

TOWARD ROBUST REAL-WORLD AUDIO DEEPFAKE DETECTION: CLOSING THE EXPLAINABILITY GAP

Anonymous authors

Paper under double-blind review

ABSTRACT

The rapid proliferation of AI-manipulated or generated audio deepfakes poses serious challenges to media integrity and election security. Current AI-driven detection solutions lack explainability and underperform in real-world settings. In this paper, we introduce novel explainability methods for state-of-the-art transformer-based audio deepfake detectors and open-source a novel benchmark for real-world generalizability. By narrowing the explainability gap between transformer-based audio deepfake detectors and traditional methods, our results not only build trust with human experts, but also pave the way for unlocking the potential of citizen intelligence to overcome the scalability issue in audio deepfake detection.

1 INTRODUCTION

The rapid proliferation of AI-generated audio deepfakes poses a growing threat to media integrity, personal security, and democratic processes, with critical implications for misinformation, fraud, and election security. Although state-of-the-art detection solutions have demonstrated promising results on benchmark datasets, they often fall short in real-world scenarios due to poor generalization and a lack of explainability. Current approaches, such as those used in ASVspoof and related competitions, focus heavily on detection performance within constrained environments, but their efficacy diminishes significantly when encountering diverse and unseen samples.

In this work, we highlight the limitations of existing deepfake detection methods and introduce an attention roll-out mechanism that addresses these shortcomings by providing improved explainability for transformer-based audio classifiers. Recent deepfake audio attacks, such as those used to discredit Marti Bartes in Mexico or to influence U.S. elections by impersonating President Joe Biden, emphasize the necessity for solutions that not only detect these manipulations but also offer transparent and interpretable explanations (Goodman, 2024; Staff, 2024). These would, in turn, foster trust with both experts and the general public.

To bridge the gap between controlled benchmark results and real-world applicability, we deliver a novel benchmark that evaluates the generalization capabilities of deepfake audio classifiers by training on the ASVspoof 5 dataset and testing on the FakeAVCeleb dataset. This benchmark provides a more realistic evaluation of model robustness, simulating conditions where the data distributions of training and testing are substantially different. Additionally, we compare and contrast various explainability methods, offering a conceptual contribution that defines key requirements for explainability in deepfake audio detection. Our findings not only reveal the strengths and weaknesses of each approach but also lay the groundwork for future research.

Through this comprehensive evaluation, we outline several open challenges that need to be addressed to improve generalization and explainability, thereby enhancing trust in deepfake detection systems. By establishing these benchmarks and defining conceptual requirements, we hope to catalyze future developments in this critical research area to effectively safeguard against the evolving threat of audio deepfakes.

We list our novel contributions as follows:

- A conceptual explainability framework for deepfake audio detection (Section 4).
- Empirical evaluations of novel explainability methods for audio transformers (Section 5).
- A generalizability benchmark for deepfake audio classifiers (Section 6).

054 **2 RELATED WORK**

055

056 **Traditional Machine Learning Approaches** The early days of deepfake audio detection were
 057 dominated by traditional classifiers, such as support vector machines (SVM) and Gaussian mixture
 058 models (GMM), and traditional signal processing features. These works generally use subsets of
 059 the various hand-crafted features described above, and they generally perform well with academic
 060 datasets, in which the distributions of features between real and deepfake audio are relatively eas-
 061 ily separable. Observing that these models typically do not generalize well when presented with
 062 deepfake audio from unseen distributions, Zhang et al. (2021) propose using an SVM to learn a
 063 tight boundary around the features of real audio by only training on real audio (Zhang et al., 2021).
 064 This method outperforms most other methods on the ASVspoof 2019 dataset, but it is unlikely to
 065 perform as well when the true audio varies more widely (e.g., non-English speech, speakers with
 066 accents, non-adult speakers, etc.).

067 Recent work also suggests that using an ensemble of gradient-boosting decision trees (GBDT) may
 068 be more robust to unseen data as well as boast faster inference times than both SVM and GMM (Bird
 069 & Lotfi, 2023). In a 2023 work by Bird & Lotfi (2023), the authors demonstrate the power of
 070 the GBDT, reporting accuracies of 99.3% on the DeepVoice dataset and inference times of 0.004
 071 seconds for 1 second of input speech (Bird & Lotfi, 2023). In their work *Real-Time Detection of*
 072 *AI-Generated Speech for Deepfake Voice Conversion*, they also explore features importances and
 073 the statistical characterizations of real and deepfake audio. Another recent work by Togootogtokh
 074 & Klasen (2024) employs a GBDT for deepfake audio classification task with a custom dataset
 075 comprised of true samples from the LJ Speech Dataset and deepfake samples generated with various
 076 HuggingFace TTS models (Togootogtokh & Klasen, 2024).

077 **Self-Supervised Embedding-Based Classifiers** A few recent works use self-supervised embed-
 078 ding features as the basis of their classification algorithms, most of which use Wav2Vec features (Shi
 079 & Yamagishi, 2021). Tak et al. (2022) fine-tune a transformer model with a Wav2Vec front-end
 080 and report the lowest equal error rates for both the ASVspoof 2021 Logical Access and Deepfake
 081 databases. Xie et al. (2021) use Wav2Vec features as input to a Siamese neural network that they
 082 train to distinguish whether the speech samples in a pair belong to the same category. This work
 083 reported state-of-the-art results on the ASVspoof 2019 dataset (Chen et al., 2022). Some other re-
 084 cent works use HuBERT features, and Wang & Yamagishi (2022) compare Wav2Vec-, XLS-R-, and
 085 HuBERT-based features (Yi et al., 2023). Most recently, Le et al. (2024) combine AST features
 086 with a GBDT ensemble to detect deepfake audio, which they plan to use for a continuous learning
 087 approach.

088 **Explainability for Audio** The literature for audio explainability is limited (Akman & Schuller,
 089 2024). Though general explainability methods such as LIME and SHAP can also be used for mod-
 090 els trained on audio data, very few examples of attempts for audio explainability exist in the liter-
 091 ature (Ribeiro et al., 2016; Lundberg & Lee, 2017). Of those exceptions, Yanchenko et al. (2021)
 092 measure the similarity of deep features to hand-crafted features and attempt explain deep convolu-
 093 tional features as they relate to traditional signal processing ones. Most recently, Becker et al. (2024)
 094 introduced AudioMNIST, a novel audio dataset consisting of 30 000 audio samples of spoken digits,
 095 and proposed using Layer-wise Relevance Propagation (LRP) to explain neural network classifica-
 096 tions. They also investigate using a combination of visual and aural explanations, and find aural
 097 explanations promising if well-designed.

098 For additional related work on in-painted deepfake audio and convolutional neural network (CNN)
 099 based approaches to deepfake audio detection, refer to Appendix A

100 **3 BACKGROUND**

101

102 **3.1 TRADITIONAL METHODS**

103

104 In the previous section, we reviewed a variety of traditional methods for deepfake audio detection.
 105 Here, we will focus on features used in traditional methods and define the gradient boosting decision
 106 tree method introduced in the previous section.

108

Signal Processing Features Traditional methods typically rely on “hand-crafted” features, which
109 are not learned by neural networks but calculated with standard signal processing methods.
110

111

Mel-Frequency Cepstral Coefficients (MFCCs) are a set of 10-20 features that capture the timbre
112 of audio samples using the perceptual Mel scale, reflecting how humans perceive pitch. Computing
113 MFCCs involves applying a pre-emphasis filter, dividing the signal into 20 ms frames, applying
114 a Hann window, and performing a Fast Fourier Transform (FFT). The resulting power spectrum
115 is passed through a Mel-scale filter bank, then logarithmically transformed, and a Discrete Cosine
116 Transform (DCT) is applied to generate MFCCs, providing a compact representation of spectral
117 properties. Other spectral features include spectral centroid (indicating “brightness” by representing
118 the center of spectral mass), bandwidth (spread around the centroid), and roll-off (frequency below
119 which 85% of spectral energy is contained). These features are useful for tasks like music genre
120 classification and speech analysis.

121

Chroma features capture harmonic content by mapping the spectrum to twelve pitch classes (C, C#,
122 D, etc.) and are invariant to timbre changes, making them ideal for analyzing musical elements.
123 Zero Crossing Rate (ZCR) measures the rate of sign changes in an audio signal, aiding in pitch and
124 speech detection, while Root Mean Square (RMS) captures energy or loudness, offering a measure
125 of signal strength.

126

Gradient Boosting Decision Trees Gradient boosting decision trees (GBDT) combine three core
127 concepts in traditional machine learning: ensemble learning, boosting, and gradient descent. As
128 an *ensemble* method, GBDT combine the predictions of several weak learners—typically decision
129 trees—to produce a stronger overall prediction. The *boosting* aspect of GBDT means that the model
130 is constructed sequentially, such that at each iteration a new weak learner is added to correct for
131 the previous learners’ mistakes. Finally, the *gradient* aspect of GBDT reflects the optimization
132 technique used to find the best fit for each new weaker learner added to the ensemble.

133

At initialization, the GBDT is an ensemble containing a single weak learner that makes a prediction.
134 For binary classification, the model is initialized with a constant value $\hat{F}_0(x)$, the log-odds of the
135 positive class. At each iteration, until the maximum number of weak learners permitted in the
136 ensemble is reached, the pseudo-residuals are calculated, which are the negative gradients of the
137 loss function:

138
139

$$r_{im} = - \left[\frac{\partial L(y_i, \hat{F}_{m-1}(x_i))}{\partial \hat{F}_{m-1}(x_i)} \right] = y_i - \hat{p}_{m-1}(x_i), \quad (1)$$

140

where r_{im} is the residual for the i -th instance at iteration m , $L(y_i, \hat{F}_{m-1}(x_i))$ is the loss function
141 to be minimized, $\hat{p}_{m-1}(x_i)$ is the predicted probability of the positive class for the i -th instance at
142 iteration $m - 1$, and y_i is the true label. A new weak learner, $h_m(x)$, is then fitted to this residual
143 and its impact scaled to avoid overfitting, such that the model at iteration m is given by:

144

145

$$\hat{F}_m(x) = \hat{F}_{m-1}(x) + \nu \cdot h_m(x), \quad (2)$$

146

147

where ν is the learning rate. At the end of the learning process, the predicted probability of a data
sample’s membership in the positive class is given by:

148

149

150

$$\hat{p}(x) = \frac{1}{1 + \exp(-\hat{F}_m(x))}. \quad (3)$$

151

The final classification is given by thresholding the prediction probability p .

152

153

3.2 TRANSFORMERS

154

Since the publication of *Attention is All You Need* (Vaswani et al., 2017), transformer models have
155 become increasingly widespread for a wide variety of tasks, though most notably text generation.
156 As suggested by the title of that 2017 paper, *attention* is the core of the transformer architecture.
157 Given a sequence of input embeddings E , a transformer model encodes tokens using a *self-attention*
158 mechanism, which allows the model to focus on different parts of the input sequence when encoding
159 a particular token (Vaswani et al., 2017). A single self-attention operation (or head) is defined by:
160

161

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V, \quad (4)$$

where \mathbf{X} is the matrix of input embeddings, $Q = \mathbf{X}W_Q$ is called the query matrix, $K = \mathbf{X}W_K$ is called the key matrix, $V = \mathbf{X}W_V$ is called the value matrix, W_Q, W_K, W_V are learned weight matrices, and d_k is the dimensionality of the keys (Vaswani et al., 2017). In order to facilitate learning multiple different features, multiple self-attention heads are used. Multi-attention is then defined by:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O, \quad (5)$$

where $\text{head}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i})$ and W_O is the output projection matrix. The output of the multi-head attention is passed into a Feed-Forward Neural Network (FFN) defined by:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2, \quad (6)$$

where W_1 and W_2 are learned weight matrices, b_1 and b_2 are biases, and ReLU is the activation function. Each multi-head attention or FFN sub-layer is followed by layer normalization defined by:

$$\text{Layer Output} = \text{LayerNorm}(x + \text{Sub-layer}(x)). \quad (7)$$

The final transformer architecture stacks multiple of these multi-head attention and FFN layers to capture increasingly complex patterns.

Since the transformer network does not maintain input order, an additional positional embedding token is appended to each patch to allow the model to maintain the spatial structure of the input spectrogram (Gong et al., 2021). We also often prepend a [CLS] token to the input. After passing through all transformer layers, the final [CLS] token aggregates information from the entire input sequence for the final prediction.

The aforescribed operations are universal to the transformer architecture, but methods of creating the input sequence vary widely. The transformer architecture relies on the creation of *tokens* from raw input data and the learned *attention* between those tokens. For natural language tasks, tokens typically represent individual words. For image tasks, tokens are typically pixel patches. For audio tasks, there are a variety of approaches. Here, we introduce two popular mechanisms for generating input embeddings from audio data.

Self-Supervised Audio Features One of the most popular feature generators is Wav2Vec 2.0, produced and published by Meta in 2019 (Schneider et al., 2019). Wav2Vec uses a 7-layer CNN generate latent feature encodings, which are then put into a quantization module to make the final tokens which will be fed to the transformer (Schneider et al., 2019). As speech is continuous, Wav2Vec strives to automatically infer discrete speech units with the quantization module, such that tokens can be formulated as they would be in natural language, representing complete but discrete data units (Schneider et al., 2019).

In contrast, the Audio Spectrogram Transformer (AST), which was the first to move away from traditional convolutional neural network approaches, simplifies audio token generation (Gong et al., 2021).

As seen in Figure 1, the AST first transforms an input audio wave of length t seconds into a sequence of 128-dimensional Mel features. The resultant $128 \times 10t$ spectrogram is then used as input to the AST. The spectrogram is then split into a sequence of N 16×16 patches, with overlap in both time and frequency dimensions. Each 16×16 patch is flattened into a single-dimensional patch of size 768 with a linear layer.

Once the audio is formatted in this way, the AST feeds the input sequence to a Vision Transformer (ViT), an image transformer model trained on ImageNet (Gong et al., 2021). This approach essentially translates the audio signal into an image and then uses a transformer pre-trained on image data to make classifications.

Notably, both the AST and Wav2Vec models are pre-trained only with real audio data, only by finetuning additional layers are they suitable for classifying deepfake audio.

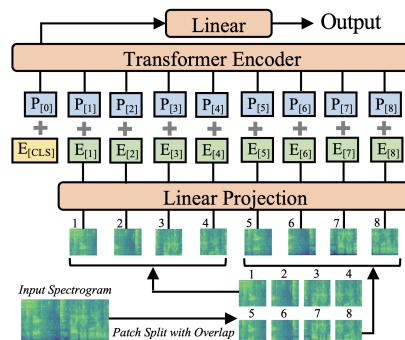


Figure 1: Diagram of the audio spectrogram transformer architecture introduced by Gong et al. (2021)

216 3.3 EXPLAINABILITY AND INTERPRETABILITY
217218 Though *explainability* and *interpretability* are often used interchangeably, we distinguish between
219 them. Interpretability emphasizes transparency and clarity, focusing on making the internal mecha-
220 nisms of a model comprehensible, such as understanding the coefficients of a linear regression model
221 or the structure of a decision tree (Rudin, 2019). In contrast, explainability goes beyond understand-
222 ing the internal, mathematical mechanism and should provide explanations of a model’s predictions
223 in a human-understandable way, even if the model itself is a complex “black box” (Rudin, 2019).224 Explainability oftens involves using post-hoc methods, LIME (Local Interpretable Model-agnostic
225 Explanations) or SHAP (SHapley Additive exPlanations), to approximate and elucidate the model’s
226 decisions without revealing the exact inner workings (Ribeiro et al., 2016; Lundberg & Lee, 2017).
227 Importantly, the interpretability of a model can contribute to its explainability, but a model’s being
228 interpretable does not necessarily imply that it is explainable.
229230
231 **Explainability for GBDT** A key advantage of traditional methods is that they are interpretable.
232 With a gradient boosting decision trees, or any other tree-based ensemble method, feature impor-
233 tances can be calculated. As described in Algorithm 1, feature importances are calculated by mea-
234 suring the model’s change in performance after each feature is permuted (Pedregosa et al., 2011). To
235 stabilize results, we permute each feature multiple times and use the mean and standard deviation of
236 each feature’s importance.
237238 However, the importance of a given feature may be obscured by the permutation feature importance
239 algorithm if multiple features are multicollinear, as is the case for the audio signal features described
240 in the previous section. Intuitively, permuting one feature will have little impact on the model’s
241 performance if the same, or very similar, information is available to the model through another
242 non-permuted feature. To combat this issue, we perform hierarchical clustering on the Spearman
243 rank-order correlations between features and keep a single feature from each cluster. This way,
244 when a feature is permuted, there should be no other non-permuted feature containing duplicate
245 information.
246247 **Explainability for Transformer Models** A challenge of working with transformer methods is the
248 lack of interpretability. Though they boast much better performance than traditional methods, again
249 see Appendix D, their output is of a “black-box” nature. In order to facilitate citizen intelligence,
250 detection methods must deliver human interpretable explanations that are sample-specific, such that
251 the explanation is not invariant to the input sample; time-specific, such that the explanation includes
252 specific timestamps that localize the distinguishing features; and feature-specific, such that specific
253 aspects like unusual noise or errant formants can be identified as unnatural.
254255 3.4 MODEL PERFORMANCE FOR DEEPFAKE AUDIO CLASSIFICATION
256257 As mentioned in the previous section, the impressive performance of Wav2Vec-based transformers
258 has already been demonstrated by Tak et al. (2022). Le et al. (2024) recently presented similar
259 results using AST features with a GBDT. To deliver on the goal of explainable results that maintain
260 the performance of transformer-based methods, we validate the performance of the finetuned AST
261 model, the finetuned Wav2Vec model, and the traditional feature-based GBDT on the ASVspoof5
262 and FakeAVCeleb datasets.
263264 These results, as shown in Appendix D, demonstrate the superior performance of the Wav2Vec and
265 AST models as well as report the GBDT baseline. Though the power of a finetuned Wav2Vec
266 transformer has already been established, the results reported in Appendix D are state-of-the-art for
267 the FakeAVCeleb dataset. For the sake of robustness, we also report results on the ASVspoof 5
268 dataset when the data has been compressed and rerecorded to measure the effect that these common
269 data augmentations have on model performance.

270 **4 METHODS**
 271

272 In this section, we introduce our proposed methods for audio explainability, with which we will
 273 experiment in the following section. We appropriate methods for vision and natural language ex-
 274 plainability and translate them to the audio domain.
 275

276 **4.1 OCCLUSION**
 277

278 Occlusion is a technique used for vision model explainability, particularly with deep learning models
 279 that might otherwise be considered “black boxes”. The core idea is to iteratively occlude, or block
 280 from view, parts of the input data, measure how the model’s prediction changes, and, ideally, identify
 281 which parts of the input data are most important.

282 Consider some input $X = [x_1, x_2, \dots, x_n]$, where X represents the original input and each x_i
 283 represents a subsection (perhaps a pixel, patch, or token) in X , and some model f . First, we generate
 284 a baseline prediction $\hat{y} = f(X)$, which will serve as the point of comparison. Then, we iteratively
 285 mask each x_i from the input, such that:

286
$$\mathbf{X}_{\text{occluded}}^{(i)} = \mathbf{X} \odot \mathbf{M}_i, \quad (8)$$

 287

288 where \mathbf{M}_i is a mask that occludes the i -th subregion of the input and \odot represents element-wise
 289 multiplication. After this operation, the occluded region will be replaced with some specified value
 290 (e.g., 0, 1, or a mean of the feature across all samples). For each occluded input, the model makes a
 291 new prediction given by:

292
$$\hat{y}^{(i)} = f(\mathbf{X}_{\text{occluded}}^{(i)}). \quad (9)$$

293 The intuition is that if a change in the model’s prediction is observed when a region is occluded, that
 294 region is likely important. After occluding different regions and observing changes in the model’s
 295 behavior, the results can be visualized in a heatmap. This method has been used for understanding
 296 importance in vision data, but we introduce it here for audio data. We treat the Mel-spectrogram
 297 representation of each audio sample as an image and occlude sections of the spectrogram to deter-
 298 mine which parts of each audio sample are most important for the transformer’s classification. This
 299 method delivers on all three aspects of sufficient explainability defined in 3.3.
 300

301 **4.2 ATTENTION VISUALIZATION**
 302

303 We also appropriate an explainability method introduced for use with natural language. As previ-
 304 ously discussed, transformer models rely on a self-attention mechanism to understand the rela-
 305 tionships between different parts of the input sequence. The attention mechanism assigns a weight to
 306 each token, which reflects the importance of each token in relation to every other token.
 307

Recall that attention is defined by:

308
$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V, \quad (10)$$

 309

310 where \mathbf{X} is the matrix of input embeddings, $Q = \mathbf{X}W_Q$ is called the query matrix, $K = \mathbf{X}W_K$
 311 is called the key matrix, $V = \mathbf{X}W_V$ is called the value matrix, W_Q, W_K, W_V are learned weight
 312 matrices, and d_k is the dimensionality of the keys (Vaswani et al., 2017). After applying a softmax
 313 on the unnormalized attention weights, we are left with a normalized $d_k \times d_k$ matrix of weights that
 314 can be visualized as a heatmap. In such a visualization, the x -axis represents position in the input
 315 sequence (or token ID), the y -axis represents the tokens for which attention is computed, and the
 316 color intensity at some (i, j) represents the attention weight for token i on token j , where greater
 317 attention is read to reflect higher importance.
 318

319 A limitation of attention visualization is that it is done per layer per head, which can make it dif-
 320 ficult to observe the overall model’s attention. To combat this, Abnar & Zuidema (2020) proposed
 321 “attention rollout” to trace the distribution of attention across multiple or all of the model’s layers.
 322 This method gives us a more complete view the model’s distribution of attention.
 323

Consider an L -layer transformer with attention matrices W_l for $l \in \{1, 2, \dots, L\}$, where each W_l
 represents the attention between different tokens at that layer. We compute a cumulative attention

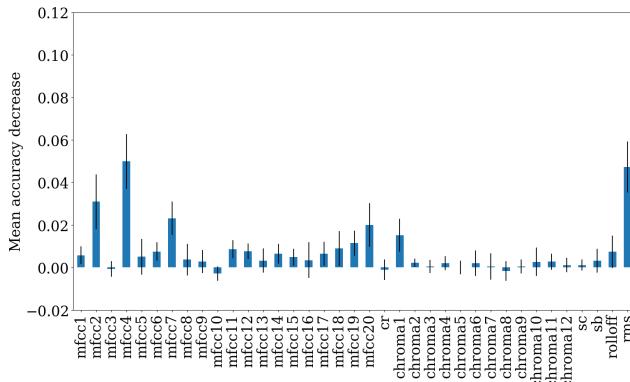


Figure 2: GBDT feature importances as measured by mean accuracy decrease with standard deviations for 6.0-second classifier.

matrix by multiplying the attention matrices across the layers, such that:

$$W^{\text{rollout}} = W^{(1)} W^{(2)} \dots W^{(L)}. \quad (11)$$

Once we have computed the cumulative attention matrix W^{rollout} , it can be visualized similarly to the single attention weights. We also extract the attention weights specific to the [CLS] token to understand what parts of the input sequence were most relevant for the final classification. Attention visualization and roll-out were introduced for use with natural language tokens. In this paper, we adapt this method for use with audio tokens.

5 EXPERIMENTS

We experiment with three models: an ensemble of gradient boosting decision trees (GBDT), an AST-based transformer, and a Wav2Vec-based transformer. An extended discussion of the ASVspoof 5 and FakeAVCeleb datasets can be found in Appendix B, while a report of hyperparameters for each of our three models can be found in Appendix C.

5.1 EXPLAINABILITY FOR GBDT

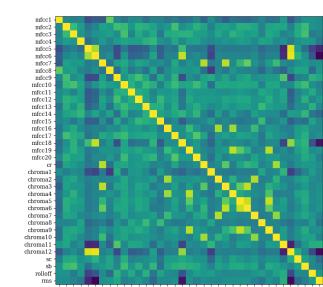
A well-noted advantage of ensembles of decision trees, as the GBDT is, is the ability to calculate feature importances. In their recent deepfake audio classification with GBDT work, Bird et al. report feature importances and draw meaning from them Bird & Lotfi (2023). Here, we mimic their approach to explain the behavior of the GBDT and to isolate some aspect of the audio sample as a signature of its deepfake classification.

We compare the GBDT feature importances, calculated with the permutation importance algorithm, for the models trained with 1.0, 3.0, and 6.0-second audio samples (Pedregosa et al., 2011). The results for 6.0-second classifier are shown in Figure 2, while supplementary results for the 1.0- and 3.0-second classifiers can be found in subsection E.1. As seen in Figure 2, the second MFCC, fourth MFCC, and RMS features are the most influential in the GBDT’s decision-making. Recall that the RMS feature is most closely associated with the loudness of an audio sample. We find the high importance of the RMS troubling as our intuition suggests that loudness should not inherently be a characteristic of deepfake audio.

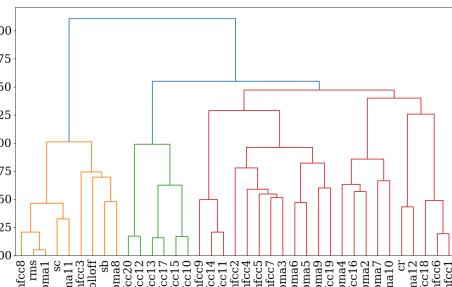
We retrain the GBDT with only the three most important features, the second MFCC feature, the fourth MFCC feature, and the RMS, and evaluate the model’s performance when given only these three features. We observe some performance degradation; when asked to classify 6.0-second audio samples, the model is only able to achieve 70.0% precision, recall, and accuracy (compared to 89.0% precision, recall, and accuracy when trained with all features).

As the vast majority of features are estimated to have a less than 2% impact on overall accuracy, we also calculate feature correlations. As shown in Figure 3, many of the features are correlated.

378
379
380
381
382
383
384
385
386
387
388
389



(a) Feature Correlation



(b) Feature Clusters

390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

Figure 3: GBDT feature correlations and clusters for 6.0-second classifier. In these figures, sc refers to the spectral centroid, sb refers to the spectral bandwidth, cr refers to the ZCR, $mfcc_i$ refers to the i -th MFCC feature, and $chroma_i$ refers to the i -th chroma feature.

We perform a hierarchical clustering of the features using Ward’s linkage, and observe that there are only a few clusters of features. Perhaps interestingly, there is no clustering of the features in which our three most important features, MFCC2, MFCC4, and RMS, are all in different clusters.

We select the most important feature from each cluster and retrain the GBDT. Retrained with RMS, MFCC 20, and MFCC 4, the GBDT achieves precision of 64.3%, recall of 64.0%, and accuracy of 63.8%. As performance was better when using the three most important features, compared to using important features with more spread, it does seem that the second MFCC feature is actually critical to the model’s decision-making. Though it is not feasible to attribute a single frequency to a single MFCC, as an MFCC is a compact representation of a spectral shape across the Mel-scale filterbank, the second MFCC captures low-frequency details, such as overall spectral slope and formant information. Each formant corresponds to a resonance in the vocal tract, and it is intuitive that deepfake audio would have anomalous resonance. While this explanation points to a potential source of inherent distinction between real and deepfake audio, it only provides an explanation as to what the model is attentive to in general rather than sample-level specificity. While the GBDT is interpretable, we find that it is not sufficiently explainable to be useful to a non-technical audience.

5.2 EXPLAINABILITY FOR AUDIO TRANSFORMERS

Occlusion As the AST model converts the raw audio signals into a spectrogram input, it is well-suited to visualization. We perform occlusion with box size (200, 50) and stride size (100, 25). Importance is measured by the magnitude of change in the predicted probability of the sample being in the positive class when a section is occluded.

As shown in Figure 4, the importance is greatest for the padded regions—regions that theoretically contain no predictive information. The audio samples shown in Figure 4 are of length 6.0-seconds, but we observe this phenomenon when calculating importance by occlusion for all sample lengths. For each sample length, the most important regions (as measured by the occlusion method) are always placed at the beginning of the padded region. This result is obviously unhelpful in explaining the model’s decision-making, but it is suggestive of a phenomenon posited by Wu et al. (2021), in which transformer models store global information at locations in the feature space which are consistent, such that the weight information there is always propagated (Wu et al., 2021).

Attention Visualization Another approach to explaining transformer models is visualizing the distribution of the attention weights over the input data. For each layer, as described in Section 3, there is an attention matrix that represents the amount of attention between each pair of tokens. This method has been employed for image and text data, but not, as of yet, to audio data (Dosovitskiy et al., 2021).

We visualize the each layer’s attention matrix for the same bonafide and spoofed samples shown in Figure 4. The resultant visualizations can be found in subsection E.2. Similar to results recorded on Vision Transformer (ViT) by Dosovitskiy et al. (2021), we observe that the attention at early

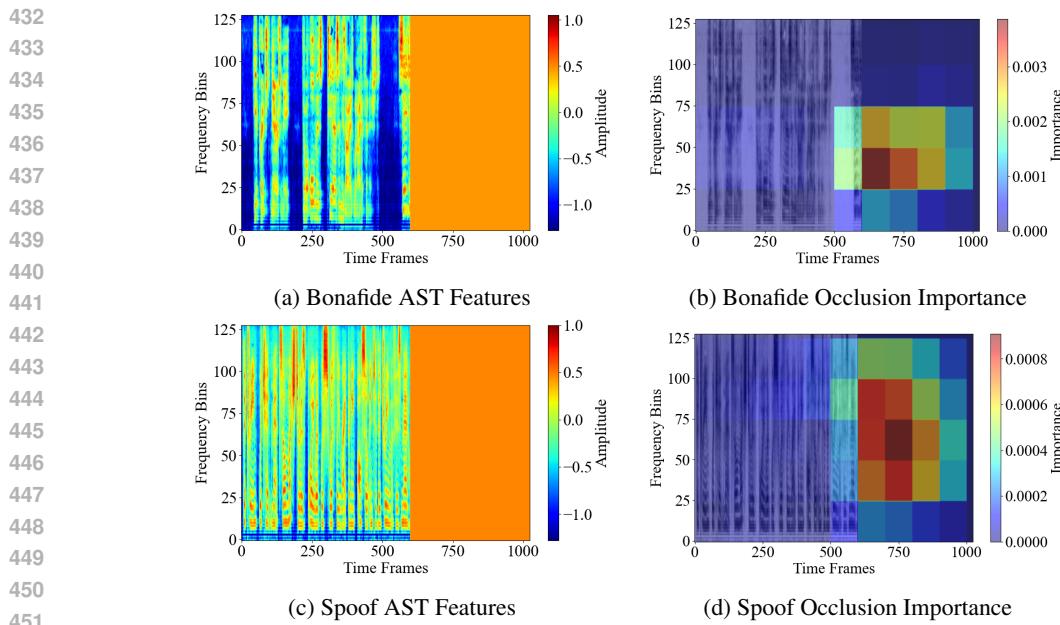


Figure 4: Importance measured by occlusion for 6.0-second audio samples.

layers is quite local with a relatively small receptive field while the attention at later layers is widely distributed (Dosovitskiy et al., 2021).

To better understand the distribution of attention across the entire model, we compute the attention roll-out, which allows us to observe the overall attention flow on each of the input tokens by recursively multiplying the weight matrices of all the layers (Abnar & Zuidema, 2020).

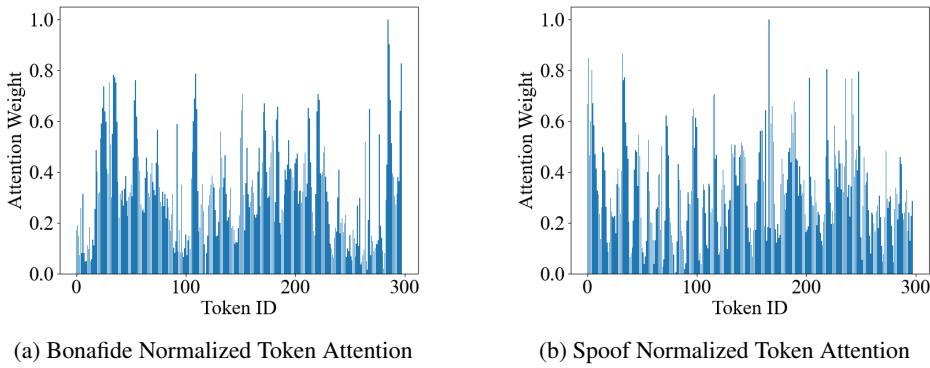


Figure 5: Distribution of attention for 6.0-second audio samples.

By normalizing the attention for the [CLS] classification token, we are able to visualize which input tokens are most important for the model’s overall classification, as seen in Figure 5. As each Wav2Vec token represents about 20 milliseconds of audio signal, we can pinpoint specific frames that were instrumental in the classification and inspect them more closely. Figure 5 allows us to identify very short audio frames that are most influential in the model’s prediction, and we observe that influential tokens typically appear in groups.

6 GENERALIZABILITY BENCHMARK

Finally, we introduce a novel benchmark to evaluate the generalization capabilities of deepfake audio classifiers to unseen data, which is critical for deploying reliable deepfake detection systems.

This benchmark measures the robustness and transferability of models across different datasets with varying characteristics, simulating real-world scenarios where deepfake classifiers may encounter out-of-domain samples.

We train each of our models with the ASVspoof 5 dataset and evaluate each model’s ability to classify samples from the FakeAVCeleb dataset. As a reminder, the evaluation performance of each model on ASVspoof 5 and FakeAVCeleb independently can be found in Appendix D. We compare performance between our baseline classifier, the GBDT, and the two transformer methods. We use 3 000 evaluation samples from FakeAVCeleb, balancing the classes to an equal number of both bonafide and spoof audio samples.

Table 1: Performance comparison of ASVspoof-trained models on FakeAVCeleb data.

Model	Class	Precision	Recall	F1
GBDT	bonafide	0.50	0.58	0.54
	spoof	0.51	0.51	0.51
AST	bonafide	0.85	0.84	0.85
	spoof	0.85	0.86	0.85
Wav2Vec	bonafide	0.73	0.98	0.84
	spoof	0.97	0.63	0.77

When using the GBDT classifier trained on 6.0-second samples of the original ASVspoof dataset, we observe that an overall accuracy of 51%, which indicates that the GBDT does essentially no better than random guessing between the two classes. Table 1 reports the precision, recall, and F1 scores for both bonafide and spoof audio for all three models. The transformer methods perform much better: the AST and Wav2Vec models achieve an overall accuracy of 85% and 81%, respectively, on the FakeAVCeleb evaluation data. Wav2Vec is by far the most popular feature encoder in the literature, likely due to its generally superior performance, but here the AST-based transformer offers much better balanced performance on the out-of-distribution data.

7 DISCUSSION AND CONCLUSION

In this paper, we address the critical issue of audio deepfake detection by introducing a novel benchmark that evaluates the generalization capabilities of state-of-the-art transformer-based models. Our experiments, conducted using the ASVspoof 5 and FakeAVCeleb datasets, demonstrate that current detection solutions often struggle with generalizability and lack sufficient explainability, especially in real-world scenarios. By incorporating explainability methods such as attention roll-out and occlusion, we highlight the strengths and limitations of these approaches, providing a clearer understanding of model decisions.

The attention roll-out mechanism, in particular, shows promise in visualizing transformer-based models’ attention across multiple layers, enabling a more transparent analysis of how decisions are made. Our results indicate that while transformer models like Wav2Vec and AST outperform traditional models on unseen data, there remains a significant gap in their ability to provide human-understandable explanations. This finding underscores the need for more research in this area, especially to improve interpretability for non-technical users and domain experts.

Looking ahead, future work should focus on refining explainability methods for transformer-based models and developing new benchmarks that further challenge the robustness and interpretability of deepfake detectors. Additionally, understanding and mitigating the effect of data augmentations such as compression and re-recording will be essential for creating more resilient models. A limitation of this study is the reliance on only two datasets, which may not capture the full range of manipulations seen in real-world deepfake audio. Further limitations are addressed in Appendix F. Moreover, the proposed explainability methods are still in their infancy and may not yet offer insights that are as intuitive to non-technical users. By addressing these challenges, we can move closer to building reliable, trustworthy audio deepfake detection systems that are ready for real-world deployment.

540 REFERENCES
541

- 542 Sara Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the*
543 *58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, 2020.
- 544 Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial
545 video forgery detection network. In *2018 IEEE International Workshop on Information Forensics*
546 *and Security (WIFS)*, pp. 1–7. IEEE, 2018.
- 547 Alican Akman and Björn W. Schuller. Audio explainable artificial intelligence: A review. *Intelligent*
548 *Computing*, 3:0074, 2024. doi: 10.34133/icomputing.0074. URL <https://spj.science.org/doi/abs/10.34133/icomputing.0074>.
- 549 Sören Becker, Johanna Vielhaben, Marcel Ackermann, Klaus-Robert Müller, Sebastian Lapuschkin,
550 and Wojciech Samek. Audiominist: Exploring explainable artificial intelligence for audio
551 analysis on a simple benchmark. *Journal of the Franklin Institute*, 361(1):418–428, 2024.
552 ISSN 0016-0032. doi: <https://doi.org/10.1016/j.jfranklin.2023.11.038>. URL <https://www.sciencedirect.com/science/article/pii/S0016003223007536>.
- 553 Jordan J. Bird and Ahmad Lotfi. Real-time detection of ai-generated speech for deepfake voice
554 conversion, 2023. URL <https://arxiv.org/abs/2308.12734>.
- 555 Zexin Cai, Weiqing Wang, and Ming Li. Waveform boundary detection for partially spoofed audio,
556 2022. URL <https://arxiv.org/abs/2211.00226>.
- 557 Ting Chen, Zihang Luo, Zhiliang He, Hieu Huang, Honglak Lee, Wei-Ning Hsu, and Abdelrahman
558 Mohamed. Large speech supervision improves self-supervised speech representation learning.
559 *arXiv preprint arXiv:2203.01573*, 2022.
- 560 Heinrich Dinkel, Nanxin Chen, Yanmin Qian, and Kai Yu. End-to-end spoofing detection with
561 raw waveform cldnnns. In *2017 IEEE International Conference on Acoustics, Speech and Signal*
562 *Processing (ICASSP)*. IEEE, March 2017. doi: 10.1109/icassp.2017.7953080. URL <http://dx.doi.org/10.1109/ICASSP.2017.7953080>.
- 563 Abhishek Dixit, Nirmal Kaur, and Staffy Kingra. Review of audio deepfake detection techniques: Is-
564 sues and prospects. *Expert Systems*, 40(8):e13322, 2023. doi: <https://doi.org/10.1111/exsy.13322>.
565 URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13322>.
- 566 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas
567 Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit,
568 and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at
569 scale, 2021. URL <https://arxiv.org/abs/2010.11929>.
- 570 Adnan Dzanic, Sergey Seferbekov, Ohad Fried, and Jan Kautz. Detecting deepfakes without seeing
571 any: Face forgery detection via adversarially-filtered frequency domains. In *Proceedings of the*
572 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- 573 Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer, 2021. URL
574 <https://arxiv.org/abs/2104.01778>.
- 575 Jack Goodman. Trending - The Mexican mayor and a deepfake scandal - BBC Sounds — bbc.co.uk.
576 <https://www.bbc.co.uk/sounds/play/w3ct5d9g>, 2024. [Accessed 01-08-2024].
- 577 Xiaodan Gu, Majed Abuhadra, Anis Mohammed, and Oncel Tuzel. Self-supervised video forensics
578 by audio-visual anomaly detection. In *Proceedings of the IEEE/CVF International Conference*
579 *on Computer Vision (ICCV)*, 2023.
- 580 Alexandros Haliassos, Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Lips don't lie:
581 A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF*
582 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- 583 Hasam Khalid, Shahroz Tariq, Minha Kim, and Simon Woo. Fakeavceleb: A novel audio-video
584 multimodal deepfake dataset. In *Proceedings of the Neural Information Processing Systems Track*
585 *on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. URL <https://arxiv.org/abs/2108.05080>.

- 594 Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and
 595 Alexandr Kozlov. Stc antispooing systems for the asvspoof2019 challenge, 2019. URL <https://arxiv.org/abs/1904.05576>.
- 597
- 598 Tuan Duy Nguyen Le, Kah Kuan Teh, and Huy Dat Tran. Continuous learning of transformer-based
 599 audio deepfake detection, 2024. URL <https://arxiv.org/abs/2409.05924>.
- 600
- 601 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Con-*
 602 *ference on Learning Representations (ICLR)*, 2019. URL [https://openreview.net/](https://openreview.net/forum?id=Bkg6RiCqY7)
 603 [forum?id=Bkg6RiCqY7](https://openreview.net/forum?id=Bkg6RiCqY7).
- 604
- 605 Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proced-*
 606 *ings of the 31st international conference on neural information processing systems*, pp. 4768–
 607 4777, 2017.
- 608
- 609 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Pretten-
 610 hofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and
 611 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*,
 12:2825–2830, 2011.
- 612
- 613 Yuezun Qian, Guosheng Yin, Xinying Liu, Qifeng Shen, Siwei Li, Xin Yin, and Wei Lu. Explor-
 614 ing temporal coherence for more general video face forgery detection. In *Proceedings of the*
 615 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- 616
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the
 617 predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference*
 618 *on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.
- 619
- Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias
 620 Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the*
 621 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- 622
- Cynthia Rudin. Stop explaining black box machine learning models for high stakes de-
 623 cisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215,
 624 2019. doi: 10.1038/s42256-019-0048-x. URL <https://www.nature.com/articles/s42256-019-0048-x>.
- 625
- Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised
 626 pre-training for speech recognition, 2019. URL <https://arxiv.org/abs/1904.05862>.
- 627
- Yun Shi and Junichi Yamagishi. Investigating self-supervised front ends for speech spoofing coun-
 628 termeasures. *arXiv preprint arXiv:2111.07725*, 2021.
- 629
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image
 630 recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- 631
- CNN Staff. Election deepfake threats: How ai-manipulated content is raising concerns for 2024
 632 campaigns. *CNN*, February 2024. URL <https://edition.cnn.com/2024/02/01/politics/election-deepfake-threats-invs/index.html>. Accessed: 2024-10-01.
- 633
- Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee weon Jung, Junichi Yamagishi, and Nicholas
 634 Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and
 635 data augmentation, 2022. URL <https://arxiv.org/abs/2202.12233>.
- 636
- Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural net-
 637 works. In *International Conference on Machine Learning (ICML)*, pp. 6105–6114. PMLR, 2019.
- 638
- Enkhtogtokh Togootogtokh and Christian Klæsen. Antideepfake: Ai for deep fake speech recog-
 639 nition. *arXiv preprint arXiv:2402.10218*, 2024. URL <https://arxiv.org/abs/2402.10218>.

- 648 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,
 649 Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL <https://arxiv.org/abs/1706.03762>.
- 650
- 651 Xin Wang and Junichi Yamagishi. Investigating self-supervised front ends for speech spoofing
 652 countermeasures, 2022. URL <https://arxiv.org/abs/2111.07725>.
- 653
- 654 Xin Wang, Héctor Delgado, Hemlata Tak, Jee weon Jung, Hye jin Shim, Massimiliano Todisco, Ivan
 655 Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and
 656 Junichi Yamagishi. Asvspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at
 657 scale. In *ASVspoof 5 Workshop (Interspeech 2024 Satellite)*, pp. 1–8, 2024. doi: 10.48550/arXiv.
 658 2408.08739. URL <https://arxiv.org/abs/2408.08739>.
- 659
- 660 Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and im-
 661 proving relative position encoding for vision transformer, 2021. URL <https://arxiv.org/abs/2107.14222>.
- 662
- 663 Zhenzong Wu, Rohan Kumar Das, Jichen Yang, and Haizhou Li. Light convolutional neural network
 664 with feature genuinization for detection of synthetic speech attacks, 2020. URL <https://arxiv.org/abs/2009.09637>.
- 665
- 666 Qin Xie, Yisong Zhang, Ming Tu, Zhiyuan Tang, and Dong Yu. Real-time conformer for streaming
 667 speech recognition. In *Proc. Interspeech 2021*, pp. 2257–2261, 2021.
- 668
- 669 Zeyu Xie, Baihan Li, Xuenan Xu, Zheng Liang, Kai Yu, and Mengyue Wu. Fakesound: Deepfake
 670 general audio detection, 2024. URL <https://arxiv.org/abs/2406.08052>.
- 671
- 672 Anna K. Yanchenko, Mohammadreza Soltani, Robert J. Ravier, Sayan Mukherjee, and Vahid
 673 Tarokh. A methodology for exploring deep convolutional features in relation to hand-crafted
 674 features with an application to music audio modeling, 2021. URL <https://arxiv.org/abs/2106.00110>.
- 675
- 676 Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. Audio
 677 deepfake detection: A survey. *arXiv preprint arXiv:2308.14970*, 2023.
- 678
- 679 Hossein Zeinali, Themos Stafylakis, Georgia Athanasopoulou, Johan Rohdin, Ioannis Gkinis, Lukáš
 680 Burget, and Jan "Honza" Černocký. Detecting spoofing attacks using vgg and sincnet: But-omilia
 681 submission to asvspoof 2019 challenge, 2019. URL <https://arxiv.org/abs/1907.12908>.
- 682
- 683 You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing de-
 684 tection. *IEEE Signal Processing Letters*, 28:937–941, 2021. ISSN 1558-2361. doi: 10.1109/LSP.
 685 2021.3076358. URL <http://dx.doi.org/10.1109/LSP.2021.3076358>.
- 686
- 687 Shangzhe Zhao, Zhaoyang Cheng, and Chen Change Loy. Realforensics: Leveraging real talking
 688 faces via self-supervision for robust forgery detection. In *European Conference on Computer
 689 Vision (ECCV)*, 2022.
- 690
- 691
- 692
- 693
- 694
- 695
- 696
- 697
- 698
- 699
- 700
- 701

702 **A RELATED WORK (CONTINUED)**

703

704 **A.1 CNN-BASED CLASSIFIERS**

705

706 The most popular architecture for deepfake audio detection is by far the convolutional neural net-
 707 work (CNN) based classifier Dixit et al. (2023); Wu et al. (2020); Lavrentyeva et al. (2019); Zeinali
 708 et al. (2019); Dinkel et al. (2017). CNN classifiers have been historically favored for their ability
 709 to combine spatial and temporal information through convolution. The most successful of these is
 710 called *Light Convolutional Neural Network* (LCNN) and was proposed by Wu et al. in 2020 Wu et al.
 711 (2020). LCNN consists of convolutional and max-pooling layers with Max-Feature-Max (MFM)
 712 activation. Wu et al. choose the MFM activation function instead of the arguably standard ReLU
 713 function because they observe that the MFM learns more compact features than the sparse high-
 714 dimensional ones learned with ReLU Wu et al. (2020). This distinction is what makes their CNN
 715 “light”. The LCNN has been incredibly successful in recent ASVspoof and ADD competitions;
 716 it was the best system at ASVspoof 2017 and continued to be the best system in one of the sub-
 717 tasks of ASVspoof 2019. Another successful CNN architecture, proposed by Dinkel et al., uses
 718 raw waveforms as input to a convolutional long short-term neural network. Their model combines
 719 time-convolving layers with frequency-convolving layers to “reduce time and spectral variations”
 720 with long-term temporal memory layers to capture longer-term temporal relationships Dixit et al.
 721 (2023); Dinkel et al. (2017). A variety of other CNN architectures have been proposed, but all of
 722 them generalize poorly to unseen attacks Dixit et al. (2023).

723 **A.2 DEEPFAKE AUDIO IN-PAINTING**

724

725 Another relevant research area is that of deepfake audio in-painting detection. In-painting refers
 726 to the practice of mixing real and fake audio such that only small portions of an audio sample
 727 are actually manipulated. Detecting deepfake audio in-painting requires not only identifying that a
 728 sample contains some corrupted audio but also identifying the timestamps at which the corruption
 729 begins and ends. Xie et al. (2024) propose a framework called EAT, which incorporates a ResNet,
 730 a two-layer transformer encoder, a single-layer bidirectional Long Short-Term Memory network
 731 (LSTM), and a final classification layer.

732 The EAT framework achieves an F1 score of 98% when classifying segments at 20 millisecond
 733 resolution Xie et al. (2024). However, Xie et al. only evaluate their method on a custom dataset,
 734 do not attempt to evaluate their architecture’s performance on any of the standard deepfake audio
 735 benchmarks, and focus primarily on deepfake environmental sounds and background audio Xie
 736 et al. (2024). A slightly older but broader work, from 2022, that investigates in-painted deepfake
 737 audio is that of Cai et al. (2022) in *Waveform Boundary Detection for Partially Spoofed Audio*.
 738 Cai et al. (2022) use a combination of Wav2Vec and MFCC features as their input to a series of
 739 single-dimensional CNNs, a transformer encoder, a bidirectional long short-term memory network
 740 (BiLSTM), and a final linear layer for classification (Cai et al., 2022). With their method, they
 741 achieve the best performance in the locating manipulated clips task of the ADD 2022 challenge.

742 **B DATA SOURCES**

743

744 We employ a variety of publicly available datasets to conduct our experiments. We also craft custom
 745 variations in order to better mimic real-world use cases and challenges.

746 The majority of our experiments are conducted with the ASVspoof 5 dataset. The ASVspoof5
 747 dataset is a state-of-the-art dataset containing eighteen different varieties of deepfake audio as well as
 748 true speech samples (Wang et al., 2024). Each sample is labelled with a classification as “bonafide”
 749 or “spoof”. If the sample is spoofed, the attack method that was used to generate the sample is spec-
 750 ified. The dataset, which was released in June 2024, contains 182 357 train samples and 142 134 test
 751 samples. Each deepfake (or spoofed) audio sample is generated with a novel VC or TTS method,
 752 which were trained on two English-language datasets. The final deepfake samples are made using
 753 the English-subset of the Multilingual LibriSpeech (MLS) dataset (Wang et al., 2024). The 18 dif-
 754 ferent attack types included in this dataset make it the most attack diverse of the datasets we consider
 755 in this study. We take this dataset to represent the state-of-the-art in deepfake audio generation and,
 756 in subsequent sections, refer to it simply as the ASVspoof dataset.

The audio samples provided by ASVspoof are clean. The ASVspoof bonafide samples were created in recording studios with high quality microphones; The ASVspoof fake samples were generated with TTS algorithms that add no additional noise. When deepfake audio is circulated over the social media, it undergoes compression—often multiple times. In order to identify fake audio downstream, our models must be robust to this kind of distribution modification. Additionally, bad actors will try to obscure as much as possible that an audio sample has been faked. A common approach to obscure fake audio is to play to audio aloud in a room and re-record the audio before disseminating it. We create two additional datasets from the ASVspoof dataset for an additional challenge.

Compressed ASVspoof To tackle the issue of compression, we write all of the audio samples in the ASVspoof dataset to the lossy MP3 format (from the lossless FLAC format) with a bitrate of 128k. This shrinks each file, on average, to 33.7% of its original size.

Rerecorded ASVspoof To tackle the issue of re-recording, we re-recorded audio samples in the ASVspoof dataset by playing them aloud on a 2021 MacBook Pro in a large, closed stone-walled room while simultaneously recording.

We also refer to FakeAVCeleb as a well-known benchmark in deepfake audio detection. The FakeAVCeleb dataset is a standard in the deepfake audio detection repertoire, but it is now slightly out-of-date, as it was released in 2021 (Khalid et al., 2021). It contains both deepfake audio and video, but we use only the audio component in this study. The audio subset of the FakeAVCeleb dataset contains 9 712 real audio samples and 10 843 deepfake audio samples. The language of each audio sample is English, but FakeAVCeleb includes balanced classes of male and female speakers as well as speakers who self-identify as African, East Asian, South Asian, Caucasian (American), and Caucasian (European) (Khalid et al., 2021). This makes FakeAVCeleb the most linguistically diverse of the datasets considered in this study.

C HYPERPARAMETERS

Both AST and Wav2Vec transformer models use their own specialized encodings. Otherwise, they are finetuned very similarly. Table 3 reports the hyperparameters used when finetuning the Wav2Vec and AST models, as well as the experiments performed with each model and dataset combination.

C.1 GBDT

Given its recent popularity for deepfake audio classification tasks Bird & Lotfi (2023); Togootogtokh & Klasen (2024), we use the gradient boosting classifier as the baseline in this study. The features for these tests are the hand-crafted, signal processing features described in Section 3. We use a total of 37 features, which include 20 MFCC features, 12 chroma features, spectral bandwidth, spectral roll-off, spectral centroid, ZCR, and RMS. The features for each audio sample are calculated by averaging each feature’s values across all frames. For example, the final ZCR is calculated by averaging the ZCR for each frame across the length of the entire audio sample.

To evaluate best possible performance with the GBDT, we do a hyperparameter search across maximum tree depths and maximum number of estimators. We consider maximum tree depth in $\{3, 8, 10, 15, 25\}$ and number of estimators in $\{10, 100, 200, 400, 600\}$. We do our hyperparameter search with the ASVspoof dataset. We conduct tests with 10 000 total data samples, 33% of which are held out as a test set. As GBDTs are highly sensitive to unbalanced data classes, we balance classes before conducting experiments with the GBDT.

As shown in Table 2, the classification accuracy is highest with shorter decision trees and more estimators when trained with 3-second audio samples. The performance stabilizes with 400 or more estimators, so we take 400 estimators and maximum depth of 8 as our optimal hyperparameter setting. We find that this configuration also offers best performance for other audio sample lengths. We observe that performance improves with the number of estimators, but we also see diminishing returns after 400 estimators. We conclude that 400 estimators and a maximum tree depth of 8 are the optimal hyperparameters for this training scenario.

We compute feature importances with the following algorithm (Pedregosa et al., 2011):

810
811 Table 2: Accuracy of gradient boosting classifier for various maximum tree depths and number of
estimators.

Max Depth	Number of Estimators					
	10	100	200	400	600	1000
3	0.748	0.822	0.838	0.850	0.856	0.859
8	0.792	0.840	0.854	0.860	0.859	0.862
10	0.787	0.836	0.845	0.848	0.848	0.848
15	0.745	0.791	0.814	0.814	0.814	0.814
25	0.725	0.732	0.736	0.736	0.736	0.736

Algorithm 1 Permutation Importance Algorithm

```

Input: fitted model  $m$ , dataset  $D$ , metric  $a$ , repeats  $R$ 
 $s \leftarrow a(m, D)$ 
for each feature  $j$  in  $D$  do
  for repeat in  $1 \dots R$  do
    Randomly shuffle column  $j$  in  $D$  to create corrupted  $\hat{D}_{r,j}$ 
     $s_{r,j} \leftarrow a(m, \hat{D}_{r,j})$ 
  end for
   $i_j = s - \frac{1}{R} \sum_{r=1}^R s_{r,j}$ 
end for
  
```

834 C.2 TRANSFORMERS

835
836 Both AST and Wav2Vec transformer models use their own specialized encodings. Otherwise, they
837 are finetuned very similarly. Table 3 reports the hyperparameters used when finetuning the Wav2Vec
838 and AST models, as well as the experiments performed with each model and dataset combination.
839

840 D MODEL PERFORMANCE ON ASVspoof 5 AND FAKEAVCELEB

841
842 We evaluate the performance of the gradient boosting classifier on compressed audio samples with
843 the optimal hyperparameters of 400 estimators and maximum tree depth of 8. Figure 6 shows that
844 performance is significantly worsened by both compression and rerecording. The difference in
845 performance is particularly stark on short audio samples, where the gradient boosting classifier
846 excels with the original, unmodified ASVspoof data.

847 Here, we demonstrate the superiority of the transformer-based methods we consider on the
848 FakeAVCeleb dataset.

849
850 Table 3: Hyperparameters used in the finetuning of Wav2Vec and AST transformers.

Hyperparameter	Value
Batch Size	32
Learning Rate	3×10^{-5}
Training Steps	40
Train-Test Split	0.33
Weight Decay	0.0
Warm Up Ratio	0.1
Optimizer	AdamW Loshchilov & Hutter (2019)

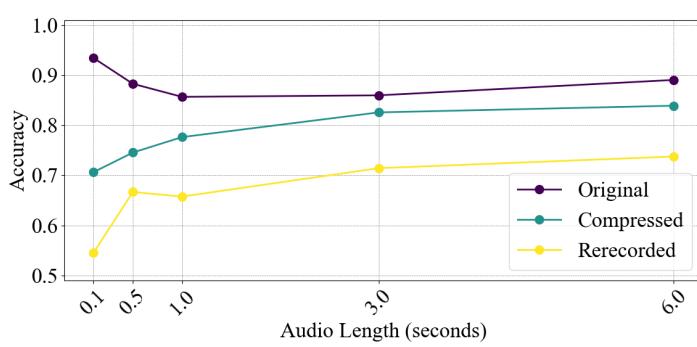


Figure 6: Accuracy over various audio lengths for the original, compressed, and rerecorded audio data for booster graphs with 10 000 datapoints, max depth of 8, and 400 estimators.

Table 4: Accuracy and ROC AUC comparison of models on FakeAVCeleb.

Model	Accuracy	ROC AUC
EfficientNet-B0 (Tan & Le, 2019)	0.500	-
MesoInception-4(Afchar et al., 2018)	0.540	-
VGG16 (Simonyan & Zisserman, 2014)	0.671	-
Xception (Rossler et al., 2019)	0.763 ¹	0.853
AD DFD	-	0.881
LipForensics (Haliassos et al., 2020)	-	0.911
FTCN (Qian et al., 2021)	-	0.931
AVAD (Gu et al., 2023)	-	0.945
RealForensics (Zhao et al., 2022)	-	0.971
FACTOR (Dzanic et al., 2023)	-	0.974
AST (ours)	0.979	0.985
Wav2Vec (ours)	0.991	0.990

D.1 DATA AUGMENTATION

Though the ASVspoof 5 dataset includes compressed and rerecorded audio samples, they are not marked or isolated within the dataset. To evaluate the impact on performance that these augmentations have, we create three distinct datasets: one with the original ASVspoof 5 data, one with all data compressed, and one with all data rerecorded. We evaluate the effect that these augmentations have on model performance.

¹Unimodal (audio-only) result.

918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937

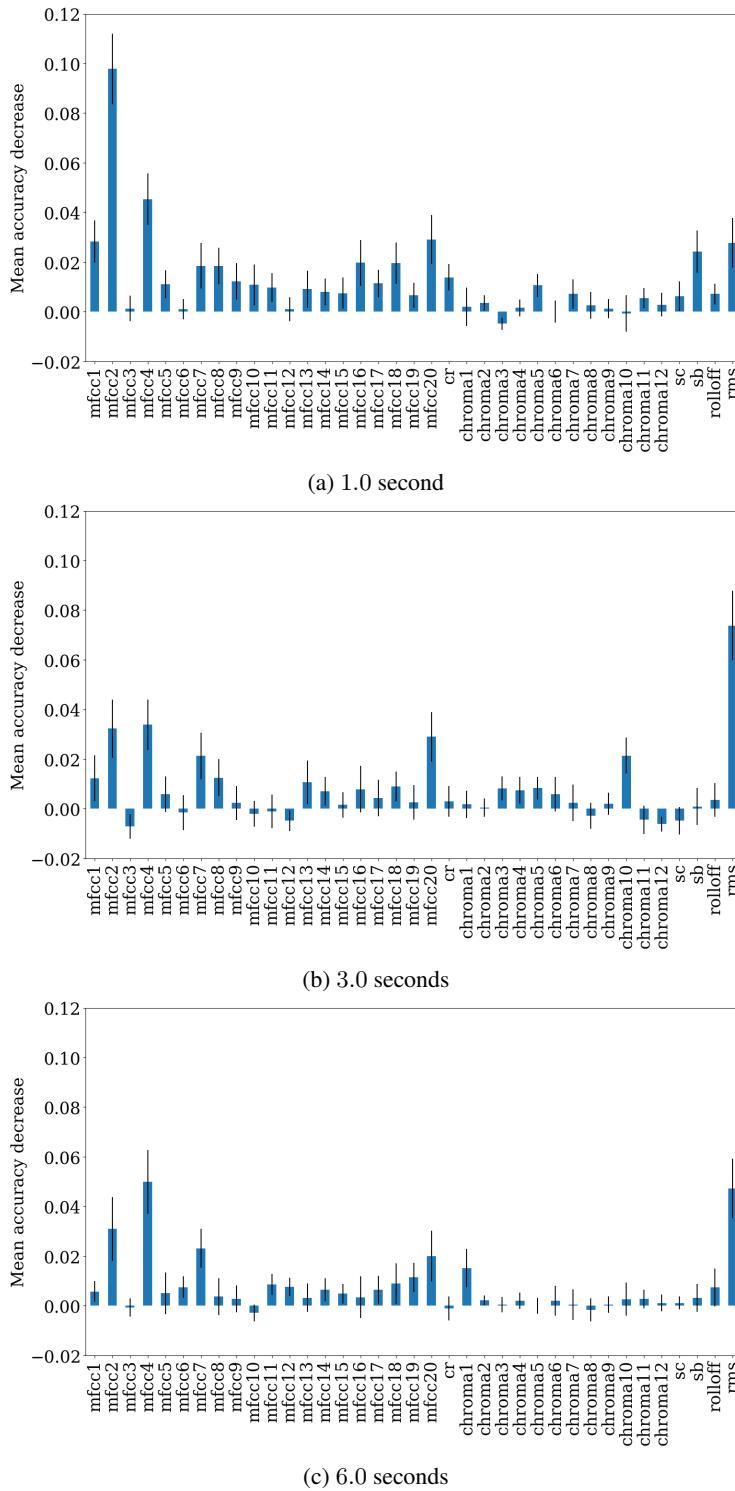
938 Table 5: Comparing the precision, accuracy, recall, and F1 of the models GBDT, AST, and Wav2Vec
 939 for 6.0-second samples of original, compressed, and rerecorded ASVspoof data.

augmentation	model	precision	accuracy	recall	F1
original	GBDT	0.903	0.896	0.891	0.894
	AST	0.981	0.992	0.987	0.984
	Wav2Vec	0.993	0.998	1.000	0.997
compressed	GBDT	0.853	0.841	0.837	0.841
	AST	0.994	0.994	0.982	0.988
	Wav2Vec	0.995	0.998	1.000	0.997
rerecorded	GBDT	0.589	0.589	0.589	0.589
	AST	0.968	0.991	0.994	0.981
	Wav2Vec	0.998	0.995	0.996	0.997

952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971

972 E EXPLAINABILITY 973

974 E.1 GBDT FEATURE IMPORTANCES 975



1024 Figure 7: GBDT feature importances as measured by mean accuracy decrease with standard deviations
1025 for the 1.0, 3.0, and 6.0-second classifiers.

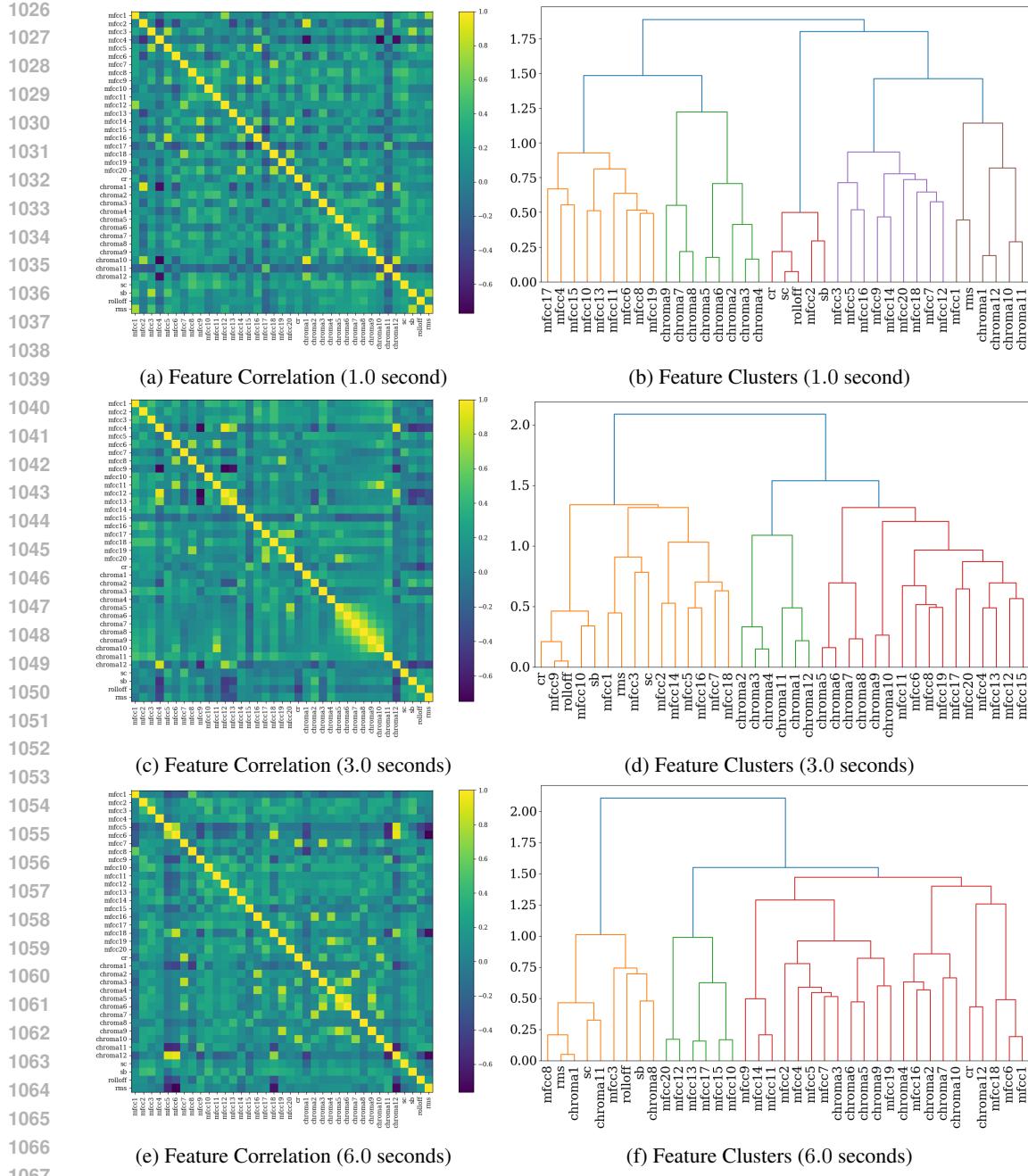


Figure 8: GBDT feature correlations and clusters for the 1.0, 3.0, and 6.0-second classifiers. In these figures, *sc* refers to the spectral centroid, *sb* refers to the spectral bandwidth, *cr* refers to the ZCR, *mfcc_i* refers to the *i*-th MFCC feature, and *chroma_i* refers to the *i*-th chroma feature.

1080
1081

E.2 LAYER-WISE ATTENTION

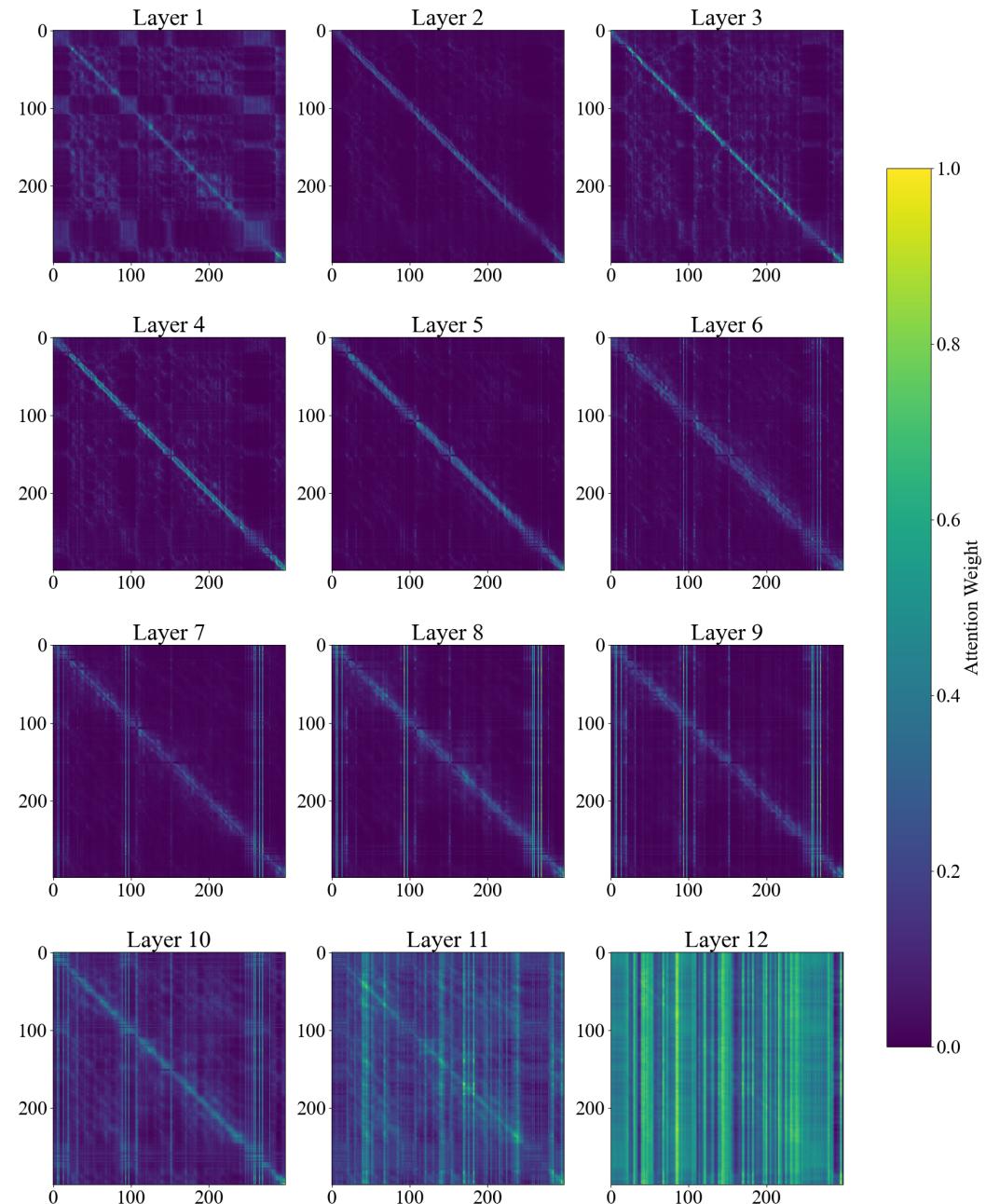
1082
1083
1084
1085
1086
1087
1088
1089
1090
10911123
1124
1125

Figure 9: Normalized attention visualized for a bonafide 6.0-second sample where axes represent input token ID.

1126
1127
1128
1129
1130
1131
1132
1133

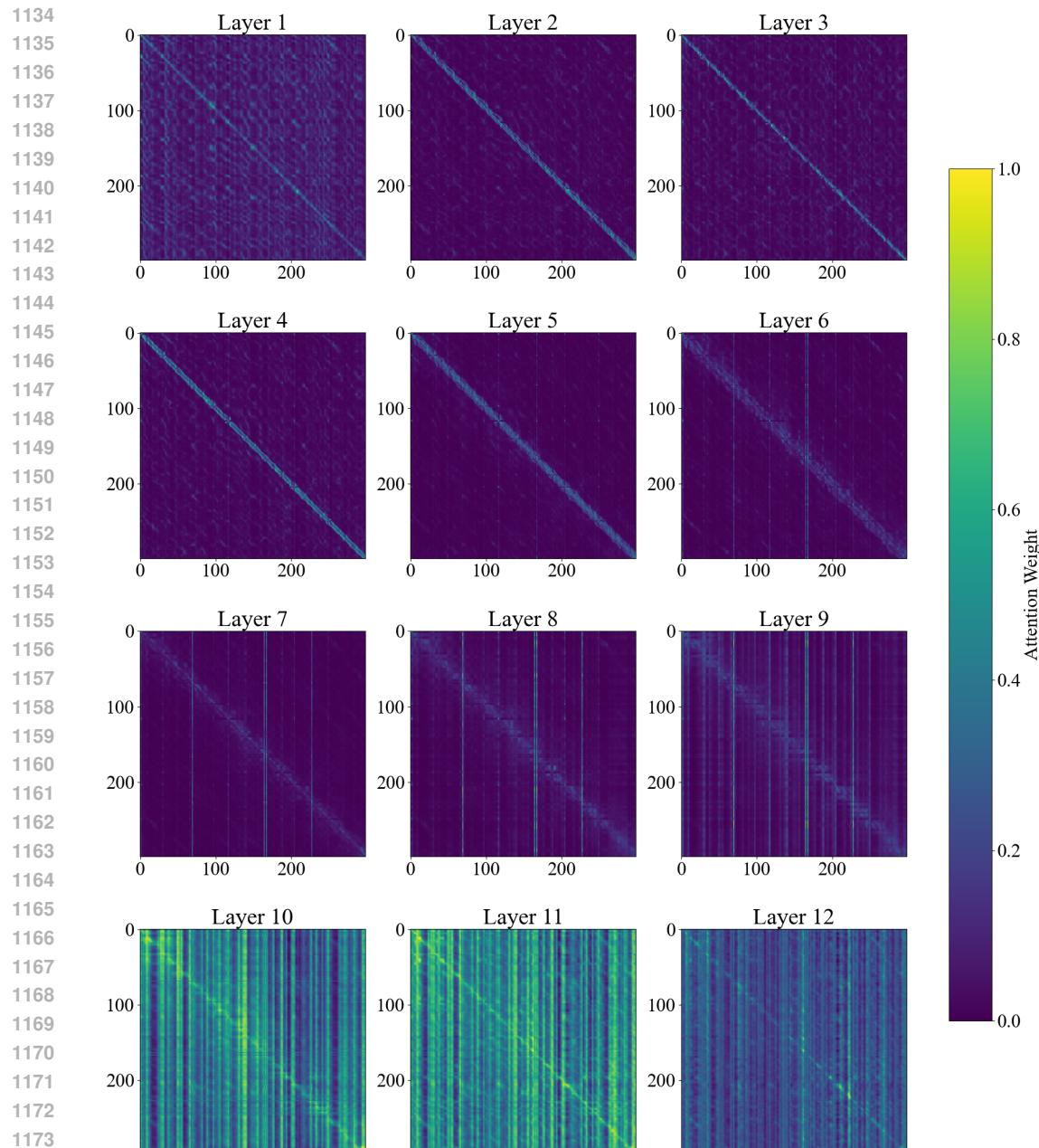


Figure 10: Normalized attention visualized for a spoof 6.0-second sample where axes represent input token ID.

1188
1189

E.3 ATTACK CLASSIFICATION

1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201

Another potentially useful piece of information is what attack likely generated a given deepfake audio segment. Though this type of “explainability” does not give insight into a model’s decision-making, it may provide a clue to the origin of a deepfake audio sample—something likely very important to a journalist or member of law enforcement. We therefore evaluate the abilities of the GBDT and transformer methods to identify which attack, if any, was used to generate an audio sample. Using 6.0-second audio samples, the GBDT achieved an overall accuracy of 51.9% with average precision of 53.0% and average recall of 52.1% over all 18 classes, with relatively stable performance across the attack types. The Wav2Vec model achieved overall evaluation accuracy of 91.8% with average precision of 91.7% and average recall of 91.8% across the 18 classes. The AST model performed similarly, achieving overall evaluation accuracy of 91.1%, average precision of 91.1%, as well as average recall of 91.1%. Though the models are quite successful at this task, previous exposure to deepfake audio generated with each attack is essential. It is not clear how well a model could detect an out-of-distribution sample.

1202
1203
1204

F LIMITATIONS

1205
1206
1207
1208
1209
1210
1211
1212
1213

While this study introduces a novel benchmark for evaluating the generalization capabilities of state-of-the-art transformer-based models, there are several limitations that need to be addressed to advance the field of audio deepfake detection. First, the reliance on only two datasets, ASVspoof 5 and FakeAVCeleb, limits the scope of our evaluation. These datasets, while diverse, may not encompass the full range of manipulations, recording conditions, and spoofing techniques found in real-world scenarios. This constraint may lead to an overestimation of model robustness and underrepresentation of other spoofing methods. Expanding the benchmark to include more diverse datasets with a wider variety of deepfake techniques would provide a more comprehensive evaluation of model performance and generalization.

1214
1215
1216
1217
1218
1219
1220

Second, the explainability methods employed, such as attention roll-out and occlusion, are still limited in their ability to provide intuitive and human-understandable explanations, particularly for non-technical audiences. While these methods help visualize and highlight the regions of the input that influence the model’s decisions, they do not yet offer a complete picture of why a model might fail on specific instances or how it generalizes across domains. Moreover, the explainability results may be sensitive to changes in model architecture and hyperparameters, which could lead to inconsistencies in interpretation.

1221
1222
1223
1224
1225
1226

Lastly, while we explore the generalization capabilities of the proposed models using cross-dataset evaluation, the impact of various environmental factors such as background noise, speaker variability, and language differences were not explicitly analyzed. This could affect the real-world applicability of the models, especially in diverse and dynamic environments. Further research should focus on robustness testing under varying conditions and on creating synthetic data to simulate potential challenges encountered during real-world deployment.

1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241