Defence Against the Deepfake Arts : Improving Audio Deepfake Detection With Context Awareness

Anonymous submission to Interspeech 2025

Abstract

Abhay : 1000 character limit; 848 currently.

The increasing use of generative AI models to create realistic deepfakes poses significant challenges to information security, particularly in the realm of audio manipulation. This paper addresses the pressing need for improved detection of audio deepfakes by proposing a novel approach that leverages textual context extracted from transcriptions as well as audio features extracted from deepfake detection models. We propose two fusion strategies, late fusion and mid fusion, to integrate these features and enhance detection accuracy. We conduct experiments using benchmark datasets and state-of-the-art deepfake detection models to evaluate the performance of our proposed approach. Our results demonstrate promising improvements in detecting audio deepfakes, highlighting the potential of context awareness in enhancing detection capabilities.

Index Terms: audio deepfake detection

1. Introduction

With the rapid advances in the digital landscape and new media technologies, users become increasingly exposed to different types of manipulations. Nowadays, everyone can create highly realistic AI-generated content such as images, audio, and even videos. Distinguishing between those and the real ones becomes nearly impossible for the ordinary user. Audio deepfakes or voice clones are like digital ventriloquists, using various deep learning techniques to create voices that convincingly impersonate real people [1]. We can distinguish between a couple of different types of deepfakes. [2]. The first approach is imitation-based (voice conversion), wherein the original audio signal is modified to mimic another targeted voice. This technique is mainly used in show business and, therefore, is applied by a third person or deep learning software. The next approach is synthetic-based, producing artificially generated speech using text-to-speech techniques (TTS). This process involves three stages: text analysis, acoustic, and vocoder. The third technique is centred on replaying existing recordings of the target speaker. Two techniques are notable here: cut-and-paste detection, where a victim's recorded segment is played through a phone handset for capture, and far-field detection, which involves crafting sentences for manipulation of a text-dependent system. There are other types of deepfakes, such as emotion fake. This technique alters the emotions of a speaker's voice while preserving the original words and speaker identity. [3]. Another type is scene fake, which involve modifying the background sounds of a recording, leaving the speaker and content untouched. [4]. The last type of deepfake is the partially fake, which involves replacing specific words within the original audio signal with either real or synthetic audio segments. The speaker remains the same throughout, but the content is altered [5].

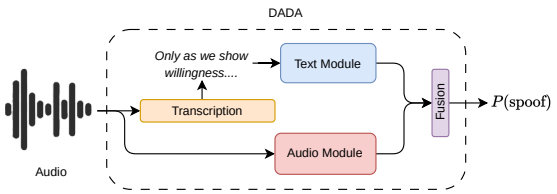

Figure 1: *Overview of the DADA pipeline. Audio files are first transcribed to obtain textual content. Features are then extracted from both the audio and text using separate models. Finally, these features are combined (using either mid fusion or late fusion strategies) to generate the final prediction.*

All these different techniques can have far-reaching consequences, affecting domains ranging from political discourse to personal security. These deepfakes are not only used in creating political propaganda to manipulate public opinion [6] but have also become tools for conducting phone scams [7], posing significant threats to individual privacy and security. The ability to convincingly replicate or alter voices using AI have affirmed the urgent need for robust detection mechanisms. This paper contributes to proactive measures in detecting audio deepfakes, particularly focusing on detecting audio deepfakes of public figures. Public figures are especially susceptible to the detrimental effects of audio deepfakes as their extensive online presence renders them prime targets for malicious actors seeking to disseminate misinformation.

We propose a new pipeline that aims to not only analyze the audio utterances but also incorporate contextual information. Furthermore, DADA has two different branches, one responsible for detecting if the audio is real or fake and a text branch performing the task of Authorship Attribution on the transcription of the audio. We want to answer the question of what is the probability of the anticipated speaker having made the given statement.

Recent advancements in authorship attribution using deep learning have showcased the efficacy of various encoder-based architectures in capturing linguistic and stylistic patterns unique to individual authors. We can encounter multiple papers that explore BERT or T5-based models for identifying authorship in literary texts, achieving notable accuracy [8, 9]. For instance, BertAA [10] is a fine-tuned pretrained BERT model for authorship attribution, demonstrating competitive performance on multiple datasets. We can also find several works that use tech-

niques from contrastive learning in creating discriminative embedding spaces. Works like DeepStyle, [11] focusing on short-text authorship attribution, leveraging Triplet Loss, which outperformed existing baselines using Twitter and Weibo datasets and using Siamese networks for large-scale author identification [12]. [13] proposed a transformer-based model using arXiv manuscripts, achieving high attribution accuracy in a large-scale dataset with up to 2,000 authors.

2. Related Work

Keeping all related work in Introduction 1 for now, in line with papers from Interspeech24.

3. Method

In this section, we present the DADA architecture for audio deepfake detection that improves the traditional methods analyzing only the utterances of the audio files, by incorporating contextual information to improve detection capabilities. Specifically, our approach seeks to determine not only whether an audio sample is a deepfake but also the likelihood of the supposed speaker having made the given statement, thus embedding contextual awareness into the detection pipeline.

Our method operates in three stages: transcription, feature extraction and fusion. First, the audio files are transcribed using Whisper [14] to obtain the textual content. Following transcription, the architecture leverages two separate models trained independently a text and an audio model. The extracted audio and text feature are then combined using two distinct fusion strategies to obtain the final classification score.

3.1. Text Model

The main idea of the text model is to fine-tune a pretrained encoder-based LLM for authorship attribution. To improve the model's ability to differentiate between stylistic and linguistic patterns unique to specific author, we extracted the features from the last three hidden states of the encoder, instead of only the CLS token. Transformer models encode information hierarchically across layers. The lower layers capture phrase-level or segment-level interactions, where the intermediate layers represent syntactic information like grammar and vocabulary and the higher layers encode task-specific or abstract semantic information. By using multiple layers we want capture a broader spectrum of features, combining syntactic, semantic, and task-specific information [15]. The extracted features are aggregated using a Mean Pooling layer, which computes the mean while accounting for the attention mask. This vector is passed to task-specific layers, such as a several fully connected layers. To model the space of authors effectively and identify the characteristics of their speech sets, we employ techniques from contrastive learning such as Triplet loss. This loss function helps create a well-defined embedding space, where texts attributed to the same author are closer together, while those from different authors are further apart [16]. The function is defined using triplets (A,P,N), where A is a randomly sampled anchor (reference point), P is a positive sample that has the same label as A and N is the negative sample of a different random class than the anchor. The goal is to ensure that the distance between the anchor and the positive is smaller than the distance between the anchor and the negative by at least a certain margin m. We define the function as follows:

Abhay: Aliter

We define the loss on features f_θ^A, f_θ^P and f_θ^N generated by the model on a triplet as follows:

changelog[Abhay]: added label, modified notation to reduce width

$$\mathcal{L} = [d(f_\theta^A, f_\theta^P) - d(f_\theta^A, f_\theta^N) + \lambda]_+ \quad (1)$$

Where $f_\theta(x)$ is the embedding of the input x generated by the model, λ is the margin that enforces a minimum separation between similar and dissimilar pairs, $d(x, y)$ is the distance function between the two embeddings, and $[\cdot]_+ = \max(\cdot, 0)$. We experimented with two different distance functions, the squared Euclidean distance (L2-norm) and the cosine similarity between the features.

3.2. Audio Model

3.3. Fusion Strategies

After independently training the audio and text models, we freeze their parameters and combine their features using one of two fusion strategies: mid fusion or late fusion.

3.3.1. Mid Fusion

In this strategy, the extracted features from the audio and text models are concatenated to create a joint representation. This combined representation is then passed through a Multi-Layer Perceptron (MLP) network. The MLP learns to integrate the audio and text features effectively to make a final prediction. The training of the MLP is guided by the Binary Cross-Entropy (BCE) loss function, which is appropriate for binary classification tasks such as distinguishing between real and spoof audio samples. With this approach we hope that one of the branches will compensate for the mistakes of the other and hopefully increase the performance of the detector.

3.3.2. Late Fusion

In this approach, the audio and text models are treated as independent branches, and each produces its own prediction score. The final prediction is obtained by taking the weighted mean of the scores from the two branches. This simple averaging method ensures that the final decision benefits from the independent contributions of both modalities while maintaining the interpretability of individual scores.

!!! Abhay: Need to discuss this

We model late fusion for two possible scenarios: one where there exists a purported speaker (consistent with deepfakes claiming to be a specific individual/celebrity) and one where the speaker is unknown. In the former case, the text model outputs the probability of the anticipated speaker having made the given statement, while the audio model predicts the likelihood of the audio being real or fake. In the latter case, the text model predicts the probability of the speaker being any of the known authors, while the audio model predicts the likelihood of the audio being real or fake. The final prediction is obtained by taking the weighted mean of the scores from the two branches. This simple averaging method ensures that the final decision benefits from the independent contributions of both modalities while maintaining the interpretability of individual scores.

Formally,

Both fusion strategies provide distinct advantages: mid fusion enables the model to learn deeper interactions between the modalities, while late fusion offers simplicity and preserves the autonomy of each model.

4. Experiments and Results

Abhay: Need to decide whether we group by task{text, audio, fusion} or by {datasets,metrics,baselines}. Further, should we rename the text model the context module?

4.1. Text / Context Module

4.1.1. Dataset

For the authorship attribution task we incorporate an additional dataset Wikiquotes, which provides a rich corpus of speeches from various authors. Since only real samples are needed for training the text model, relying only on the transcriptions provided by Whisper proved insufficient for learning the nuanced characteristics of individual authors.

4.1.2. Metrics

ABX accuracy is a commonly used metric in speech and speaker representation learning, often used in tasks related to contrastive learning. Similar to the triplet loss we have three samples A,B and X. Where A and B are two embeddings from different categories and X is a third representation, and the task is to determine whether X is more similar to A or B based on the learned embeddings. The accuracy measures the percentage of times the model correctly assigns X to its corresponding label (closer to A or B). The decision rule is: $d(X, A) < d(X, B)$, where $d(x, y)$ is the distance function.

4.1.3. Benchmark

We consider several state-of-the-art encoder-only architectures to benchmark the effectiveness of the text model, including BERT, RoBERTa, DeBERTa, Modern BERT, and the encoder of T5. Encoder-only models are particularly well-suited for the task of authorship attribution because they focus exclusively on encoding the input sequence into a fixed-size representation without introducing additional complexities of the sequence-to-sequence transformations. This focus allows them to better capture nuanced relationships in the text, such as stylistic patterns and semantic coherence. We wanted to identify the best architecture by comparing these models for capturing each author's writing style and improving our detection system's overall performance.

5. Conclusions / Discussion

6. References

[1] P. Kawa, M. Plata, and P. Syga, "Defense against adversarial attacks on audio deepfake detection."

[2] A. Dixit, N. Kaur, and S. Kingra, "Review of audio deepfake detection techniques: Issues and prospects," *Expert Systems*, vol. 40, 04 2023.

[3] Y. Zhao, J. Yi, J. Tao, C. Wang, X. Zhang, and Y. Dong, "Emofake: An initial dataset for emotion fake audio detection," 2023.

[4] J. Yi, C. Wang, J. Tao, Z. Tian, C. Fan, H. Ma, and R. Fu, "Scenefake: An initial dataset and benchmarks for scene fake audio detection," 2022.

[5] J. Yi, Y. Bai, J. Tao, H. Ma, Z. Tian, C. Wang, T. Wang, and R. Fu, "Half-truth: A partially fake audio detection dataset," 2023.

[6] G. STĂNESCU, "Informational war: Analyzing false news in the israel conflict."

[7] Y. Mirsky, "Df-captcha: A deepfake captcha for preventing fake calls," 2022.

[8] R. M. Hicke and D. Mimno, "T5 meets tybalt: Author attribution in early modern english drama using large language models," *arXiv preprint arXiv:2310.18454*, 2023.

[9] K. Silva, I. Frommholz, B. Can, F. Blain, R. Sarwar, and L. Ugolini, "Forged-gan-bert: Authorship attribution for llm-generated forged novels," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 2024, pp. 325–337.

[10] M. Fabien, E. Villatoro-Tello, P. Motlicek, and S. Parida, "Bertaa: Bert fine-tuning for authorship attribution," in *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, 2020, pp. 127–137.

[11] Z. Hu, R. K.-W. Lee, L. Wang, E.-p. Lim, and B. Dai, "Deepstyle: User style embedding for authorship attribution of short texts," in *Web and Big Data: 4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China, September 18-20, 2020, Proceedings, Part II 4*. Springer, 2020, pp. 221–229.

[12] C. Saedi and M. Dras, "Siamese networks for large-scale author identification," *Computer Speech & Language*, vol. 70, p. 101241, 2021.

[13] L. Bauersfeld, A. Romero, M. Muglikar, and D. Scaramuzza, "Cracking double-blind review: Authorship attribution with deep learning," *Plos one*, vol. 18, no. 6, p. e0287611, 2023.

[14] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[15] G. Jawahar, B. Sagot, and D. Seddah, "What does bert learn about the structure of language?" in *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[16] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," *Advances in neural information processing systems*, vol. 32, 2019.

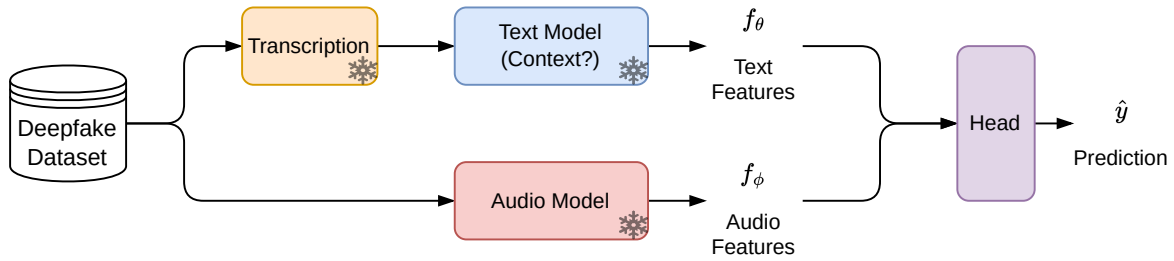

Figure 2: *Pipeline for Mid-Fusion. Features from the text and audio modules are concatenated and passed through an MLP to produce the final classification.*

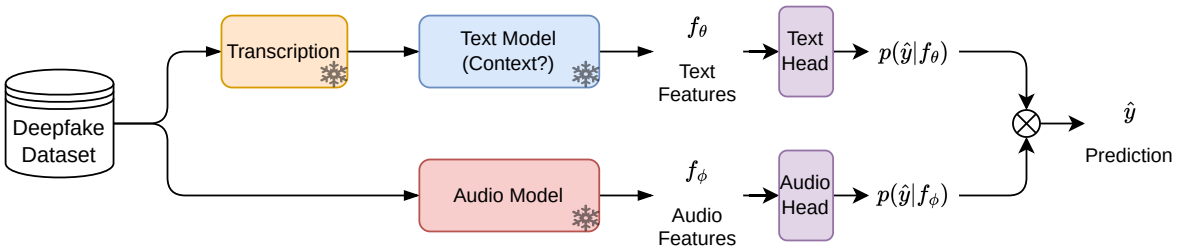

Figure 3: *Pipeline for Late-Fusion. Predictions from the audio and text models are combined using a weighted mean to produce the final classification.*

Abhay: I need to fix these arrowheads. Are centered captions better?