Creating More Meaningful Explanations for Audio Deepfake Detection

Bachelorarbeit

im Studiengang
Wirtschaftsinformatikim 6. Semester

vorgelegt von

Jaocb LaRock
Matr.-Nr.: 1667321
am 31.03.2025

Universität Siegen

Fakultät III: Wirtschaftswissenschaften, Wirtschaftsinformatik und Wirtschaftsrecht

Erstprüfer: Gunnar Stevens Gunnar.Stevens@uni-siegen.de

1 Abstract

Abstract filler text blablabla!

2 Introduction

As the use of generative methods for the creation of synthetic voices becomes more widespread, so grows the need for reliable and usable detection methods for the protection of the security of people and businesses alike. In particular, the rise of deepfake technology has led to concerns about its potential misuse in areas such as politics, entertainment, and national security by, for instance, potentially allowing malicious actors to create fake audio recordings that appear to be genuine statements made by public figures or to create genuine-seeming recordings of events that never happened, which could have significant consequences if used to spread misinformation, from defaming individuals to creating political tension [17, 1]. The field of Explainable AI (XAI) shows promise for producing useful and interpretable results from such models. Explainable AI refers to the ability of artificial intelligence systems, such as machine learning models and neural networks, to provide understandable and interpretable explanations for their decisions or predictions [9]. This means that XAI systems can articulate why they made a particular decision, what factors influenced it, and how confident they are in their conclusion [9]. For the detection of audio deepfakes, this could mean that the model can justify its classification with feature-based evidence, allowing for verification of the result by the end user. Much of the existing research, further discussed in the next section, follows the path of producing a model with the best possible benchmarks against datasets of samples, without taking into account if the results can be usefully interpreted, while existing explorations into making audio deepfake detection explainable have produced results that often require a lot of background knowledge to meaningfully interpret. Aside from this, explainability in the area of audio deepfake detection remains an open challenge [5], one even promoted by

regularatory bodies such as the European Union with the "right to explain" [7]. With the context of the research discussed in the next section, I would like to use new combinations of features with a novel model architecture to generate explanations for fake or real audio classification, with the goal of singling out features that are useful not only for an accurate classification but also for an understandable explanation.

3 Related Work

Several researchers have already followed similar or relevant directions to that of this work. This section will summarize some efforts in previous research categorized by the use of perceptible features or attempts to create explainable models.

3.1 Perceptible Features

The work of Barrington et al. [2] explores the idea of classification using perceptible features, highlighting their potential for improving explainability in the space of deepfake audio detection. They do, however also demonstrate a drop in performance in comparison with imperceptible hand-crafted and deep-learning features in their experiments, which were performed on a combination of synthetic and real audio datasets.

Chaiwongyen et al. [3, 4] also approach using perceptible features for classification in their works. They trained and tested using the dataset of the ADD2022 Challenge [20] and a simple model with a single hidden layer, resulting in lackluster performance in 2022, with better results using an expanded and improved feature set in 2023.

Li et al. [11] also approach perceptible features in their work, combining them with imperceptible (referred to as physical in the work) features and using various neural networks in their experiments. Overall, the combination of perceptible and imperceptible features was able to produce the best performance results

of the experiments, which were performed as well on the dataset of the ASV2022 Challenge [20].

3.2 Explainable Models

One previous example of an implementation of an explainable model for use in audio deepfake detection came from Ge et al. [6], who explain feature influence on models using the SHAP (SHapley Additive exPlanations) method. They apply this method using log-scaled power spectrograms as a feature, training and testing using the datasets from the ASV2019 challenge [18], they are able to determine and graphically represent areas of importance on the spectrogram, as well as globally summarize the SHAP-values.

Another example is presented by Haq et al. [8], who use the changes in emotional state as an input feature and represent "unlikely" changes on a graph so that it can be understood by an eventual end user. In this case, they achieved their results by combining the output of fake video and fake audio classifiers in order to produce a final classification for a video sample with audio. Testing against the presidential deepfake dataset [16], they achieve impressive results in comparison to the standing benchmark on the dataset at the time.

4 Experiments

4.1 Dataset

The dataset used for these experiments is the "In-the-Wild" dataset, one which focuses on the generalization of audio deepfake detection models by collecting real-world data, in comparison to other examples that use more controlled laboratory conditions [13].

4.2 Black-Box Model

The model used for this experiment was made, trained and evaluated using the tools available in the TensorFlow and Keras libraries [12]. This section will go into further detail about the experimental setup for the black-box model as well as the way in which explanations were created and summarized.

4.2.1 Features

Several features were used for the final version of the black-box model. Some are features that can be called generic, widely used in research involving the detection of synthetic while others are more specific and are cited accordingly. The features that are not summarized for a whole sample are extracted using a sliding window method, preventing a loss of fidelity that can be caused by compression of the feature to a standard size. The features used are as follows:

- Harmonic-to-noise ratios: Inspired by previous research [3, 4, 11] but done in a sliding window fashion instead of on the whole file, the HNRs are the ratios of the strengths of harmonic frequencies to the strengths of "noise", the total strength outside the harmonic frequencies. Unlike some previously seen examples [3, 4], I calculate this ratio for each fundamental frequency length, instead of using one value for the whole sample. Given that γ_i is the harmonic energy in a given fundamental frequency cycle and ι_i is the residual energy in a given fundamental frequency cycle, the HNR at that cycle is calculated as follows:

$$20log\frac{\gamma_i}{\iota_i}$$

- Mel-spectrograms: A widely-used feature in synthetic audio detection, the mel-spectrogram is a spectrographic representation of an audio sample transformed in a manner to better represent human perception of frequencies [14].

- Mel-frequency cepstral coefficients: The derivative cepstral coefficients of

the mel-spectrogram, also widely-seen in research on synthetic audio detection.

- Fundamental frequency lengths: The lengths of every fundamental frequency cycle in the sample. This has also been previously used in certain research on synthetic audio detection [19]. The output is one-dimensional with time as the axis.

- Onset strength: As used in previous research [11], the strengths of each onset in the audio sample, where an onset is point where there is a sudden rise in energy across the audio spectrum. This results in a one-dimensional output with time as the axis.

- Intensity: Also inspired by previoous research [11], intensity is the total power at each point in the audio sample, given in db. This is calculated by creating a fourrier transformation of the sample and then summing across the frequency-axis for every point on the time-axis. Resulting in a one-dimensional output.

- Pitch-fluctuations: Calculated for every sample in the audio as the difference between pitch at the given point and the pitch at the previous point. Pitch is estimated based on the maximum power harmonics. This feature is similar to the use of summarized pitch fluctuations in previous research [10]. This feature is also one-dimensional with time as the axis. Given that H_i is the set of harmonic frequencies at fundamental frequency cycle i with $h_i \in H_i$ as a frequency of the set and $s(h_i)$ is the power of a given harmonic frequency at cycle i, the pitch can be estimated as follows:

  $$p_i = s(max(H_i))$$

  Then, given an offset x, the pitch fluctuations can be calculated as follows:

  $$p_i - p_{i-x}$$

- Jitter features: As defined in previous attempts to identify synthetic audio [4] with perceptible features, jitter-based features measures the absolute

variations in fundamental frequency cycles in comparison to the nearest x neighbors using the following method:

$$\frac{\frac{1}{N-1}\sum_{i=1}^{N-1}|T_i(\frac{1}{x}\sum_{n=i-m}^{i+m}T_n)|}{\frac{1}{N}\sum_{i=1}^{N}T_i}$$

where T_i represents the extracted fundamental frequency lengths and N is the number of fundamental frequency periods.

- Shimmer features: Similarly defined to the jitter features, shimmer features instead use the amplitudes at each fundamental frequency period. Their purpose is to capture irregular vocal fold vibrations which may be an indication but not a guarantee of a synthetic voice [4].

$$\frac{\frac{1}{N-1}\sum_{i=1}^{N-1}|A_i(\frac{1}{x}\sum_{n=i-m}^{i+m}A_n)|}{\frac{1}{N}\sum_{i=1}^{N}A_i}$$

4.2.2 Model Architecture

The general architecture is designed to allow the combination of diverse features into one final prediction, regardless of whether the individual features have the same shape. This is achieved by using a separate model for each feature, with its own input and output layers (further referred to as sub-models), which are then concatenated together, processed though a final model that pools into a final output value (which I will further refer to as the terminus model). The modular structure allows not only better performance and less memory use, due to the fact that fewer transformations are required in input preparation, but also allows for more flexibility in the construction of the model, making it easy to add and remove features using pre-defined functions depending on feature type. This kind of architecture also provides an advantage to the generation of explanations, as only the terminus part of the entire must be explained to interpret the importance of each input feature. Further explanation is in the following section. Figure 1 represents the layout of the model layers.

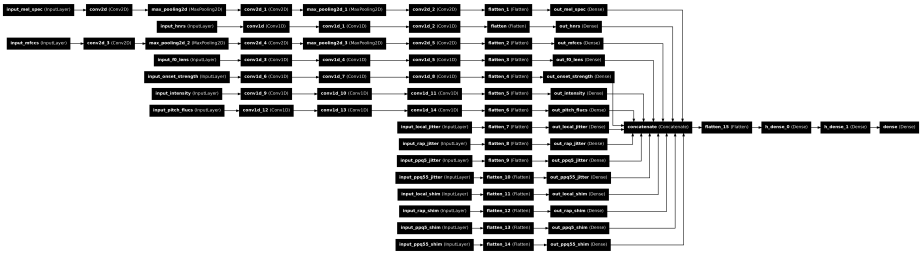

Figure 1: The architecture of the model used for the experiment.

4.2.3 Training and Evaluation

The black-box model described in the previous section was trained on the first //TODO// samples in the dataset. Evaluation was then performed on the remaining samples in the dataset. Training batches consisted of 100 samples each with every window position for each sample and an upper limit of 1,000,000 lines per batch. For each batch, two epochs were performed. Because a single sample can have multiple input lines, evaluation cannot be performed on a per-line basis, but must instead be summarized. Two methods are possible: median and mean, with the difference in results being negligible in my experiments. The default threshold is 0.5, meaning that a result under 0.5 is a false result and over 0.5 is a true result. This threshold can however be adjusted, useful for computing the EER.

4.3 Explanations

The explanations using the previously described surrogate model are generated with the Local Interpretable Model-agnostic Explanation (LIME) method [15]. This section will describe in detail how individual explanations are generated at a local level for individual samples, as well as how the usefulness of the explanations can be evaluated in general.

4.3.1 Generation of the Explanations

Because of the multi-shaped nature and independence of the input layers of the model, the explanations cannot be generated directly from the input itself. However, because the sub-models are separate from one another, having no influence on each other before being concatenated at the terminus, only the terminus part of the model is relevant for assessing the importance of the features in a single evaluation. This does, however, present two challenges:

- There are multiple input rows per sample.
- The features cannot be used directly for assessment at the terminus input layer.

The first problem can be solved by taking the mean of the explanation weights of each feature produced by each row of the sample features. This results in an average contribution of each feature to the classification of the model. Because the end-classification of the surrogate is also a mean, the means of the LIME-results are an accurate representation of the aggregate influence.

I approached the second problem by generating intermediate data for the LIME-evaluation. Intermediate data was generated by first decomposing the surrogate model into its component models, the sub-models and the terminus, and for each sub-model, running a prediction on the given input features. The results of the predictions of the sub-models were stored and used for further analysis. This was done not only for the sample explanation data but also for the training data, to use as an input for the LIME explainer.

The results of the LIME explainer are then summarized together on a per-feature basis and normalized to produce a decimal number between -1 and 1 for every feature. In this case, a negative value implies that the given feature pushed the result of the model in the negative direction (i.e. not fake), while a positive value indicates the opposite.

4.3.2 Evaluation of the Explanations

In order to assess if the explanations have the potential to be useful to an end user, I pursued the goal of summarizing many explanations generated using the previously described method together in a meaningful way, in order to make a conclusion about the general usefulness of the individual features and the influence and potential use of the perceptible features. However, due to a lack of existing standardized metrics for the global evaluation of local metrics, I propose two metrics, with which I will make conclusions about the usefulness of this method.

In order to contextualize the metrics, I will make the following definitions.

- Let S be the set of samples with $s \in S$ as an element of the set.
- Let F be the set of features with $f \in F$ as an element of the set.
- Let w_{fs} be the weight value of feature f of sample s as produced by the explainer.
- Let l_s be the correct label of the sample s.

The first metric I will define is the mean of the absolute values of the weights of the explanations, summarized along the sample axis, delivering one value per feature. I will further refer to this metric as importance, and it can be mathematically defined as follows:

$$\frac{\sum_{s \in S} |w_{fs}|}{|S|}$$

The second metric I will define is the average aggregate correctness on a per-feature basis, which I will further refer to as trust. The trust per feature can be defined as follows:

$$\frac{\sum_{s \in S} w_{fs}(2l_s - 1)}{|S|}$$

5 Results

5.1 Black-Box Model Results

5.2 Explanation Results

6 Discussion

7 Conclusion

References

[1] Marwan Albahar and Jameel Almalki. "DEEPFAKES: THREATS AND COUNTERMEASURES SYSTEMATIC REVIEW". In: . *Vol.* 22 (2005).

[2] Sarah Barrington et al. *Single and Multi-Speaker Cloned Voice Detection: From Perceptual to Learned Features*. Sept. 27, 2023. DOI: 10.48550/arXiv.2307.07683. arXiv: 2307.07683[cs]. URL: http://arxiv.org/abs/2307.07683 (visited on 12/18/2024).

[3] Anuwat Chaiwongyen et al. "Contribution of Timbre and Shimmer Features to Deepfake Speech Detection". In: *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2022 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). Chiang Mai, Thailand: IEEE, Nov. 7, 2022, pp. 97–103. ISBN: 978-616-590-477-3. DOI: 10.23919/APSIPAASC55919.2022.9980281. URL: https://ieeexplore.ieee.org/document/9980281/ (visited on 05/28/2024).

[4] Anuwat Chaiwongyen et al. "Deepfake-speech Detection with Pathological Features and Multilayer Perceptron Neural Network". In: *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). Taipei, Taiwan: IEEE, Oct. 31, 2023, pp. 2182–2188. ISBN: 9798350300673. DOI: 10.1109/APSIPAASC58517.2023.10317331. URL: https://ieeexplore.ieee.org/document/10317331/ (visited on 05/28/2024).

[5] Luca Cuccovillo et al. "Open Challenges in Synthetic Speech Detection". In: *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. Dec. 12, 2022, pp. 1–6. DOI: 10.1109/WIFS55849.2022.9975433. arXiv: 2209.07180[eess]. URL: http://arxiv.org/abs/2209.07180 (visited on 12/18/2024).

[6] Wanying Ge et al. *Explaining deep learning models for spoofing and deepfake detection with SHapley Additive exPlanations*. Apr. 26, 2024. DOI: 10.48550/arXiv.2110.03309. arXiv: 2110.03309[eess]. URL: http://arxiv.org/abs/2110.03309 (visited on 12/18/2024).

[7] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI Magazine* 38.3 (Sept. 2017), pp. 50–57. ISSN: 0738-4602, 2371-9621. DOI: 10.1609/aimag.v38i3.2741. arXiv: 1606.08813[stat]. URL: http://arxiv.org/abs/1606.08813 (visited on 01/18/2025).

[8] Ijaz Ul Haq, Khalid Mahmood Malik, and Khan Muhammad. "Multimodal Neurosymbolic Approach for Explainable Deepfake Detection". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* (Sept. 20, 2023), p. 3624748. ISSN: 1551-6857, 1551-6865. DOI: 10.1145/3624748. URL: https://dl.acm.org/doi/10.1145/3624748 (visited on 05/11/2024).

[9] Michael Hind. "Explaining explainable AI". In: *XRDS: Crossroads, The ACM Magazine for Students* 25.3 (Apr. 10, 2019), pp. 16–19. ISSN: 1528-4972, 1528-4980. DOI: 10.1145/3313096. URL: https://dl.acm.org/doi/10.1145/3313096 (visited on 06/05/2024).

[10] Zahra Khanjani et al. "Learning to Listen and Listening to Learn: Spoofed Audio Detection Through Linguistic Data Augmentation". In: *2023 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2023 IEEE International Conference on Intelligence and Security Informatics (ISI). Charlotte, NC, USA: IEEE, Oct. 2, 2023, pp. 01–06. ISBN: 9798350337730. DOI: 10.1109/ISI58743.2023.10297267. URL: https://ieeexplore.ieee.org/document/10297267/ (visited on 05/28/2024).

[11] Menglu Li, Yasaman Ahmadiadli, and Xiao-Ping Zhang. "A Comparative Study on Physical and Perceptual Features for Deepfake Audio Detection". In: *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. MM '22: The 30th ACM International Con-

ference on Multimedia. Lisboa Portugal: ACM, Oct. 14, 2022, pp. 35–41. ISBN: 978-1-4503-9496-3. DOI: 10.1145/3552466.3556523. URL: https://dl.acm.org/doi/10.1145/3552466.3556523 (visited on 05/11/2024).

[12] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[13] Nicolas M. Müller et al. *Does Audio Deepfake Detection Generalize?* Apr. 21, 2022. arXiv: 2203.16263[cs,eess]. URL: http://arxiv.org/abs/2203.16263 (visited on 07/04/2024).

[14] Abu Qais et al. "Deepfake Audio Detection with Neural Networks Using Audio Features". In: *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*. 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP). Hyderabad, India: IEEE, July 21, 2022, pp. 1–6. ISBN: 978-1-66547-258-6. DOI: 10.1109/ICICCSP53532.2022.9862519. URL: https://ieeexplore.ieee.org/document/9862519/ (visited on 05/28/2024).

[15] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. Aug. 9, 2016. DOI: 10.48550/arXiv.1602.04938. arXiv: 1602.04938[cs]. URL: http://arxiv.org/abs/1602.04938 (visited on 01/18/2025).

[16] Aruna Sankaranarayanan et al. "The Presidential Deepfakes Dataset". In: ().

[17] Namosha Veerasamy and Heloise Pieterse. "Rising Above Misinformation and Deepfakes". In: *International Conference on Cyber Warfare and Security* 17.1 (Mar. 2, 2022), pp. 340–348. ISSN: 2048-9889, 2048-9870. DOI: 10.34190/iccws.17.1.25. URL: https://papers.academic-conferences.org/index.php/iccws/article/view/25 (visited on 09/23/2024).

[18] Xin Wang et al. *ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech*. July 14, 2020. arXiv: 1911.01601[cs,eess]. URL: http://arxiv.org/abs/1911.01601 (visited on 07/03/2024).

[19] Jun Xue et al. "Audio Deepfake Detection Based on a Combination of F0 Information and Real Plus Imaginary Spectrogram Features". In: *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. MM '22: The 30th ACM International Conference on Multimedia. Lisboa Portugal: ACM, Oct. 14, 2022, pp. 19–26. ISBN: 978-1-4503-9496-3. DOI: 10.1145/3552466.3556526. URL: https://dl.acm.org/doi/10.1145/3552466.3556526 (visited on 05/11/2024).

[20] Jiangyan Yi et al. *ADD 2022: the First Audio Deep Synthesis Detection Challenge*. July 2, 2024. DOI: 10.48550/arXiv.2202.08433. arXiv: 2202.08433[cs]. URL: http://arxiv.org/abs/2202.08433 (visited on 01/22/2025).