Explaining Audio Deepfake Detection with Perceptible Features and LIME

Bachelorarbeit

im Studiengang
Wirtschaftsinformatik im 6. Semester

vorgelegt von

Jaocb LaRock
Matr.-Nr.: 1667321
am 17.03.2025

Universität Siegen

Fakultät III: Wirtschaftswissenschaften, Wirtschaftsinformatik und Wirtschaftsrecht

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

1 Abstract

With the unprecedented advancement of Generative Artificial Intelligence (GenAI), the threat of voice scams using synthetic voices has become a serious concern across various sectors. Recent efforts have focused on identifying fake voices through handcrafted features, deep learning models, and hybrid approaches. However, most existing methods lack explainability, rendering their predictions non-transparent to users. This paper proposes a novel, interpretable, and transparent method for fake voice identification by introducing a hybrid deep learning model that leverages multiple extracted features. The hybrid model consists of two main components: the first component addresses heterogeneous feature spaces by employing deep convolutional sub-models tailored to individual features, while the second component, the terminus model, utilizes the concatenated representations from the final layers of each sub-model as input. The terminus model follows a typical multi-layer perceptron architecture, enabling effective integration and classification of the diverse feature representations. To enhance interpretability, we decompose the model's decisions using Local Interpretable Model-agnostic Explanations (LIME), exploiting the identical feature representation before the concatenation layers to address challenges related to multi-dimensional feature representations. To evaluate the importance and trustworthiness of features in the generated explanations, we propose two metrics: importance and trust. Extensive experiments are conducted on the In-the-Wild dataset, which is designed to test the generalization capability of synthetic audio detection methods. The experimental results demonstrate that our approach achieves performance comparable to benchmark methods. Furthermore, the results based on our proposed metrics conclude that certain perceptible features demonstrate promise for generating explanations that are meaningful to general users. For reproducibility, the source code for these experiments is available in the following repository: `https://github.com/jacoblarock/fake_voices_xai`

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 [26, 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 [14]. 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 [14]. 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 [7], one even promoted by regulatory bodies such as the European Union with the "right to explain" [11].

In this work, we make a distinction of two main categories of input features for

a synthetic voice detection model: perceptible and imperceptible features. Perceptible features are features that can be perceived by the human ear, often vocal qualities such as jitter, shimmer or pitch fluctuation that have a wide range of uses even outside of audio classification such as diagnosis of disease [6], while imperceptible features are typically out of the range of human hearing, and may not directly reflect a vocal quality, otherwise referred to as "speaker-independent" features [17]. We use a new combination of features for our method, with an emphasis being placed on perceptible features, as we believe these to be the most likely to be understood by the layperson.

We introduce a hybrid deep-learning model, with the purpose of combining these multidimensional features into a single output. The hybrid model consists firstly of a component sub-model for each of the input features, in order to address the problem of varying dimensionality of the extracted features without the drawbacks of costly transformations in the pre-processing phase. The final part of our hybrid model is what we will refer to as the terminus model, a model that uses the concatenated outputs of the individual sub-models as inputs and distills down to a singular output with a classic multi-layer perceptron architecture, allowing for an effective combination of the features into a final singular classification result.

In order to introduce explainability into our method, we generate explanations using a method that first generated the intermediate output data from the sub-models and then applies the Local Interpretable Model-agnostic Explanations (LIME) [23] on the terminus model, which allows us to, for an individual classification, assess the impact of each input feature on the final result through the local explanation, as, due the independence of the sub-models from one another, the importance of their outputs directly correlates with the importance of their inputs when it comes to the end-classification made by the terminus model.

We then stood before the problem of assessing our explanations, more specifically, assessing the impact of the input features on the average local explanation, in order to make a conclusion about the usefulness or promise of our chosen fea-

tures for providing understandable explanations of our model's classifications. In order to tackle the problem with our local explanations, and due to a lack of metrics to assess the aggregated average value of local explanations, we also introduce two metrics for such an aggregate evaluation of large number of generated LIME explanations, importance and trust, on a per-feature basis, which we use to make a conclusion about the value of the selected feature set for the purpose of sensible explanations.

The contributions of this work are in summary: Our hybrid model-architecture and modified LIME-based method for generating explanations of classifications of audio voice samples as synthetic or bona-fide and the two metrics that we introduce in order to assess the impact of individual features over numerous local explanations.

This paper is divided into the following sections:

- §3 Related Work, where we discuss the most relevant related research in the field, in order to create a picture of the current state-of-the-art
- §4 Method, where we describe the general model implementation, as well as the implementation of our LIME-based method for generating explanations in more detail.
- §5 Experimental Results, where we discuss how we performed our experiments, how our hybrid model performs, as well as use our proposed metrics in order to assess the feature impact in the average explanation.
- §6 Discussion, where we discuss the implication of the results of our experiments, including the potential of the explanations as well as the shortcomings and limitations of our method.

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 divided

into categories of perceptible and imperceptible features, as well as previous efforts at explainability in this field.

3.1 Synthetic Voice Detection With Imperceptible Features

The most common approach in this field makes use of either learned or hand-crafted imperceptible features. Examples include spectrographic features such as the mel-spectrogram and its hand-crafted derivative the mel-frequency cepstral coefficients (MFCCs), both of which are present in our method due to their widespread successful use.

The work of Anagha et al. [3] is an example that makes use of mel-spectrograms in combination with an architecture based on convolutional neural networks, achieving well-performing results on the ASVSpoof2019 dataset [27].

A further work from Yan et al. [30] makes use of MFCCs, in combination with other hand-crafted imperceptible features such as linear frequency cepstral coefficients (LFCCs) to not only classify audio samples as synthetic or authentic, but also to correctly classify the vocoder used to produce the sample with nearly perfect accuracy on a handcrafted dataset.

Other researchers, such as in the work of Qais et al. [21] make use of Fourier transforms, such as the short term Fourier transform (STFT). The aforementioned example uses these features combined with others such as the aforementioned mel-spectrogram based features. They also pair these features with a convolutional architecture and achieve notably accurate results on the ASVSpoof2017 dataset [8].

Yang et al. [31] demonstrate a comparison of multiple features, as well as a feature selection method, with the goal of maximizing model efficiency. They run experiments with several hand-crafted features as well as learned features. They train and test using three datasets, ASVSpoof2019 [27], ASV2021 [18] and In-the-Wild [20], presenting their results for each feature, as well as the results

of their selection and classification fusion methods.

3.2 Synthetic Voice Detection With Perceptible Features

There is also a smaller but still notable body of work encompassing the use of perceptible features for the detection of synthetic voices, making use of a variety of features, some of which were selected for use in our experiments.

The work of Barrington et al. [4] explores the idea of classification using perceptible features, highlighting their potential for improving explainability in the space of deepfake audio detection, although they only implement a classifier and not an explainer. They do, however, also demonstrate a drop in performance when using perceptible features 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. [5, 6] also approach using perceptible features for classification in their works. They developed a perceptron architecture with a single hidden layer and trained and tested using the dataset of the ADD2022 Challenge [32], resulting in lackluster performance in 2022, with better results using an expanded and improved feature set in 2023.

Li et al. [16] 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. With their combination of perceptible and imperceptible features, they were able to produce the best performance results of their experiments in comparison to experiments run with only perceptible or imperceptible features, which were performed as well on the dataset of the ASV2022 Challenge [32].

3.3 Explainable Models

There have also been some efforts in the previous research to pair synthetic voice detection models with explainable methods.

One previous example of an implementation of an explainable model for use in audio deepfake detection came from Ge et al. [10], 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 [27], 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. [13], 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 [24], they achieve impressive results in comparison to the standing benchmark on the dataset at the time.

4 Method

4.1 Black-Box Model

We created, trained and evaluated the model used for our experiments using the tools available in the TensorFlow and Keras libraries [19]. 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.

4.1.1 Features

In order to increase the likelihood of the explanations being useful and understandable to the end user, we placed a focus on using multiple perceptible features as inputs for the classifier. We added used two imperceptible features with the hypothesis, based on previous research [4, 5, 6, 16], that they would positively of increasing the model performance.

Several features were used for the final version of the black-box model. Some are features 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 we considered perceptible are as follows:

- Harmonic-to-noise ratios: A perceptible feature inspired by previous research [5, 6, 16] 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 [5, 6], 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 hnr_i is calculated as follows:

$$hnr_i = 20log\frac{\gamma_i}{\iota_i}.$$

- Fundamental frequency lengths: Considered a perceptible feature, the fundamental frequency lengths (f0 lengths) are the lengths of every fundamental frequency cycle in the sample. This has also been previously used in certain research on synthetic audio detection [29]. The output is one-dimensional with time as the axis.

- Onset strength: As used in previous research [16], this perceptible feature represents 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 previous research [16], intensity, also a perceptible feature, is the total power at each point in the audio sample, given in db. This is calculated by creating a fourrier transformation of the sam-

ple and then summing across the frequency-axis for every point on the time-axis. Resulting in a one-dimensional output.

- Pitch-fluctuations: Also classified as a perceptible feature, the pitch fluctuations are 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 [15]. 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 fluctuation at cycle i, pf_i, can be calculated as follows:

$$pf_i = p_i - p_{i-x}.$$

- Jitter features: As defined in previous attempts to identify synthetic audio [6] with perceptible features, jitter-based features measures the absolute variations in fundamental frequency cycles in comparison to the nearest x neighbors. The jitter of a sample in relation to the nearest x samples can be described as follows:

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

  where T_i represents the extracted fundamental frequency length at cycle i, N is the number of fundamental frequency periods and m is $\lfloor \frac{x}{2} \rfloor$

- Shimmer features: Similarly defined to the jitter features, shimmer features, also considered perceptible and are used in the same research. They instead make use of 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. The shimmer of a sample in comparison to the nearest x fundamental frequency cycles is calculated as follows: [6].

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

where A_i is the amplitude at cycle i, N is the number of fundamental frequency cycles in the sample and m is once again $\lfloor\frac{x}{2}\rfloor$

We also chose to include two perceptible features in our hybrid model, due to the performance-enhancing impact that perceptible features can have [6] when paired with perceptible features. These features are the mel-spectrogram and their derivative mel-frequency cepstral coefficients. We choose these features because they are well established in the research and have demonstrated consistent good performance in other examples, [21, 3, 9, 2, 12, 30]. They allow for a spectrographic representation and analysis of an audio sample transformed in a way to better represent human perception of frequencies [21].

4.1.2 Model Architecture

We designed our model architecture to allow the combination of diverse features into one final prediction, regardless of whether the individual features have the same shape. We achieved this 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 and processed though a model that pools into a final output value (which we 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 for the 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 hybrid model must be explained to interpret the importance of each input feature. Further clarification is in

§4.2.

Figure 1 depicts the meta-structure of our hybrid model, with each feature first being processed in its own separate sub-model before being concatenated into the terminus model at the end, which produces the final result. Figure 2 offers a deeper look into the general architecture of a singular sub-model in our method. The sub-models begin with an input layer, taking either a vector or matrix depending on the dimensionality of the feature to which the model is tailored. The inputs are then passed into alternating convolutional and pooling layers, in order to increase the localized pattern detection capability within the sub-models. Figure 3 is a generalized representation of the terminus model, which has a classic perceptron architecture, with a varying number of hidden layers, depending on the experiment.

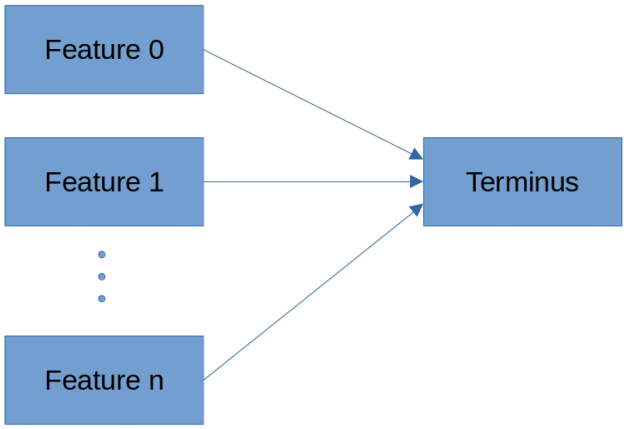

Figure 1: The general architecture for the models used in these experiments, containing n features. The features are processed in their tailored sub-models before being processed together in the terminus model.

4.1.3 Training and Evaluation

We trained the black-box model described in the previous section on the first portion of the dataset, with the exact size of the training set differing depending

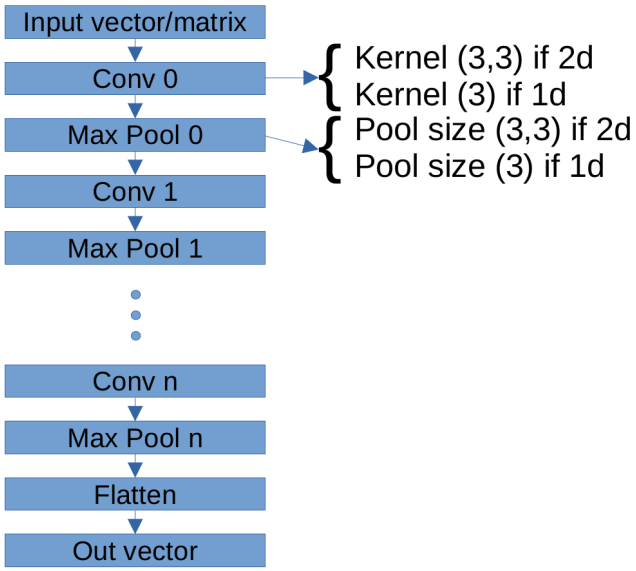

Figure 2: A generalized representation of a sub-model, with the input as either a vector or a matrix followed by alternating convolutional and pooling layers, with the output as a flattened vector. Here, n is the number of convolutional-pooling pairs.

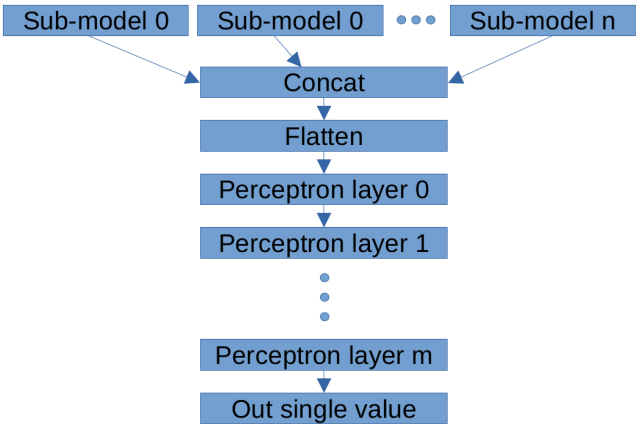

Figure 3: A generalized representation of the terminus model paired with n sub-models and containing m hidden layers.

on the exact experiment (more details in §5). 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, we performed either one or two epochs. 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 our experiments. The default threshold is 0.5, meaning that a result under 0.5 is a negative result and over 0.5 is a positive result. This threshold can however be adjusted, useful for computing certain metrics.

4.2 Explanations

We generate the explanations using the previously described surrogate model with the Local Interpretable Model-agnostic Explanation (LIME) method [23]. This section will describe in detail how individual explanations are generated at a local level for individual samples, as well as how we evaluate the usefulness of the explanations in at a global scale.

4.2.1 Generation of the Explanations

Because of the multi-shaped nature and independence of the input layers of the model, we cannot generate the explanations 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.

We solved the first problem 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.

We approached the second problem by generating intermediate data for the LIME-evaluation, the prediction outputs of the sub-models, in order to determine what the input of the terminus model would be. We generate intermediate data 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, so that we have the intermediate results of each sub-model before the processing of the terminus model. We stored the results of the predictions of the sub-models and used for further analysis. This was done not only for the sample explanation data but also for a random sampling of the training data, so that the LIME explainer could use this data to generate localized estimated of model behavior.

We then summarized The results of the LIME explainer together on a per-feature basis and normalized them 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.

4.2.2 Evaluation of the Explanations

In order to assess if the explanations have the potential to be useful to an end user, we 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 aggregate, global-scale evaluation of local explanations, we propose two metrics, with which we can better assess the usefulness of our method.

In order to contextualize the metrics, we 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 we 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. We will further refer to this metric as importance, and it can be mathematically defined as follows for every $f \in F$:

$$I(f) = \frac{\sum_{s \in S} |w_{fs}|}{|S|}.$$

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

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

5 Experimental Results

Using the methods described in the previous section, we performed experiments training and testing the surrogate model as well as generating many explanations for the purpose of aggregate evaluation. In this section, we will first discuss the dataset that we used for our experiments, the results and performance of the surrogate model, followed by an evaluation of the aggregated explanations.

5.1 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 [20]. We have chosen it for these experiments because of the aforementioned focus on generalization, its relative recency compared to some others and the fact that it has also been used previously by some other experiments, which provides a useful perspective against which we can compare our method.

5.2 Black-Box Model Results

Table 1: Summary of the evaluation results of the experiments

Terminus	Features	Training	Accuracy	EER	AUC
Simple	standard	3474	95.02%	0.03702	0.90297
3 hidden layers	standard	10000	**96.27%**	**0.03214**	0.81763
Simple	standard	23833 2 epochs	90.07%	0.90069	0.78889
Simple	standard	10000 2 epochs	94.43%	0.04408	0.89445
Simple	standard	10000	94.28%	0.04840	0.90182
3 convolutional layers	standard	10000	62.80%	0.37198	0.50051
Simple	expand pitchflucs	10000	93.31%	0.05661	**0.92727**
3 hidden layers	expand pitchflucs	10000	93.87%	0.05317	0.83263

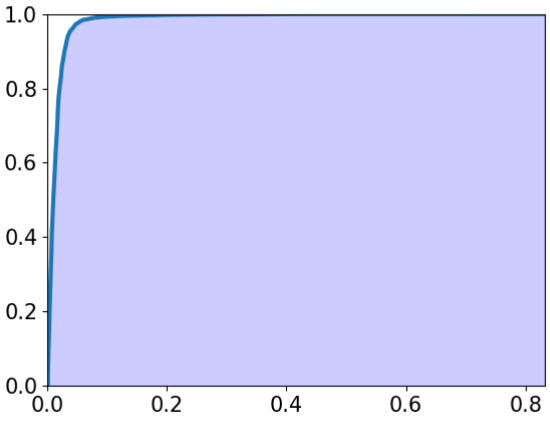

Figure 4: ROC curve of the best-performing model, with 3 hidden layers in the terminus and that standard feature set.

We provide summary of some experiments based on the method described above

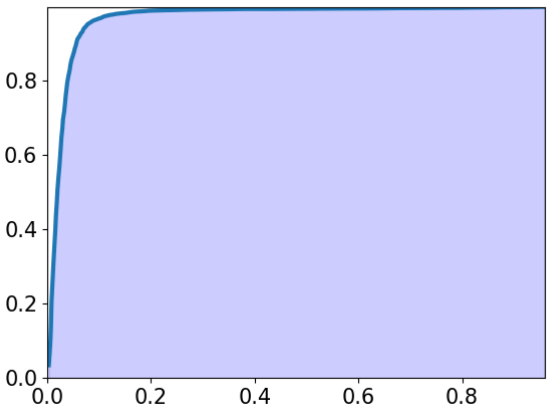

Figure 5: ROC curve of the model with expanded pitch fluctuation features.

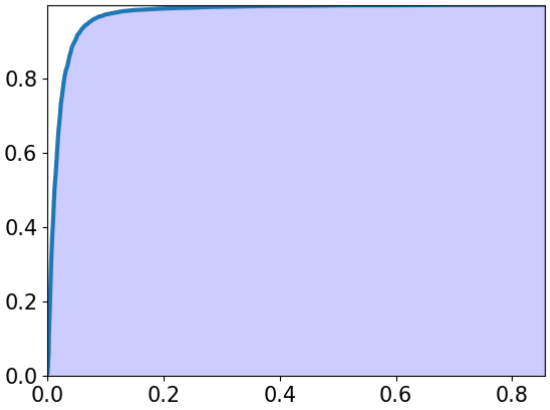

Figure 6: ROC curve of the model with expanded pitch fluctuation features and a terminus with three hidden layers.

in table 1. The table details the following information: The type of terminus model used, where a "simple" terminus model does not include hidden layers; the set of features used, where "standard" is the feature set as described above including only one pitch-fluctuation feature, with expanded pitch fluctuations being the same feature set with the addition of more pitch fluctuation features with different comparison distances; the training batch size and the number of epochs per batch if there were more than one; followed by several performance metrics. Figures 4, 5 and 6 depict the ROC curve of three of the models, which were selected for use in explanation generation in the following section.

The model that resulted in the best accuracy and EER had the standard feature set and was paired with a terminus with three hidden perceptron layers. We trained this model on the first 10000 samples of the dataset, and we evaluated it on the rest of the dataset using the methods previously described. It demonstrated an evaluation accuracy of 96.3% with the default threshold of 0.5, as well as an EER of approximately 0.03214. The best AUC score, however, was achieved in another experiment, where the model had expended pitch fluctuations and a terminus model without hidden layers, with a value of 0.92727.

5.2.1 Comparison to Other Methods

In order to contextualize our results, we compare them to other results produced using tests on the same dataset. Table 2 presents some experimental results that other methods have achieved on this dataset, sorted by source and model architecture. The results are given in EER, and the best result is in bold. In comparison to the other methods, our method performs very well, only beaten out by one other [22], which uses an expanded RawNet2-based network, taking the raw waveform as an input. They trained their method on 70% of the dataset, used 10% for validation and the remaining 20% for testing. This indicates a potential for raw-waveform based learned feature in the accurate detection of audio deepfakes, but does not offer a clear path for explainability, as the model does not make any intrinsic distinction of audio characteristics that humans are

Table 2: A summary of the experimental results of other methods that have been evaluated against the same dataset. (When multiple experiments with one method were done, the best result was used for this table. The results only include evaluations on the full length samples, and not cropped versions, as we tested on the full-length samples)

Model Architecture	EER
STATNet [22]	**0.00199**
Fusion [31]	0.2427
ASSERT [33]	0.2473
Selection [31]	0.2598
ResNet18 [31]	0.2748
LCNN [33]	0.3514
RawGAT-ST [20]	0.37154
MesoInception [20]	0.37414
GMM [33]	0.3749
RawNet2 [20]	0.37819
Transformer [20]	0.43775
CRNNSpoof [20]	0.44500
RawPC [20]	0.45715
MesoNet [20]	0.46939
ResNet18 [20]	0.49759
LSTM [20]	0.53711
LCNN-LSTM [20]	0.61500
LCNN [20]	0.65559
LCNN-Attention [20]	0.66684

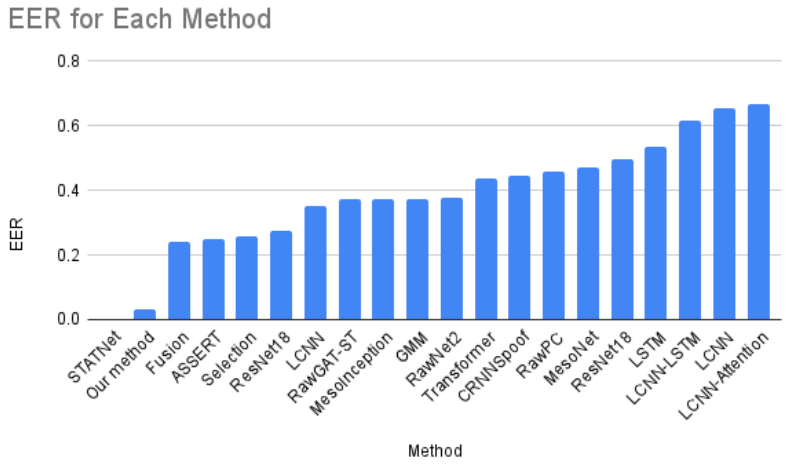

Figure 7: A visual representation of the EER results of our method against some other methods from the literature.

able to perceive. Our model remains, however, competitive, indicating that it can provide a good base for trustworthy explanations.

5.3 Explanation Results

In order to obtain a full picture of the quality of the explanations generated by this method, we have used selected three models from the last section and generated explanations on the first 500 samples of the testing data. The models selected were the model with the best accuracy and EER results, with the standard features and with three hidden layers in the terminus model, the model with expanded pitch-fluctuation features and a simple terminus, which achieved the best AUC result, as well as the model with expanded pitch-fluctuation features and a terminus with three hidden layers. We selected these models were selected because we wanted to evaluate the best performing models, as well as see how the changed pitch-fluctuation feature affected the evaluation of the explanations. We then aggregated and summarized the explanations generated using both of the previously introduced metrics. A summary of the

Table 3: Summary of the aggregated explanation results of the best performing model

Feature	importance	trust
HNRs	0.1140	-0.0543
mel spectrogram	0.1084	-0.0378
MFCC	**0.7029**	**0.3753**
f0 lengths	0.0	0.0
onset strengths	0.0	0.0
intensities	0.0	0.0
pitch fluctuations	0.0	0.0
jitter features	0.1277	0.0250
shimmer features	0.1227	0.0017

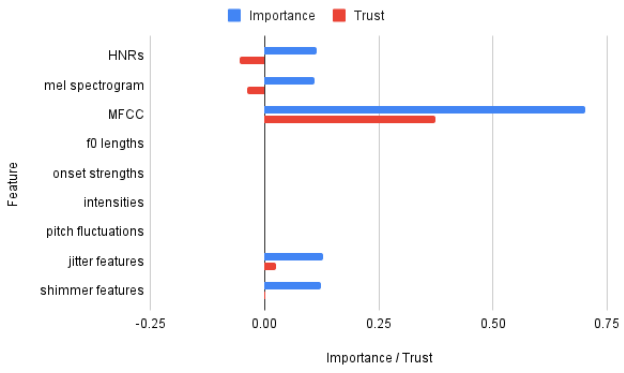

Figure 8: Graphical representation of the aggregated explanation results of the best performing model

Table 4: Summary of the aggregated explanation results of the model with expanded pitch fluctuation features

Feature	importance	trust
HNRs	0.1072	-0.0148
mel spectrogram	0.0428	-0.0187
MFCC	**0.4383**	**0.2345**
f0 lengths	0.0340	-0.0057
onset strengths	0.0101	-0.0027
intensities	0.0021	0.0021
pitch fluctuations	0.0059	0.0020
jitter features	0.2439	0.0019
shimmer features	0.1664	0.0070

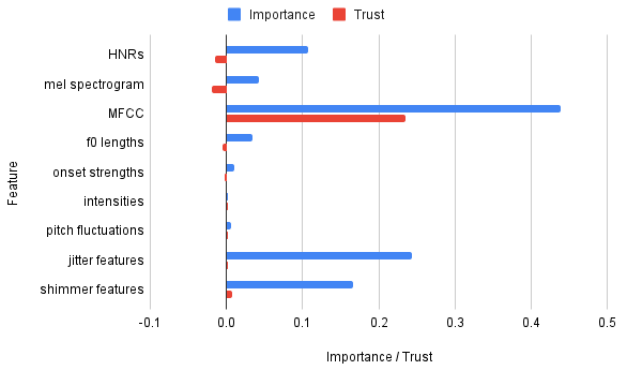

Figure 9: Graphical representation of the aggregated explanation results of model with expanded pitch fluctuation features.

Table 5: Summary of the aggregated explanation results of the model with expanded pitch fluctuation features and a terminus with three hidden layers.

Feature	importance	trust
HNRs	0.2667	-0.0788
mel spectrogram	0.0982	-0.0434
MFCC	**0.2879**	**0.2408**
f0 lengths	0.1690	-0.0521
onset strengths	0.1384	-0.0613
intensities	0.1576	-0.0504
pitch fluctuations	0.0	0.0
jitter features	0.1444	0.0118
shimmer features	0.1747	0.0116

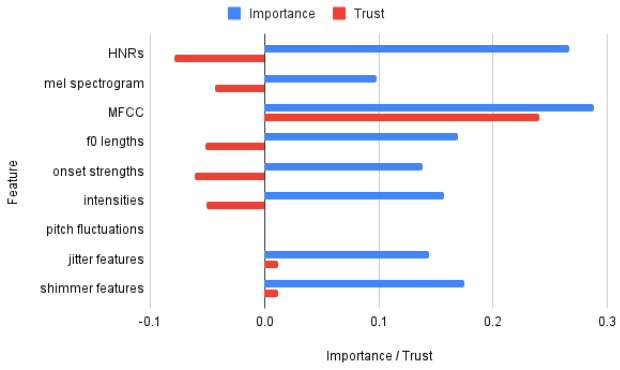

Figure 10: Graphical representation of the aggregated explanation results of the model with expanded pitch fluctuation features and a terminus with three hidden layers.

results of both metrics summarized by the feature is presented in tables 3, 4 and 5. The same statistics are also represented graphically in the figures 8, 9 and 10 respectively.

5.3.1 Model with Standard Feature Set and Complex Terminus

Measuring by both metrics, the MFCCs had the most positive and the most correct influence on the classification results of the model, With the jitter features as a distant second for trust and importance. This indicates that the MFCCs had not only the most influence, but also the most correct influence on the result of the classification, while the other features had either little, no or negative influence on the classification. Jitter features are, however, still notable for being perceptible features with a positive trust value.

5.3.2 Model with Expanded Pitch-Fluctuation Features

This model had comparable results, with the main difference being that all metrics have non-zero values, meaning that all the present features had an influence on the classifications which were used to generate the explanations. As with the previous model, HNRs and mel-spectrograms had negative overall trust values, with the addition of the onset strengths also presenting a negative value, implying that these features have caused more harm than good in the tested classifications. In this example, both jitter and shimmer features are notable for their positive trust values indicating their potential as trustworthy perceptible features for explanations.

5.3.3 Model with Expanded Pitch-Fluctuation Features and Complex Terminus

The model with multiple pitch-fluctuation features in combination with a terminus containing multiple hidden layers fared similarly to the other two, this time with the pitch fluctuations having no influence on the outcome whatsoever. Unlike the previous examples, the trust value of the intensities feature was negative

here. The trust of the jitter and shimmer features, however, remain positive.

6 Discussion

In this section, we will discuss the implications of the experimental results of the previous sections with a special focus being placed on the results of the explanation tests as it relates to the potential explainability for potential end users of such a system.

6.1 Viability and Potential of the Explanations

Because of these models' inclusion of several perceptible features, we made the hypothesis that this method has the potential to produce explanations that can be used as the basis for results more presentable to an end user. As can be seen in the previous section in the tables 3 4, the perceptible features did not contribute as much to the end result of the classification as their imperceptible counterparts, in spite of their increased presence in the overall set of input features. Two features, the HNRs and the mel-spectrograms even consistently had a net-negative impact on the correctness of the classifications in the experiments, according to our trust metric, leaving the question open, if future experiments may perform better without these features. In the case of the model without expanded pitch-fluctuations, four perceptible features, f0 lengths, onset strengths, intensities and pitch fluctuations had an overall average influence on the classification of 0, measured on both metrics, indicating that the model did not learn a meaningful correlation between these features and the authenticity of audio samples in the dataset. These features did have an impact on the performance of the model with expanded pitch fluctuations, but the difference was still minimal. In the case of perceptible features with a positive trust score, the potential remains for their potential use in understandable explanations as, even though the classification needle of the model was not moved significantly by these features, they still demonstrate a certain reliability. In spite of this,

we believe that it is still worth investigating if other kinds of perceptible features will have more of an influence on the overall classification or if there are other imperceptible features that could be used in the place of MFCCs that may provide a better balance between classification correctness and a proportional influence on the end result, so that the usefulness of the explanations can be even stronger than the method using the above feature combinations, and we believe that this kind of model architecture paired with LIME and evaluated with the presented metrics has the potential to be a backbone for future research.

6.2 Limitations

In spite of the potential that we see with this method, it does also present some limitations when it comes to its real-world use, aside from the shortfalls of the perceptible features discussed in the previous section. The first such limitation is in the computational performance of the model. Because of its relative size, the model requires relatively capable hardware in addition to several gigabytes of memory in order to classify samples. This means that the devices this or a similar model can run on are limited, and the possibility of running this or a similar model locally on a handheld device such as a smartphone is limited to none at the time of writing. Another limitation with this method is that, even though it is known that perceptible features such as the ones used in our method are able to be heard by the human ear and are even used in medicine, for example, for diagnoses [6], it is not yet clear to what extent exactly these features can be understood by the average person. Cursory research [28, 25] is already present into what factors humans use to identify fake and real audio samples, as well as their performance, but the named aspects of the samples have not yet been mapped to such perceptible features as the ones used in this method, leaving the question of the usability of such explanations open, in addition to the question of what additional perceptible features can possibly be extracted to mirror what people have specifically identified in fake voice samples. Finally, we have not addressed in our experiments how the features can be presented to

the end user. Certain implementations of LIME already have means to present their results graphically, but the effectiveness of such methods has yet to be studied.

7 Conclusion

To conclude, we have presented a model architecture using a combination of various features, perceptible and imperceptible, with the hypothesis that such a model could be used to generate explanations of its own results using the LIME method that are more useful to a potential end user who is not already familiar with the field. This model architecture resulted in a high level of performance in comparison to several other methods, indicating the potential for consistent and correct classifications. The explanations, however, placed most of the weight on the selected imperceptible features, with most perceptible features either having little, no or negative impact on the end classification with the exception of jitter and shimmer, which offer potential to be used in explanations.

We believe that this is a method that offers a large potential, but is not yet in a fully mature state. We suggest, however, that this method could be changed and improved in the future with other features or other changes to base parameters that could potentially produce more useful explanations while either holding the accuracy high like it is now or even improving it further.

References

[1] Marwan Albahar and Jameel Almalki. "Deepfakes: Threats and countermeasures systematic review". In: *Journal of Theoretical and Applied Information Technology* 97.22 (2019), pp. 3242–3250.

[2] Islam Altalahin et al. "Unmasking the truth: A deep learning approach to detecting deepfake audio through mfcc features". In: *2023 International Conference on Information Technology (ICIT)*. IEEE. 2023, pp. 511–518.

[3] R Anagha et al. "Audio deepfake detection using deep learning". In: *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE. 2023, pp. 176–181.

[4] Sarah Barrington et al. "Single and multi-speaker cloned voice detection: From perceptual to learned features". In: *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2023, pp. 1–6.

[5] 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)*. IEEE. 2022, pp. 97–103.

[6] 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)*. IEEE. 2023, pp. 2182–2188.

[7] Luca Cuccovillo et al. "Open challenges in synthetic speech detection". In: *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2022, pp. 1–6.

[8] Héctor Delgado et al. "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements". In: *The Speaker and Language Recognition Workshop*. ISCA. 2018, pp. 296–303.

[9] Abderrahim Fathan, Jahangir Alam, and Woo Hyun Kang. "Mel-spectrogram image-based end-to-end audio deepfake detection under channel-mismatched conditions". In: *2022 IEEE international conference on multimedia and expo (ICME)*. IEEE. 2022, pp. 1–6.

[10] Wanying Ge et al. "Explaining deep learning models for spoofing and deepfake detection with SHapley Additive exPlanations". In: *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2022, pp. 6387–6391.

[11] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI magazine* 38.3 (2017), pp. 50–57.

[12] Ameer Hamza et al. "Deepfake audio detection via MFCC features using machine learning". In: *IEEE Access* 10 (2022), pp. 134018–134028.

[13] 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* 20.11 (2024), pp. 1–16.

[14] Michael Hind. "Explaining explainable AI". In: *XRDS: Crossroads, The ACM Magazine for Students* 25.3 (2019), pp. 16–19.

[15] 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)*. IEEE. 2023, pp. 01–06.

[16] 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*. 2022, pp. 35–41.

[17] Xin Liu et al. “Hidden-in-wave: A novel idea to camouflage ai-synthesized voices based on speaker-irrelative features”. In: *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE. 2023, pp. 786–794.

[18] Xuechen Liu et al. “Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 2507–2522.

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

[20] Nicolas M Müller et al. “Does audio deepfake detection generalize?” In: *arXiv preprint arXiv:2203.16263* (2022).

[21] 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)*. IEEE. 2022, pp. 1–6.

[22] Rishabh Ranjan, Mayank Vatsa, and Richa Singh. “Statnet: Spectral and temporal features based multi-task network for audio spoofing detection”. In: *2022 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2022, pp. 1–9.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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.

[24] Aruna Sankaranarayanan et al. “The presidential deepfakes dataset”. In: *CEUR Workshop Proceedings*. Vol. 2942. CEUR-WS. 2021, pp. 57–72.

[25] Filipo Sharevski et al. “Blind and Low-Vision Individuals’ Detection of Audio Deepfakes”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2024, pp. 4867–4881.

[26] Namosha Veerasamy and Heloise Pieterse. “Rising above misinformation and deepfakes”. In: *International Conference on Cyber Warfare and Security*. Vol. 17. 1. Academic Conferences International Limited. 2022, pp. 340–348.

[27] Xin Wang et al. “ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech”. In: *Computer Speech & Language* 64 (2020), p. 101114.

[28] Kevin Warren et al. “” Better Be Computer or I’m Dumb”: A Large-Scale Evaluation of Humans as Audio Deepfake Detectors”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2024, pp. 2696–2710.

[29] 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*. 2022, pp. 19–26.

[30] Xinrui Yan et al. “An initial investigation for detecting vocoder fingerprints of fake audio”. In: *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. 2022, pp. 61–68.

[31] Yujie Yang et al. “A robust audio deepfake detection system via multiview feature”. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 13131–13135.

[32] Jiangyan Yi et al. “Add 2022: the first audio deep synthesis detection challenge”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 9216–9220.

[33] Jiangyan Yi et al. “Audio deepfake detection: A survey”. In: *arXiv preprint arXiv:2308.14970* (2023).