a discussion on the results in Section 5 and finish with conclusions and outlook in Section 6.

2. OVERVIEW OF Nkululeko

Nkululeko is a command line tool written in Python, best used

[1]https://github.com/felixbur/nkululeko/

in conjunction with the Visual Studio code editor (but can be run stand-alone). To use it, a text editor is needed to edit the experiment configuration. You would then run Nkululeko like this:

```
python -m nkululeko.explore --config conf.ini
```

and inspect the results afterward; they are represented as images, texts, and even a fully automatically compiled PDF report written in latex.

Nkululeko's data import format is based on a simple CSV formalism, or alternatively, for a more detailed representation including data schemata, audformat.[2] Basically, to be used by Nkululeko, the data format should include the audio file path and a task-specific label. Optionally, speaker ID and gender labels help with speech data. An example of a database labelled with emotion is

```
file, speaker, gender, emotion
x/sample.wav, s1, female, happy
...
```

As the main goal of Nkululeko is to avoid the need to learn programming, experiments are specified by means of a configuration file.[3]

The functionality is encapsulated by software *modules* (interfaces) that are to be called on the command line. We list the most important ones here:

- **nkululeko**: do machine learning experiments combining features and learners
- **demo**: demo the current best model on the command line
- **explore**: perform data exploration (used mainly in this paper)
- **augment**: augment the current training data. This could also be used to reduce bias in the data, for example, by adding noise to audio samples that belong to a specific category.
- **predict**: predict features like SDR, MOS, arousal/valence, age/gender (for databases that miss this information), with Deep neural nets (DNN) models, e. g. as a basis for the *explore* module.

The configuration (INI) file consists of a set of key-value pairs that are organised into several sections. Almost all keys have default values, so they do not have to be specified.

Here is a sample listing of a database section:

```
[EXP]
name = explore-androids
[DATA]
databases = ['androids']
androids = /data/androids/androids.csv
target = depression
labels = ['depressed', 'control']
samples_per_speaker = 20
min_length = 2
[PREDICT]
sample_selection = all
targets = ['pesq', 'sdr', 'stoi', 'mos']
[EXPL]
value_counts = [['gender'], ['age'],
 ↪  ['est_sdr'], ['est_pesq'],
 ↪  ['est_mos']]
[REPORT]
latex = androids-report
```

As can be seen, some of the values simply contain Python data structures like arrays or dictionaries. Within this example, an experiment is specified with the name *explore-androids*, and a result folder with this name will be created, containing all figures and textual results, including an automatically generated Latex and Portable document format (PDF) report on the findings.

The *DATA* section sets the location of the database and specifies filters on the sample, in this case limiting the data to 20 samples per speaker at most and at least 2 seconds long. In this section, the split sets (training, development, and test) are also specified. There is a special feature named *balance splits* that lets the user specify criteria that should be used to stratify the splits, for example, based on SDR.

With the *predict* module, specific features like, for example, SDR or MOS are to be predicted by deep learning models. The results are then used by a following call to the *explore* module to check whether these features, as well as some ground truth features (*age* and *gender*), correlate with the target variable (*depressed* in the given example) in any way.

The Nkululeko configuration can specify further sections:

- **FEATS** to specify acoustic features (e.g. opensmile [13]) or deep learning embeddings (e.g. wav2vec 2.0 [14]) that should be used to represent the audio files.
- **MODEL** to specify statistical models for regression or classification of audio data.

All the images shown in this paper were generated automatically by Nkululeko as a result of the performed investigations.

[2] https://audeering.github.io/audformat/

[3] There is also a blog series on the usage of Nkululeko: http://blog.syntheticspeech.de/?s=nkululeko

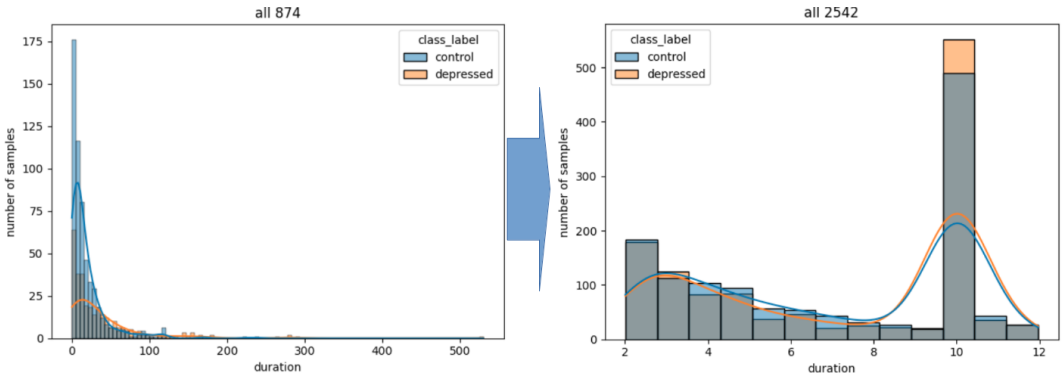

Fig. 1. Distributions of sample duration before (left) and after segmentation.

3. DATABASES

We used the Androids corpus on depression, published at Interspeech 2023 [15], to check potential bias in the data, as well as the UASpeech database [16] focused on dysarthria.

3.1. Androids corpus

The Androids corpus contains audio recordings from interviews as well as read texts. We focused for this paper on the interview task and neglected the read texts by setting a Nkululeko filter. We down-sampled all audio from 44.1 kHz to 16 kHz, because it is required by most machine learning models (Section 4). This can be done automatically by Nkululeko.

We then automatically segmented the database with the Silero [17] voice activity detection (VAD), which is integrated into Nkululeko. It is a deep learning model that has been trained on large corpora that include over 100 languages and performs well on audio files from different domains with various background noise and quality levels.

To document the segmentation, Nkululeko visualises the duration distribution of the samples automatically, which can be seen in Figure 1. Some of the original samples are very long – up to 500 seconds. After segmentation, they are restricted to 10 seconds (an adjustable parameter with Nkululeko), plus two seconds if the end of the sample (and thus the next segment) would have been shorter than two seconds otherwise.

3.2. UASpeech corpus

The UASpeech corpus [16] is a database of dysarthric speech produced by 19 speakers with cerebral palsy. Speech materials consist of 765 isolated words per speaker. The samples are labelled by diagnosis with four labels: *spastic*, *athetoid*, *mixed*, and *unclear*. Obviously, the state of dysarthria might overlap with these labels. Ref. [18] argue that these labels have to be recorded under different acoustic conditions and justify this with a very simple test: after having split the samples with a VAD algorithm into speech and non-speech segments, they found that the non-speech segments can be used as a basis for classification with same or even better results than the speech segments. With respect to the experiments done in this paper, we would expect this to be reflected by an imbalance of the labels with respect to signal quality measures like SDR or MOS. Indeed, we see a significant influence of SDR for the dysarthria category prediction, as discussed in Section 5.

With the size of 138 700 samples, the database is quite large. We reduced the size with Nkululeko by setting the *limit-samples-per-speaker* value to 200, resulting in 2800 samples (19 speakers).

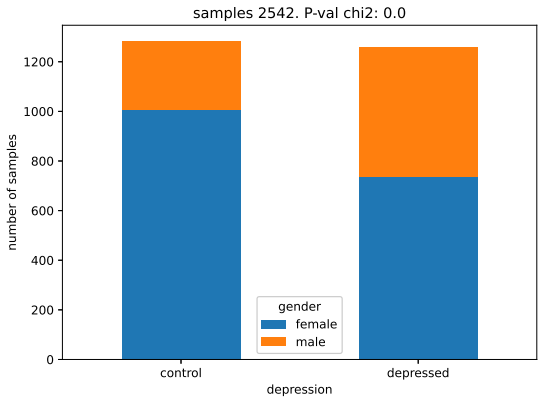

Fig. 2. Distribution of depression per gender for the Androids corpus.

4. PREDICTION MODELS

Besides the ground truth labels that are annotated with the two databases, namely "age" and "gender", we predict several other speech and speaker characteristics with deep learning models:

4.1. SQUIM model

We used the Speech Quality and Intelligibility measures (SQUIM) model [19] to predict the subjective measure, MOS, and two objective ones: (Scale Invariant) SDR and Perceptual Evaluation of Speech Quality (PESQ) predictions. The model works in a reference-less manner and thus does not require a clean sample for comparison. It is based on a multi-task deep learning model that combines dual-path recurrent neural networks (DPRNNs) with transformer layers. Using the DNS Challenge 2020 dataset [20], the model reaches .985 and .958 Pearson's correlation coefficient (PCC) for SDR and PESQ, respectively. The size of the model is 7,387,658 parameters in total.

The performance of the MOS predictions is reported in [21]; it varies strongly with the databases but compares well with the baselines. The PCC for the VoiceMOS-main challenge [22] is .89.

4.2. Emotion model

To predict valence, arousal, and dominance, we utilise the model published by [23], a Wav2vec 2.0 [14] based model fine-tuned for emotion prediction on the MSP-Podcast database [24]. These predictions are mainly interesting for the Androids corpus as it deals with depression – we would expect some correlation with the emotional dimensions. The number of parameters in the model is 165,335,759.

5. RESULTS AND DISCUSSION

We evaluate the influence of the predictions on the targets with the Nkululeko *explore* module depending on the nature of the target and the confounding variables. If the effect of two categorical variables gets investigated, we report χ^2, i. e., the p-value for the significance that the distributions are not independent. For example, in Figure 2, it can be seen that the gender labels are distributed quite unbalanced in the data, which is reflected by a p-value below .1 for the χ^2 test.

In the case of a categorical compared to a numeric variable, we report Cohen's D:

$$d = \frac{\mu_1 - \mu_2}{\sigma},$$

with σ being the pooled standard deviation, and μ_1 and μ_2 being the means of the distributions, to estimate the effect of the predictions on all category combinations. The one with the largest effect size is then reported, in the case of Figure 3, the difference between control and depression. A commonly used interpretation is to refer to effect sizes as small (d = 0.2), medium (d = 0.5), and large (d = 0.8) based on benchmarks suggested by [25; 26]. If both variables are numerical, PCC gets reported – we omit the formula as this is not the case in this data.

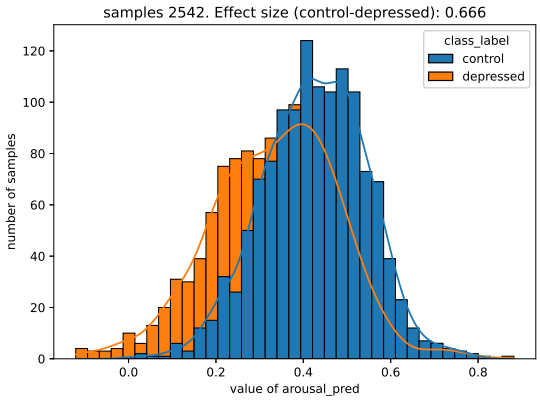

Fig. 3. Distributions of arousal predictions for the Androids.

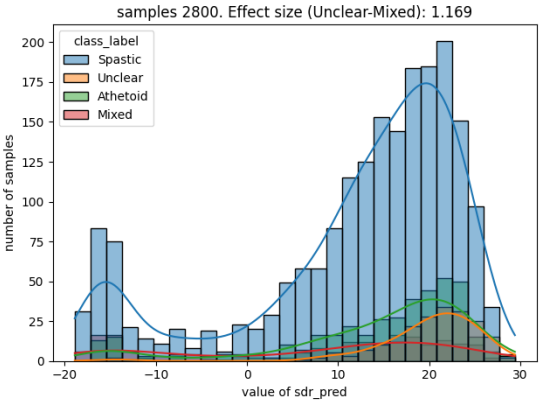

Fig. 4. Influence of SDR on disarthria prediction.

The results for all comparisons are shown in Table 1. Statistical significance is denoted by bold font. The duration of the samples reveals a very large effect with the UASpeech corpus, as the samples labelled with *unclear* are distinctly shorter than the ones with *mixed*. Gender is non-uniformly distributed for both databases, as can be seen, for example, in Figure 2. Estimated PESQ and SDR have quite a strong influence on the dysarthria categories of UASpeech (see Figure 4), again strongest for the difference between *unclear* and *mixed* – perhaps, already a hint to the conspicuousness mentioned in [18]. With respect to the Androids database, there is a clear effect size for arousal predictions and depressed speech, as can also be seen in Figure 3, as well as with valence. Depressed speech has lower valence and arousal than the control data.

In fact, a model using the predictions of the emotion model solely – arousal, valence, and dominance –as features already perform quite well on the classification of depression, as can be seen in Figure 5. We used the folds that are distributed with the original dataset for this experiment and a support vector machines (SVM) classifier with C-val=10.

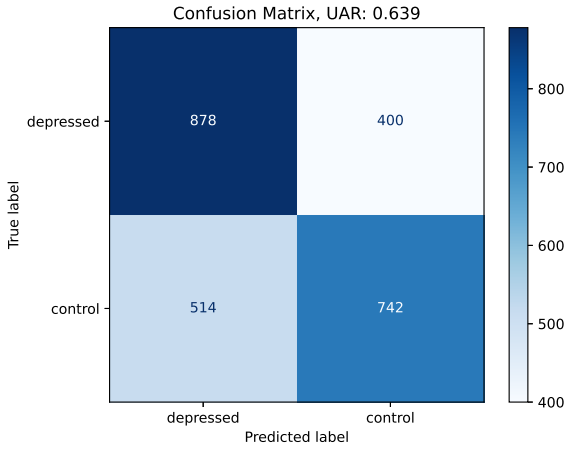

Fig. 5. Confusion matrix for the Androids corpus predicted with arousal, valence and dominance.

Feature	Androids	UASpeech
Duration	.067	Unclear-Mixed: **1.912**
Age	.128	Spastic-Athetoid: .169
Gender: χ^2 p	$\sim$ **0**	$\sim$ **0**
PESQ	.407	Unclear-Mixed: **2.014**
MOS	.459	Unclear-Athetoid: .385
SDR	.019	Unclear-Mixed: **1.169**
Arousal	**.666**	Spastic-Athetoid: .256
Valence: C.d	.415	Athetoid-Mixed: .289

Table 1. Results of the influence of various predicted influencing factors with respect to the target labels. Apart from gender, all values are Cohen's d effect sizes that are at least medium and are in bold.

We get an average (for the five folds) unweighted average recall (UAR) of .639; with wav2vec2 embeddings, we get .797 UAR and with eGeMAPS features even .921 average UAR. When we combine eGeMAPS features with the emotional dimensions as features, we get .928.

6. CONCLUSIONS AND OUTLOOK

In this paper, we introduced new functionalities of Nkululeko, a software framework to do audio-related machine learning experiments without the need to program. We especially focused on bias detection by analysing confounding variables and their correlation with the target. We hope that Nkululeko will be helpful to other authors to check databases at an early stage so they can try to prevent strong bias, for example, by automatically stratifying split sets or augmenting part of the data.

We will continue to develop Nkululeko; the next step with regard to data exploration is the addition of more statistical algorithms to estimate bias in the data.

7. DISCLAIMER

We would like to point out some issues that leave open questions with this paper. First, we are well aware that only two databases are not enough to prove the general usability of the approach, and this will be extended in the future. Secondly, the validity of the claims stated by an automated bias analysis depends on the accuracy of the models that predict bias factors such as gender or MOS. This is challenging as the databases that are to be predicted typically would come from a very different domain than the training data and might even have out of distribution (OOD) issues. Therefore, we would see our approach as a first test that something might not be in order and should be investigated.

8. ACKNOWLEDGMENTS

FB, AD, FE, and BS have received funding from the European SHIFT (MetamorphoSis of cultural Heritage Into augmented hypermedia assets For enhanced accessibiliTy and inclusion) project (Grant Agreement number: 101060660). BTA was supported by projects JPNP20006 commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and JSPS KAKENHI Grant Number 24K02967.

9. REFERENCES

[1] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," *Proc. 14th Python Sci. Conf.*, no. Scipy, pp. 18–24, 2015.

[2] Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and V. Quenneville-Belair, "Torchaudio: Building Blocks for Audio and Speech Processing," in *ICASSP 2022 - 2022 IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2022-May. IEEE, may 2022, pp. 6982–6986.

[3] T. Giannakopoulos, "pyAudioAnalysis: An open-source python library for audio signal analysis," *PLoS One*, vol. 10, no. 12, pp. 1–17, 2015.

[4] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPNet: End-to-end speech processing toolkit," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2018-Septe, no. September, pp. 2207–2211, 2018.

[5] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad,

A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[6] S. Suwelack, "Spotlight," https://github.com/Renumics/spotlight/, 2023.

[7] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis *et al.*, "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1356, 2020.

[8] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[9] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of google flu: traps in big data analysis," *science*, vol. 343, no. 6176, pp. 1203–1205, 2014.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you? explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[11] A. Bailey and M. D. Plumbley, "Gender bias in depression detection using audio features," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 596–600.

[12] J. Gratch, R. Artstein, G. M. Lucas, G. Stratou, S. Scherer, A. Nazarian, R. Wood, J. Boberg, D. DeVault, S. Marsella *et al.*, "The distress analysis interview corpus of human and computer interviews." in *LREC*. Reykjavik, 2014, pp. 3123–3128.

[13] F. Eyben, M. Wöllmer, and B. Schuller, "opensmile – the munich versatile and fast open-source audio feature extractor," *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference*, pp. 1459–1462, 01 2010.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

[15] Fuxiang, A. Esposito, and A. T. Vinciarelli, "The androids corpus: A new publicly available benchmark for speech based depression detection," 2023.

[16] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Proceedings Interspeech*, 2008.

[17] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier," https://github.com/snakers4/silero-vad, 2021.

[18] G. Schu, P. Janbakhshi, and I. Kodrasi, "On using the ua-speech and torgo databases to validate automatic dysarthric speech classification approaches," in *Proceedings ICASSP*, 2023.

[19] A. Kumar, K. Tan, Z. Ni, P. Manocha, X. Zhang, E. Henderson, and B. Xu, "Torchaudio-squim: Reference-less speech quality and intelligibility measures in torchaudio," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[20] C. K. A. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, P. Rana, S. Srinivasan, and J. Gehrke, "Theinterspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," *Proceedings Interspeech*, 2020.

[21] P. Manocha and A. Kumar, "Speech quality assessment through mos using non-matching references," 2022.

[22] W.-C. Huang, E. Cooper, Y. Tsao, H.-M. Wang, T. Toda, and J. Yamagishi, "The voicemos challenge 2022," 2022.

[23] J. Wagner, A. Triantafyllopoulos, H. Wierstorf, M. Schmitt, F. Burkhardt, F. Eyben, and B. W. Schuller, "Dawn of the transformer era in speech emotion recognition: Closing the valence gap," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–13, 2023.

[24] R. Lotfian and C. Busso, "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," *IEEE Transactions on Affective Computing*, vol. PP, pp. 1–1, 08 2017.

[25] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.

[26] B. T. Atmaja and A. Sasou, "Evaluating Self-Supervised Speech Representations for Speech Emotion Recognition," *IEEE Access*, vol. 10, pp. 124 396–124 407, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9964237/