Phoneme Discretized Saliency Maps for Explainable Detection of AI Generated Voice

Anonymous submission to Interspeech 2024

Abstract

In this paper, we propose 'PDSM: Phoneme Discretized Saliency Maps', a discretization algorithm for saliency maps that takes advantage of phoneme boundaries for explainable detection of AI generated voice. We experimentally show with two different Text-to-Speech systems (Tacotron2, Fastspeech2) that the proposed algorithm produces saliency maps that result in more faithful explanations compared to standard posthoc explanation methods. Moreover, by associating the saliency maps to the phoneme representations, this methodology results in explanations that tends to be more understandable than saliency maps on magnitude spectrograms.

Index Terms: Deep Fake Detection, Explainable Machine Learning, Posthoc interpretability

1. Introduction

In recent years, generative-AI has reached impressive levels of realism across a variety of domains [1]. This includes text [2], conversational text [3], images [4, 5], and applications in the speech domain such as voice cloning / text-to-speech [6–8].

While these advanced generative modeling methods have relatively benign applications in areas such as education, movie production, or the video gaming industry, unfortunately there also exists a plethora of malevolent use cases. These range from students cheating on homeworks using ChatGPT [9], to fakenews generation [10], using deepfakes in order to instigate scandals [11], to using AI-generated voice for scam calls [12], and generating fake product reviews [13]. The perception in the general public against generative AI has reached to a point where Time magazine has ranked generative AI as the third biggest global threat for 2023, and a significant threat to democracies around the world [14] as they argue that generative AI could be used to manipulate public perception in favor of malevolent actors.

The generative model techniques are slated to improve even further with the relentless pace of advancement in the machine learning literature, and therefore it is natural to expect that the advances in generative-AI technology will only exacerbate the social risks in the upcoming years. This is especially risky when human voices are replicated, as malevolent actors can use faked human voices for various malicious acts.

It is thus essential to develop counter measures against systems that generate human voice. One such counter measure is to use machine learning to automatically detect AI generated voice. Detecting AI generated voice with machine learning is a developing field. Existing works include [15], which performs a bispectral analysis on the input audio to classify whether the input audio is generated by a TTS system or not. Another system in the literature performs classification by using weights of an ASR system [16]. Also, ASVSpoof 2021 is a popular spoofed voice detection challenge, where the top performing systems are GMM based light weight convolutional network, [17], and rawnet2 [18]. More recent works also include a spatio-temporal graph attention network [19], using pretrained models [20], a self-distillation framework [21], and a speaker verification model that aims to compare biometric characteristics [22].

Even though these methods give high accuracy in detecting AI generated voice, their predictions are not interpretable or explainable, and therefore they remain black-box models which are opaque for end users. In this work, our focus is therefore particularly on faked voice detection, in a posthoc interpretable manner (i.e. producing interpretations for trained networks). A common way to process audio / speech signals is to use time-frequency representations such as the short-time-fourier-transform or mel-frequency spectrograms. A straightforward approach for producing explanations is to apply standard posthoc interpretation methods such as Integrated Gradients [23], GradientSHAP [24], or Layerwise Relevance Propagation [25] on a magnitude spectrogram. One such study has been conducted by [26] for generated voices, however as we qualitatively and quantitatively show in this paper, producing heatmaps with standard posthoc interpretations typically results in interpretations that are difficult to understand.

As we discuss in the next section, the difficulty in producing interpretations for AI generated voice stems from the fact that preconceived notions for what constitutes an an acceptable explanation is absent. In this paper, we therefore propose **P**honeme **D**iscretized **S**aliency **M**aps (PDSM), which expresses the produced interpretations in terms of phonemes where the phoneme boundaries are extracted with an ASR model through a posterior distribution over phonemes for each time point [27]. More specifically we show that,

- PDSM yields a more faithful explanation for the neural network decisions compared to standard saliency methods. In other words, PDSM explanations follow the neural network decisions more closely.
- PDSM expresses the neural network decision in terms of phonemes, and therefore is able to produce more understandable explanations compared to standard saliency methods.

2. Posthoc Interpretations for Audio

Magnitude spectrograms are a very common input domain representation for classifying audio signals. There exists several works in the literature that aim to produce interpretations on magnitude spectra [28–30]. The goal is to identify time-frequency regions in the spectrogram that maximally contribute to the classifier decision.

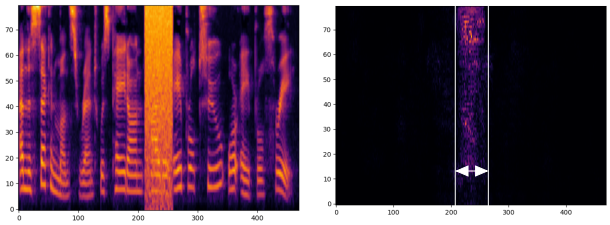

Figure 1: ***(left)*** *Input Spectrogram,* ***(right)*** *Interpretations obtained for a trained network with the GradSHAP algorithm. We overlay the boundaries of the time-limited Gaussian noise on top of the interpretations.*

In image domain, the literature mainly deals with cases where the target object is clearly defined, and in such cases the posthoc interpretation methods are able to produce interpretation results that are placed on the target object [23, 31–33].

In audio domain, below we show an analogous example, where the 'target object' is clearly defined. We contaminate LJSpeech [34] utterances with time-limited standard Gaussian noise (we add this noise in the time domain, and the time boundaries are randomly sampled), and then train a CNN14 architecture to detect the presence of this noise. We then run a posthoc interpretation method for this trained classifier to highlight regions of the spectrogram which are deemed to be maximally important for the classifier. For this particular example, we used the GradSHAP algorithm [24], but note that another posthoc interpretation algorithm could be used also to demonsrate the same thing. We show these results in Figure 1, where we observe that in this simple case the posthoc interpretation method is able to clearly identify the cause of the decision when applied on a trained network (right panel).

2.1. Faithfulness of the interpretations

In order to evaluate how much does the interpretation reflect the salient causes which trigger the classifier decision, in [28], the authors introduced a faithfulness metric which is calculated by measuring the drop in class-specific output probabilities, when the masked out portion of the input is fed to the classifier. This amounts to calculating,

$$\text{FF}_n := f(X_n)_c - f(X_n \odot (1 - M))_c, \tag{1}$$

where $f(.)_c$ denotes the classifier output for class c, and M denotes the saliency map estimated by the interpretation method. If this metric is large, this signifies that the parts of the spectrogram highlighted by the interpretation method are highly influential for the classifier decision for class c, as the second term removes the parts of the input which are highlighted by the interpretation mask M. However, we notice that even when the saliency maps make intuitive sense, the faithfulness metric might have a very low value. For instance, we see from Figure 1 that the produced interpretations point towards the part of the spectrogram that corresponds to the Gaussian noise which the classifier is trained to detect, and therefore, intuitively, this should result in large faithfulness value. However, the naive application of this metric typically results in poor faithfulness value because it fails to remove the gaussian noise entirely, which the classifier is able to detect in the masked-out input (i.e. $X \odot (1-M)$). This can be potentially mitigated by discretizing the continuous saliency map to fill its own bounding box shown in the right panel of Figure 1, thereby accurately encapsulating the target object. The core idea of the method proposed in this paper is to associate the high energy regions in the saliency maps with time boundaries that accurately align with a notion of a 'target object', namely phonemes.

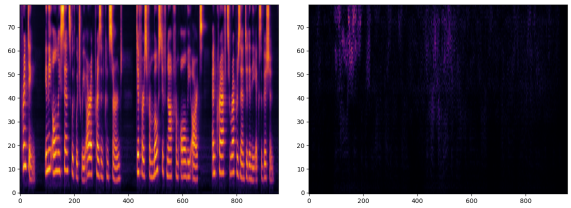

Figure 2: ***(left)*** *Input Spectrogram for an AI generated voice,* ***(right)*** *GradSHAP interpretations obtained for a CNN14 trained network*

As we show in the next section, in more complicated problems such as detection of AI generated voice, where the 'target object' is not clearly defined, evaluating the faithfulness of the interpretations and understanding the interpretations themselves prove to be difficult. We need techniques to be able to i) increase the understandability of interpretations ii) measure the faithfulness accurately. For this purpose we propose to discretize the continuous saliency maps by using phoneme boundaries that encapsulate high-energy portions of saliency maps. We show that such discretization not only enhances understandability of interpretations, but also result in more faithful saliency maps.

2.2. Producing Interpretations for AI Generated Voice

We now consider producing posthoc interpretations for a classifier that detects AI generated voice. For this purpose, we created a deepfake dataset using LJSpeech. We used $13,000$ utterances ($10,000$ train and $3,000$ test) from the LJSpeech Dataset as bonafide (real) speech. We generate $13,000$ fake utterances using the Tacotron2 model [35] and another $13,000$ fake utterances using Fastspeech2 model [36]), along with a HiFiGAN vocoder. We used a popular open source speech package - SpeechBrain [37] for this purpose. We trained two CNN14 models [38] to distinguish between real and generated speeches, one for Tacotron2 and another for Fastspeech2. Both classifiers are able to obtain a detection accuracy very close to 100% on the test set.

In Figure 2, we show GradSHAP interpretations obtained for an AI generated voice correctly classified by our classifier. We observe that the interpretations produced for the sample (right panel) are unlike the time-localized Gaussian noise example, where the 'target object' is clearly defined, so in this case it is hard to ascertain i) the exact meaning of the produced interpretations, ii) if the produced interpretations are indeed faithful. We now propose Phoneme Discretized Saliency Maps, and show how they help us alleviate both these issues.

3. Phoneme Discretized Saliency Maps (PDSM)

In this section, we propose PDSM, a method that discretizes saliency map interpretations produced by attribution methods such as Integrated Gradients (IG), GradSHAP, and others, using phoneme posteriorgram (PPG) representations.

The phoneme posteriorgrams [27] provide a posterior dis-

tribution over the spoken phoneme for each time frame in a spectrogram audio representation. PPGs can be obtained via an ASR model that is trained with phoneme labels (e.g. on the TIMIT dataset [39]) or can be obtained through forced alignment [40]. In this paper, we use an ASR model that is trained on TIMIT.

Given a saliency map M, obtained via a saliency method (e.g., IG) for a classifier decision on a speech utterance, we discretize M by pooling its weights contained within phoneme boundaries. These phoneme boundaries are derived from the PPG representation of the utterance. We describe the exact procedure in Algorithm 1. (Note that we use the python array notation. That is, $B[:, s_i : e_i]$ takes the entire first dimension of a matrix B, and takes the slice that runs between s_i and e_i for the second dimension.) We also pictorially explain the algorithm in Figure 3. We would like to note that there are few hyperparameters of this algorithm which affect the faithfulness of generated binary maps - specifically the choice of the *preprocess* and the *pool* functions.

Once we obtained a discretized Saliency map, we convert it into a binary mask by keeping only the k most important phonemes as judged by the map.

Algorithm 1 Phoneme Discretization for Saliency Maps

Input: Saliency Map M of shape $[F, T]$, PPG X^p of shape $[N, T]$, number of phonemes to keep k.
Output: Phoneme Discretized Saliency Map $\widehat{M}$.
Hyperparameters: functions $preprocess$, $pool$.

- $\widetilde{M} = preprocess(M)$
- Get time frames to phoneme alignment:
  $\widetilde{X}^p_t = \arg\max_i X^p_{i,t}, \text{ for } t \in \{1, \ldots, T\}$.
- Get phonemes and their start and end boundaries:
  $(p_1, s_1, e_1), \ldots (p_n, s_n, e_n)$ from $(\widetilde{X}^p_t)$.
- Let energy of phoneme p_i be $E(p_i) := pool(\widetilde{M}[:, s_i : e_i])$.
- Let $J \subset \{1, 2, \ldots n\}$, $|J| = k$ be the indices of k highest energy phonemes.
- Initialize $\widehat{M} = zeros(F, T)$.
- $\widehat{M}[:, s_j : e_j] = 1, \forall j \in J$
- Return $\widehat{M}$

Note that N is the vocabulary size of the PPG representation, and n is the number of phonemes in this utterance. We first preprocess the saliency map - which may involve thresholding the map to remove small values and/or passing the map through an abs (absolute value) function. We noticed that these exact preprocessing decisions are crucial and serve as hyperparameters for our algorithm. For our experiments with fake voice generated using Tacotron2, we found that thresholding the maps and not taking their abs resulted in more faithful discrete maps. However, for FastSpeech2, we obtained better maps by thresholding and taking an abs of the original saliency maps.

Similarly, the choice of the pooling function affects the faithfulness of the obtained discrete maps. We found mean pooling to be more effective for Tacotron2, but sum pooling worked better for FastSpeech2. We will release the code of our algorithm in the camera ready version.

4. Evaluating the Faithfulness of PDSMs

In order to systematically measure the faithfulness obtained on the real/fake classification problem, we calculate the faithfulness metric defined in Equation (1) for standard saliency methods and our proposed phoneme discretized saliency maps (PDSM). In order to also assess the effectiveness of picking phonemes for the binary mask through PDSM we compare our methods with the faithfulness obtained by randomly assigning phonemes to the mask as a baseline. We compared faithfulness for various saliency methods (IG [23], GradSHAP [24], Grad-Input [41], GuidedBP [42], Gradient Saliency [41], DeepLift [43]). In Figure 4, we compare the behavior of the faithfulness metric for different number of phonemes retained (shown on the x-axis of the plots) in the discretized map. We observe in the left-most and the middle-left panel that PDSM obtains better faithfulness metrics for all values of k (number of phonemes in the mask) compared to the baseline that randomly picks k phonemes (except the vanilla Gradient saliency method). In the middle-right and right-most panels of Figure 4, we also do this same comparison but normalized by the length of the utterance. We again see that PDSM results in much more faithful explanations as compared to faithfulness obtained from randomly picking phonemes or faithfulness resulting from original salilency maps.

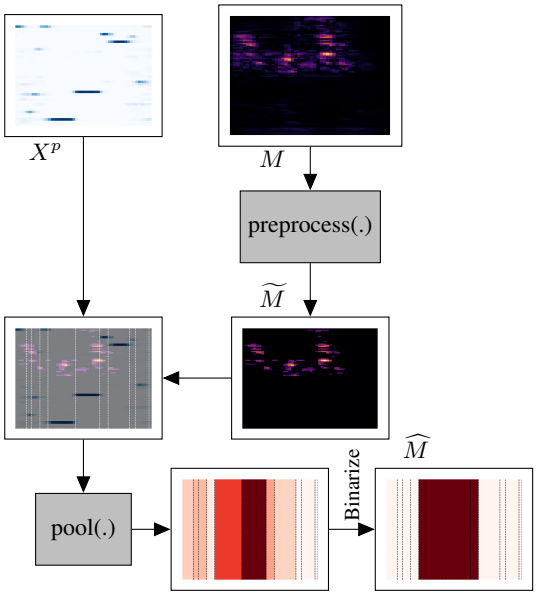

Figure 3: *Steps of the PDSM algorithm.*

In Table 1, we compare the faithfulness of PDSM compared to the straightforward application of different saliency methods vs. their application after going through the PDSM algorithm. We observe that the application of the PDSM algorithm significantly increases the faithfulness of the saliency map compared to direct computation of faithfulness on the original saliency maps.

5. Evaluating the Understandability of PDSMs

First, we show that PDSMs provide understandable explanations for AI generated voice detection. In figure 5, we provide a population level analysis to show highest ranked phonemes in terms of saliency weights for Tacotron2 and FastSpeech2. To obtain these plots, we compute the total pooled energy assigned to a phoneme by the saliency maps obtained using Integrated Gradients accumulated over the entire test set. We then normalize this total energy by the duration(as measured in number of

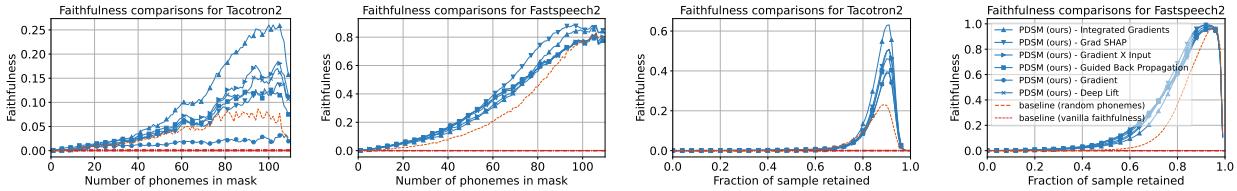

Figure 4: ***(left-most, left-center)*** *Analysis of faithfulness with respect to different numbers of phonemes retained in the binary mask for Tacotron2 and FastSpeech2 respectively.* ***(right-center, right)*** *Same analysis of faithfulness normalized over the length of audio sample, thus representing fraction of audio retained for the binary mask.*

Table 1: *Comparison of average faithfulness of PDSM (in bold) vs. plain saliency methods (in parantheses), for Tacotron 2 (denoted as TT2), and Fastspeech 2 (denoted as FS2). On the right most column we also include the random phoneme selection baseline (denoted with RP) discussed in Section 4.*

	IG	GradShap	GradInput	GuidedBP	Gradient	DeepLift	RP
TT2	**0.10** (10^{-5})	**0.052** ($2 \cdot 10^{-3}$)	**0.066** ($-1.9 \cdot 10^{-5}$)	**0.056** ($-2.4 \cdot 10^{-5}$)	**0.015** ($-2.4 \cdot 10^{-5}$)	**0.059** ($-2.4 \cdot 10^{-5}$)	0.037
FS2	**0.35** (10^{-3})	**0.396** ($5 \cdot 10^{-4}$)	**0.337** ($6 \cdot 10^{-4}$)	**0.352** ($2 \cdot 10^{-4}$)	**0.352** ($2 \cdot 10^{-4}$)	**0.346** ($2.8 \cdot 10^{-5}$)	0.27

time frames occupied in mel representation) of this phoneme computed over the entire test set. We notice two trends which ascertain the fact that PDSM is giving phoneme importances as expected. i) We observe that in both cases silent phonemes are assigned large relative importance, which are typically problematic in TTS systems. ii) We also know that Tacotron2 sounds more realistic overall compared to FastSpeech2, and therefore we see that overall smaller number of phonemes are assigned high importance.

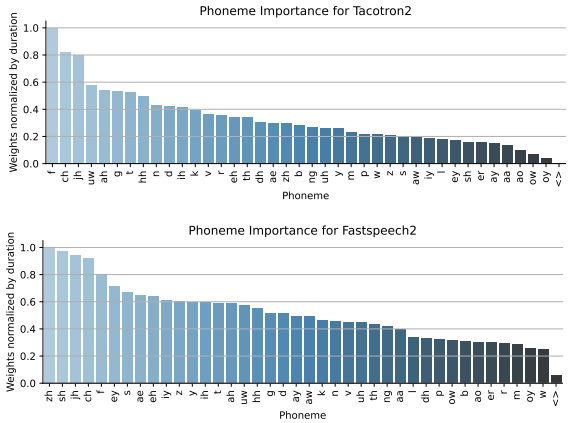

Figure 5: *Global Phoneme Importances Obtained by Tacotron2 and FastSpeech2.*

Next, we show an analysis on a particular example for FastSpeech2. In Figure 6, we show the phonemes with largest assigned saliency values. We observe that when contrasted with the same phoneme boundaries in the real speech, PDSM is able to accurately detect phonemes that appear unnatural. We also observe that the algorithm points out to phonemes which have been given large global importance as shown in the bottom panel of Figure 5.

6. Conclusions and Discussions

In this paper, we proposed PDSM, a method that discretizes saliency maps through phoneme boundaries in order to produce faithful and understandable explanations for complex classification tasks where the users do not have preconceived notions to

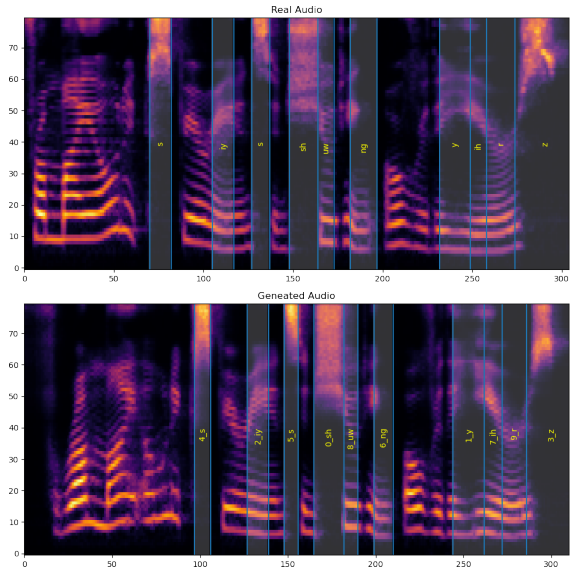

Figure 6: *Sample wise analysis of understandability of PDSM interpretations.* ***(top)*** *Mel-Spectrogram for a real audio* ***(bottom)*** *Mel-Spectrogram for a generated audio that corresponds to the same utterance. In both figures we overlay the phoneme boundaries which are assigned high importance, and we also show the phoneme identity and the corresponding importance order. We see that the algorithm is able accurately focus on phonemes which appear unnatural on the spectrogram.*

be able to understand the saliency maps. We showed that PDSM is able to produce more faithful results compared to standard posthoc interpretation methods. We also showed that the produced interpretations can be broken down in terms of phonemes which can help users understand the neural network decision much more easily.

We also would like note that even though in this paper we have focused on using PDSM for explainable detection of AI generated voice, it is a general algorithm that can potentially be used in other complex speech/audio classification tasks. Therefore as future work, we are planning to expand the scope of PDSM to cover complex speech classification tasks such as emotion recognition, and speech related medical applications.

7. References

[1] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," 2023.

[2] T. B. Brown *et al.*, "Language models are few-shot learners," 2020.

[3] "Chatgpt: Optimizing language models for dialogue," 2022.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[6] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "Adaspeech: Adaptive text to speech for custom voice," 2021.

[7] D. Paul, M. P. Shifas, Y. Pantazis, and Y. Stylianou, "Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion," 2020.

[8] E. Casanova, J. Weber, C. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," 2023.

[9] https://www.cbc.ca/news/canada/montreal/chatgpt-chatbot-quebec-teachers-1.6724519, Accessed 10.03.2024.

[10] https://www.bbc.com/news/technology-49446729, accessed 10.03.2024.

[11] https://listverse.com/2021/05/06/top-10-outrageous-deepfake-scandals/, accessed 10.03.2024.

[12] https://www.independent.co.uk/tech/ai-voice-clone-scam-kidnapping-b2319083.html, accessed 10.03.2024.

[13] Https://www.abc.net.au/news/2023-04-12/artificial-intelligence-ai-scams-voice-cloning-phishing-chatgpt/102064086, accessed 10.03.2024.

[14] Https://time.com/6243988/top-global-risks-of-2023/, accessed 10.03.2024.

[15] E. A. AlBadawy, S. Lyu, and H. Farid, "Detecting ai-synthesized speech using bispectral analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[16] R. Wang, F. Juefei-Xu, Y. Huang, Q. Guo, X. Xie, L. Ma, and Y. Liu, "Deepsonar: Towards effective and robust detection of ai-synthesized fake voices," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[17] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," 2019.

[18] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," 2021.

[19] H. Tak, J. weon Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," 2021.

[20] Z. Lv, S. Zhang, K. Tang, and P. Hu, "Fake audio detection based on unsupervised pretraining models," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 9231–9235.

[21] J. Xue, C. Fan, J. Yi, C. Wang, Z. Wen, D. Zhang, and Z. Lv, "Learning from yourself: A self-distillation method for fake speech detection," 2023.

[22] A. Pianese, D. Cozzolino, G. Poggi, and L. Verdoliva, "Deepfake audio detection by speaker verification," 2022.

[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017.

[24] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[25] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, 2015.

[26] S.-Y. Lim, D.-K. Chae, and S.-C. Lee, "Detecting deepfake voice using explainable deep learning techniques," *Applied Sciences*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248193287

[27] C. Churchwell, M. Morrison, and B. Pardo, "High-fidelity neural phonetic posteriorgrams," 2024.

[28] J. Parekh, S. Parekh, P. Mozharovskyi, F. d'Alché-Buc, and G. Richard, "Listen to interpret: Post-hoc interpretability for audio networks with nmf," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[29] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis," in *International Society for Music Information Retrieval Conference*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:795766

[30] S. Becker, J. Vielhaben, M. Ackermann, K.-R. Müller, S. Lapuschkin, and W. Samek, "Audiomnist: Exploring explainable artificial intelligence for audio analysis on a simple benchmark," 2023.

[31] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2014.

[32] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, no. 2, p. 336–359, Oct. 2019.

[34] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[35] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2018.

[36] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," 2022.

[37] M. Ravanelli *et al.*, "Speechbrain: A general-purpose speech toolkit," 2021.

[38] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," 2020.

[39] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.

[40] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[41] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2015.

[43] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," 2019.