On Breathing Pattern Information in Synthetic Speech

Zohreh Mostaani[1,2] *and Mathew Magimai.-Doss*[1]

[1]Idiap Research Institute, Martigny, Switzerland
[2] Ecole polytechnique fédérale de Lausanne, Switzerland
{zohreh.mostaani, mathew}@idiap.ch

Abstract

The respiratory system is an integral part of human speech production. As a consequence, there is a close relation between respiration and speech signal, and the produced speech signal carries breathing pattern related information. Speech can also be generated using speech synthesis systems. In this paper, we investigate whether synthetic speech carries breathing pattern related information in the same way as natural human speech. We address this research question in the framework of logical-access presentation attack detection using embeddings extracted from neural networks pre-trained for speech breathing pattern estimation. Our studies on ASVSpoof 2019 challenge data show that there is a clear distinction between the extracted breathing pattern embedding of natural human speech and synthesized speech, indicating that speech synthesis systems tend to not carry breathing pattern related information in the same way as human speech. Whilst, this is not the case with voice conversion of natural human speech.

Index Terms: Breathing pattern estimation, Synthetic speech, Neural network, Presentation attack detection

1. Introduction

Speech production system is a combination of several physiological systems such as respiratory system, oral system, and nervous system. There is a close relation between respiration and speech since the lungs provide the necessary energy to produce sounds by pushing air through the vocal folds. There have been studies on the relation between speech and respiration. Winkworth et. al. investigated the association between linguistic factors and lung volumes during read speech [1]. Other studies show that the type of speech can affect the breathing pattern [1, 2, 3]. The breathing can also shape the speech [4]. In recent years, it has been shown that it is possible to predict breathing patterns from speech signals [5].

Besides natural speech produced by humans, advances in speech technology also have made it possible to generate speech. Text-to-speech synthesis (TTS) methods have evolved over time. For a long time concatenative TTS [6] and statistical parametric TTS [7] were the main methods of generating speech from text but recently there has been a shift towards deep learning based methods [8, 9]. The speech generated by deep learning based methods are reportedly very natural sounding and in some cases are indistinguishable from human speech.

A research question that arises is: whether synthetic speech carries breathing related information in the same way as natural human speech? Besides scientific curiosity, answer to this research question is of potential interest to other complementary research directions, such as, (a) TTS systems can be used to fake identity, e.g. presentation attack on automatic speaker verification systems [10] and (b) TTS is being explored for synthesizing speech with pathological conditions to develop objective pathological speech methods [11]. In pathological speech such as dysarthric speech, breathing phenomenon is intrinsically related to impaired speech production [12, 13]. This paper aims to address the aforementioned research question by leveraging two different research directions, namely, speech-based breathing pattern estimation and detection of logical-access presentation attack.

The remainder of the paper is organized as follows. Section 2 introduces our study design. Section 3 and Section 4 present the experimental setup and the results with analyses. Section 5 finally concludes the paper.

2. Study design

In recent years, with advances in deep learning, speech-based breathing pattern estimation methods have emerged. In these methods, either raw waveform or short-term spectral feature is input and the output of the neural network predicts breathing signal. Nallanthighal et al. used a Convolutional Neural Network (CNN) and a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) with log Mel Spectrogram of speech as input to predict the breathing signal [14, 15]. Approaches with raw speech waveform as input to CNN or RNN have been also developed for predicting breathing signals [5, 16, 17]. In [5], it was demonstrated that breathing signals can be estimated in a cross-database or cross-domain manner, e.g., training on one database and testing on another database.

In this work, we build upon these advances to investigate the aforementioned research question using pre-trained breathing pattern estimation neural networks. However, one issue is that to evaluate the output breathing pattern we need a reference breathing pattern to compare to, which in the case of human speech production can be measured through sensors but not in the case of synthetic speech. In recent works, it has been demonstrated that pre-trained breathing pattern estimation neural networks can be employed for other speech tasks such as, speech-based COVID detection [18, 19]. In [18], a pre-trained encoder network is used to estimate breathing patterns from the speech signal, which is then passed to a decoder network for COVID detection from cough audio. In [19], it has been demonstrated that neural embeddings extracted from pre-trained breathing pattern estimation neural networks could be used for COVID detection. We take inspiration from these works to recast the research question as: whether natural human speech and synthetic speech can be distinguished based on embeddings extracted from pre-trained breathing pattern estimation neural networks. The underlying hypothesis being that: should synthetic speech exhibit breathing pattern information similar to natural human speech then the two speech signals will not be easily distinguishable.

Figure 1 illustrates our framework. The breathing pattern embeddings (BPE) extracted from a pre–trained neural network are used to train a binary classifier, natural versus synthetic speech. We investigate two classification approaches. In the first approach, frame level neural embeddings are classified using a multi layer perceptron (MLP). The output class probabilities are averaged over the utterance and a decision is made. In the second approach, the embeddings are aggregated using functionals (mean, standard deviation) or bag-of-audio-words (BoAW) representation to obtain an utterance-level fixed length representation and then classified using classifiers such as, support vector machine (SVM) and random forest (RF).

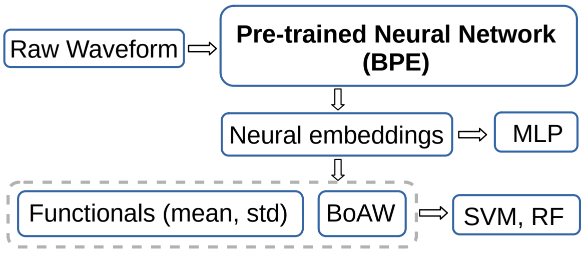


Figure 1: *Framework to distinguish natural human speech and synthetic speech based on breathing pattern embeddings.*

To investigate this question, we leverage from the automatic speech verification community's effort in developing anti-spoofing methods to detect logical-access attacks generated using TTS systems and voice conversion (VC) systems through organization of ASVspoof challenge [20, 21]. Besides well-defined protocols and employing state-of-the-art approaches to generate logical-access attacks, the ASVSpoof challenge provides the means to systematically investigate the research questions. First, there are two types of TTS systems [21]: (a) synthetic speech purely generated using neural models trained on speech and textual data (neural TTS) and (b) synthetic speech generated by concatenating segments of natural human speech waveforms (concatenative synthesis). Second, there are also attacks generated through voice conversion alone (VC-alone). The VC-alone system takes a natural human speech signal as input and converts it to the target speaker's voice by altering the source and system information. So, as a by-product, it allows us to investigate whether such alterations done on a single speaker speech affect the breathing pattern related information.

3. Experimental setup

This section first presents the ASVSpoof 2019 database and protocol. Next presents extraction of neural embeddings using pre-trained breathing pattern estimation neural networks, and finally the development of binary classifiers (natural human speech vs. synthetic speech).

3.1. Database and protocols

We used the ASVSpoof2019 challenge [21] database for our investigation. The ASVSpoof2019 challenge provides presentation attacks for two use case scenarios: logical-access (LA) and physical-access (PA). Our investigation focused on the LA scenario in which attacks are generated using TTS and VC technologies. The database includes three sets, namely training, development, and evaluation which comprise of speech from 20 (8 male, 12 female), 10 (4 male, 6 female) and 48 (21 male, 27 female) speakers respectively. The training set includes 2580 bonafide and 22800 spoofed utterances while the development set comprises 2548 bonafide and 22296 presentation attacks. The evaluation set includes 7355 bonafide and 63882 presentation attacks.

The database includes bonafide and presentation attacks generated by 17 different TTS and VC systems. From these systems, 6 are considered as known attacks which are the only presentation attacks present in the training and development sets. The remaining 11 are considered unknown attacks. The 11 unknown attacks and 2 of the known attacks comprise presentation attacks available in the evaluation set. The VC systems use neural-network-based and spectral-filtering-based approaches [22]. Concatenation-based and neural-network-based systems are used for TTS systems with different vocoders [8, 23]. The TTS-VC systems use various waveform generation methods such as Griffin-Lim [24] and generative adversarial networks [25] among others.

In all our experiments, we followed the protocols as per ASVSpoof2019 challenge, i.e., the training and development sets are used for training the binary classifiers and evaluation is carried out on the evaluation set. We use equal error rate (EER) and area under the receiver operating characteristic curve (AUC) as the evaluation measures.

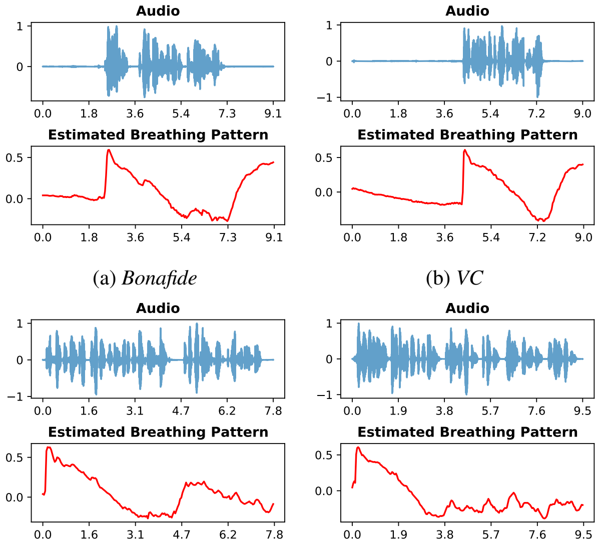
3.2. Neural embeddings based feature representation

In our previous work, we had developed CNN-based breathing pattern estimation neural network that maps raw speech signal to breathing pattern [5, 26]. The CNNs consist of four convolution layers, one hidden layer with ten nodes and one output unit. The architecture of the networks is summarized in Table 1. These CNNs were trained on two different databases: (a) Philips databases [14] consisting of read speech and (b) UCL Speech Breath Monitoring (UCL-SBM) database consisting of conversational speech, provided as part of Interspeech 2020 ComParE challenge [16]. Besides different databases, different CNNs were trained with different lengths of speech input (2 seconds - 4 seconds) and different loss functions. In that study, we found that the performance of the CNNs in breathing pattern estimation were comparable. So, for the present study, we chose two CNNs each trained on Philips database and UCL-SBM database with mean squared error loss function, one with 2 seconds speech as input and the other with 3 seconds speech as input. Figure 2 shows the estimated breathing pattern output by the 3 seconds input CNN pre-trained on the Philips database for a bonafide speech, VC attack, TTS attack and TTS_VC attack from the ASVSpoof2019 database. When extracting the 10-dimensional neural embeddings from the hidden layer (before activations), for the utterances with duration of shorter than twice the size of the input window (i.e., 2 seconds or 3 seconds), we simply repeated the whole utterance and included them in our study to avoid changing the ASVSpoof2019 challenge protocol. This is acceptable as we are not interested in extraction of breathing patterns in an absolute sense.

As mentioned earlier, the neural embeddings are modeled by different classification techniques. In the case of the MLP classifier, no further processing was needed. In the case of fixed length representation using functionals, denoted as $f_{\mu\sigma}$(BPE), utterance level mean and standard deviation were computed and concatenated to obtain a 20-dimensional representation. In the case of BoAW-based fixed length representation, denoted as $BoAW$(BPE), we used the openXBOW toolkit [27] with a codebook size of 100 to obtain 100 dimensional BoAW representation.

Table 1: *Breathing pattern estimation CNN architecture*

	Components	Details
4	Convolution batch_normalization activation (Relu) max-pooling	number of filters in convolution layers: 128-256-512-512 kernel size: 30-10-4-3 kernel strides: 10-5-2-1. strides of max-pooling layers: 2-3-1-1
1	Hidden layer batch normalization activation (Tanh)	number of nodes: 10
1	output (Linear)	number of nodes: 1



(a) *Bonafide* (b) *VC*

(c) *TTS* (d) *TTS_VC*

Figure 2: *Estimated breathing pattern with a CNN pre-trained on Philips database with input speech window length of 3 seconds for different examples from ASVSpoof2019 database with natural and synthetic speech.*

3.3. Classification framework

Three different classifiers were trained with the features obtained from the CNNs as explained in 3.2. The 10-dimensional frame level embeddings denoted as BPE were classified using an MLP with two fully connected layers. The input layer consisted of 10 nodes. The first and second hidden layers consisted of 128 and 64 nodes, respectively. The MLP was trained using the binary cross entropy loss function and the Adam optimizer [28] with a learning rate of 0.001. The system was implemented using Pytorch [29] framework.

The utterance level embeddings denoted as $f_{\mu\sigma}$(BPE) and $BoAW$(BPE) were modeled by an SVM with linear kernel and a random forest classifier (RF). The grid search methodology integrated in the *Scikit-learn* [30] toolkit with AUC as optimization criterion was used to find the optimized parameters on the development data. The SVM was tuned for different values of the regularization parameter C. For RF classifier, the parameters for grid search were as following: number of estimators $\{50$, 500, 1000, $2000\}$, maximal number of features $\{$“auto”, “sqrt”, “log2”$\}$, criterion $\{$“gini”, “entropy”$\}$, and minimal samples leaf $\{1, 2, 4\}$. In all cases we used the *StandardScalar* method of *Scikit-learn* for normalizing the data.

4. Results and analysis

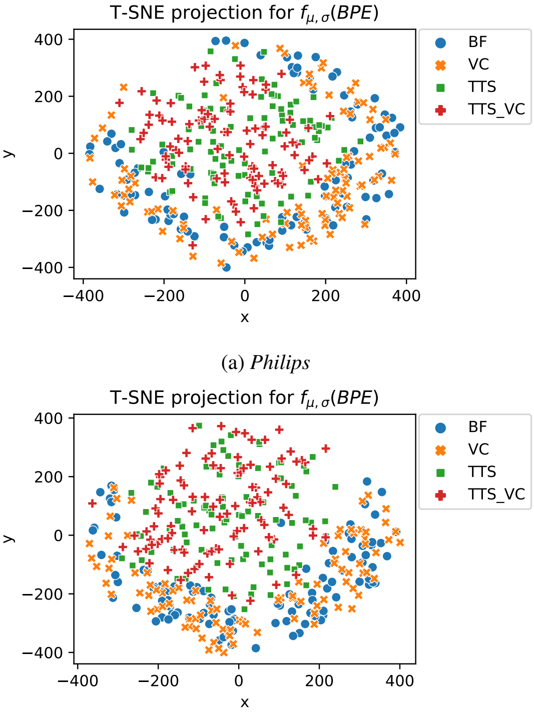
Table 2 shows the AUC and EER in percentage on the evaluation set for the features obtained from CNNs pre-trained on both Philips and UCL_SBM databases. The system performance over all the samples in the evaluation set is presented under the column “All”. The results under the other columns are reported over a subset of the evaluation data with all the bonafide files and only specific types of presentation attacks, namely VC, TTS, and TTS_VC.

Table 2: *The AUC and EER in percentage on the evaluation set for embeddings obtained from CNNs pre-trained on the Philips and UCL_SBM database with input speech window length of 3 seconds and 2 seconds. Column “All” presents the system performance over all the evaluation data while the results under other columns are reported over a subset of evaluation data with all the bonafide files and only the presentation attacks with the type mentioned as the title of the column. VC stands for voice conversion, TTS for Text-to-speech, and TTS_VC is a combination of the two.*

Features	Classifier	Measure	All	VC	TTS	TTS_VC
Embeddings from CNN pre-trained on Philips database						
3 seconds speech input						
BPE	MLP	AUC	90.35	59.92	99.4	99.7
		EER	16.88	42.75	2.53	1.32
$f_{\mu\sigma}$(BPE)	SVM	AUC	89.51	59.42	98.42	98.78
		EER	16.98	43.54	4.29	3.48
	RF	AUC	90.65	62.44	98.93	99.54
		EER	17.02	41.02	4.22	2.6
$BoAW$(BPE)	SVM	AUC	89.35	61.43	97.54	98.17
		EER	17.69	41.62	7	6.01
	RF	AUC	90.86	62.72	99.16	99.62
		EER	17.69	40.04	4.14	2.5
2 seconds speech input						
BPE	MLP	AUC	84.56	47.94	95.08	96.63
		EER	21.5	51.59	10.89	8.72
$f_{\mu\sigma}$(BPE)	SVM	AUC	87.52	57.92	95.85	97.7
		EER	20.08	44.39	10.63	7.49
	RF	AUC	89.15	56.68	98.61	99.55
		EER	18.23	45.41	5.25	2.7
$BoAW$(BPE)	SVM	AUC	88.18	52.17	98.91	99.18
		EER	19.28	48.57	4.24	2.97
	RF	AUC	88.04	51.14	99.01	99.34
		EER	19.51	48.46	4.61	3.2
Embeddings from CNN pre-trained on UCL_SBM database						
3 seconds speech input						
BPE	MLP	AUC	90.02	58.33	99.43	99.73
		EER	17.27	43.65	2.56	1.65
$f_{\mu\sigma}$(BPE)	SVM	AUC	90.23	59.86	99.2	99.66
		EER	17.48	42.96	3.62	2.35
	RF	AUC	90.76	60.85	99.64	99.93
		EER	17.29	41.77	1.6	0.59
$BoAW$(BPE)	SVM	AUC	90.11	58.32	99.58	99.8
		EER	17.29	44.44	2.75	1.74
	RF	AUC	90.5	60.11	99.49	99.9
		EER	17.07	42.62	2.46	0.63
2 seconds speech input						
BPE	MLP	AUC	89.84	60.24	98.42	99.41
		EER	17.48	43.14	5.56	3.08
$f_{\mu\sigma}$(BPE)	SVM	AUC	88.07	58.28	97.25	96.45
		EER	18.97	43.92	8.67	9.44
	RF	AUC	88.48	54.97	98.55	98.46
		EER	18.71	47.1	5.85	6.25
$BoAW$(BPE)	SVM	AUC	89.25	55.91	99.21	99.34
		EER	17.69	45.79	3.53	3.03
	RF	AUC	90.01	57.92	99.62	99.7
		EER	17.19	43.72	2.49	2.44

When the whole evaluation set is taken into consideration (i.e., “All” column), it can be observed that irrespective of the database on which the CNNs are trained, input speech length window or type of classifier, the AUC ranges between 84.56% and 90.86% and the EER ranges between 16.88 % and 21.5 %. Looking into the results segregated in terms of the attack types reveals that TTS and TTS_VC can be classified relatively easier based upon BPE than VC. We achieve AUCs as high as 99.64% and 99.93% with EERs as low as 1.6% and 0.59% for TTS and TTS_VC attacks, respectively, while a best AUC of 58.32% and EER of 44.44% is achieved for VC attack. The performance for VC is close to the chance level.

We observe the same trend whether we classify a single BPE frame through an MLP and aggregate the output probabilities or first aggregate the BPE into an utterance level fixed length through computation of first order and second order statistics or BoAW representation and then classify with SVM or RF. So, to get an insight into the BPE space, we generated a T-SNE [31] projection visualization for $f_{\mu\sigma}$(BPE) features extracted from CNNs with 3 seconds speech input pre-trained on Philips and UCL_SBM databases. Figure 3 presents the T-SNE projection. It can be observed that in both cases the samples from bonafide and VC are grouped together (blue circles and orange crosses) and TTS and TTS_VC are grouped together (green squares and red pluses). Furthermore, even though the distribution is different between the embeddings extracted from the two CNNs, there is a good separation between the two groups. It is worth mentioning that the T-SNE projections with 2 seconds input CNNs also yielded the same observations (not presented due to space limitations). The T-SNE visualizations reaffirm the observations made from the results presented in the table above.



(a) *Philips*

(b) *UCL_SBM*

Figure 3: *TSNE projection of $f_{\mu\sigma}$(BPE) embeddings extracted from CNNs pre-trained on (a) Philips and (b) UCL_SBM database with 3 seconds input speech.*

We further analyzed the AUC and EER per attack on the evaluation set by computing median value for AUC and EER obtained with both Philips and UCL_SBM CNNs trained with 3 seconds speech input. Table 3 presents median values along with the range of EER and AUC per attack and contrasts with the median EER reported in the ASVSpoof2019 challenge for all the systems in [21]. It can be observed that the systems in ASVSpoof2019 challenge have yielded high EER on attacks A10 (TTS), A13 (TTS_VC) and A17 (VC). In our case, high EERs are only observed for VC attacks. It can be also noted that on some attacks, namely, A10, A13, A14 and A15, the median EERs in our study are lower than the median EER obtained in ASVSpoof2019 challenge.

Table 3: *The median values for the AUC and EER for our systems and the median values for the EER of the systems presented in ASVSpoof2019 challenge. The values are presented in percentage. The numbers in brackets are the range of EER for our systems.*

	Attack type	Our study		ASVSpoof2019
		AUC	EER	EER [21]
A07	TTS	99.75	1.48 [0.38 - 5.39]	0.02
A08	TTS	98.65	4.76 [2.38 - 7.55]	0.09
A09	TTS	99.3	3.31 [0.45 - 12.36]	0.06
A10	TTS	99.74	1.39 [0.74 - 5.44]	12.21
A11	TTS	99.68	1.64 [0.79 - 5.35]	0.59
A12	TTS	98.85	4.6 [1.6 - 10.13]	3.75
A13	TTS_VC	99.75	1.31 [0.31 - 7.22]	12.41
A14	TTS_VC	99.67	2.13 [0.73 - 4.82]	2.88
A15	TTS_VC	99.68	1.79 [0.53 - 4.78]	3.22
A16	TTS	99.32	3.27 [1.69 - 5.37]	0.02
A17	VC	52.96	48.07 [42.93 - 51.22]	15.93
A18	VC	61.82	41.51 [38.9 - 45.38]	5.59
A19	VC	65.8	38.25 [35.66 - 39.74]	0.06

5. Conclusions

In this paper, we investigated whether synthetic speech carry breathing pattern related information in the same way as natural human speech. We investigated this question by conducting a study on ASVSpoof2019 challenge to distinguish between bonafide speech and attacks generated through TTS, combination of TTS and VC, and VC using breathing pattern embeddings estimated using networks pre-trained on two different databases, one with read speech and one with conversational speech. Our results and analyses consistently showed that attacks based on TTS speech and TTS_VC speech can be detected in a highly accurate manner when compared to attacks based on VC-alone. This indicates that, irrespective of the TTS approach i.e. whether concatenative synthesis or neural TTS, the generated synthetic speech tends to not carry breathing pattern related information in the same way as natural human speech. Furthermore, the findings also indicate that the alterations done to the natural human speech signal during voice conversion is not strongly altering the breathing pattern related information. Our future work will further investigate this point along with investigations on HMM-based TTS [7] and Diphone synthesis [32].

6. Acknowledgements

This work was funded by the Swiss National Science Foundation through the project Towards Integrated processing of Physiological and Speech signals (TIPS), grant no. 200021_188754.

7. References

[1] A. L. Winkworth, P. J. Davis, E. Ellis, and R. D. Adams, "Variability and consistency in speech breathing during reading: Lung volumes, speech intensity, and linguistic factors," *Journal of Speech, Language, and Hearing Research*, vol. 37, no. 3, pp. 535–556, 1994.

[2] Y.-T. Wang, J. R. Green, I. S. Nip, R. D. Kent, and J. F. Kent, "Breath group analysis for reading and spontaneous speech in healthy adults," *Folia Phoniatrica et Logopaedica*, vol. 62, no. 6, pp. 297–302, 2010.

[3] A. Henderson, F. Goldman-Eisler, and A. Skarbek, "Temporal patterns of cognitive activity and breath control in speech," *Language and Speech*, vol. 8, no. 4, pp. 236–242, 1965.

[4] M. Włodarczak and M. Heldner, "Respiratory Constraints in Verbal and Non-verbal Communication," *Frontiers in Psychology*, vol. 8, May 2017.

[5] V. S. Nallanthighal, Z. Mostaani, A. Härmä, H. Strik, and M. Magimai-Doss, "Deep learning architectures for estimating breathing signal and respiratory parameters from speech recordings," *Neural Networks*, vol. 141, pp. 211–224, 2021.

[6] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[7] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[8] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[9] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Proceedings of Interspeech*, 2017, pp. 4006–4010.

[10] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, 2015.

[11] B. Halpern, J. Fritsch, E. Hermann, R. Van Son, O. Scharenborg, and M. Magimai.-Doss, "An objective evaluation framework for pathological speech synthesis," in *Proceedings of ITG Conference on Speech Communication*, 2021.

[12] J. R. Duffy, *Motor speech disorders: Substrates, differential diagnosis, and management*. St. Louis, MO: Elsevier, 2013.

[13] P. Enderby, "Frenchay dysarthria assessment," *British Journal of Disorders of Communication*, vol. 15, no. 3, pp. 165–173, 1980.

[14] V. S. Nallanthighal, A. Härmä, and H. Strik, "Deep Sensing of Breathing Signal During Conversational Speech," in *Proceedings of Interspeech*, 2019, pp. 4110–4114. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1796

[15] V. S. Nallanthighal, A. Härmä, and H. Strik, "Speech breathing estimation using deep learning methods," in *Proceedings of ICASSP*, 2020, pp. 1140–1144.

[16] B. W. Schuller, A. Batliner, C. Bergler, E.-M. Messner, A. Hamilton, S. Amiriparian, A. Baird, G. Rizos, M. Schmitt, L. Stappen, H. Baumeister, A. D. MacIntyre, and S. Hantke, "The INTERSPEECH 2020 Computational Paralinguistics Challenge: Elderly Emotion, Breathing & Masks," in *Proc. Interspeech 2020*, 2020, pp. 2042–2046.

[17] M. Markitantov, D. Dresvyanskiy, D. Mamontov, H. Kaya, W. Minker, and A. Karpov, "Ensembling End-to-End Deep Models for Computational Paralinguistics Tasks: ComParE 2020 Mask and Breathing Sub-Challenges," in *Proceedings of Interspeech*, 2020, pp. 2072–2076.

[18] G. Deshpande and B. W. Schuller, "The DiCOVA 2021 Challenge — An Encoder-Decoder Approach for COVID-19 Recognition from Coughing Audio," in *Proceedings of Interspeech*, 2021, pp. 931–935.

[19] Z. Mostaani, R. Prasad, B. Vlasenko, and M. Magimai-Doss, "Modeling of pre-trained neural network embeddings learned from raw waveform for covid-19 infection detection," in *Proceedings of ICASSP*, 2022, pp. 8482–8486.

[20] "ASV Spoof," https://www.asvspoof.org/, accessed: 2022-03-26.

[21] 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 *Proceedings of Interspeech*, 2019, pp. 1008–1012.

[22] D. Matrouf, J.-F. Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in *Proceedings of ICASSP*, vol. 1, 2006, pp. I–I.

[23] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[24] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[25] K. Tanaka, T. Kaneko, N. Hojo, and H. Kameoka, "Synthetic-to-natural speech waveform conversion using cycle-consistent adversarial networks," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 632–639.

[26] Z. Mostaani, V. Srikanth Nallanthighal, A. Härmä, H. Strik, and M. Magimai-Doss, "On the relationship between speech-based breathing signal prediction evaluation measures and breathing parameters estimation," in *Proceedings of ICASSP*, 2021, pp. 1345–1349.

[27] M. Schmitt *et al.*, "openxbow–introducing the passau open-source crossmodal bag-of-words toolkit," *Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, 2017.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository (CoRR)*, vol. abs/1412.6980, 2015.

[29] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[30] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[31] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[32] K. A. Lenzo and A. W. Black, "Diphone collection and synthesis," in *Proceedings of ICSLP*, 2000, pp. 306–309.