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, DADA, 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

Audio deepfakes are like digital ventriloquists, using deep learning techniques to create voices that convincingly impersonate real people [1]. Highly realistic AI-generated content such as images, audio, and even videos can be created with open-source software. Distinguishing between fake and real media becomes nearly impossible for the ordinary user. Deepfakes thus affect a range of domains, from political discourse to personal security. These deepfakes are not only used in creating political propaganda to manipulate public opinion [2] but have also become tools for conducting phone scams [3], posing significant threats to individual privacy and security. Yet robust detection mechanisms are still lacking [4]. This paper contributes to proactive measures in detecting audio deepfakes, 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.

tation
eeded

Current automated deepfake detection methods typically use deep learning, and achieve high benchmark performance [5, 6].

Lightweight Convolutional Neural Networks (LCNN) [7] have been widely employed due to their efficiency in capturing spoofing artefacts in raw waveforms. RawNet2 [8], a variant of RawNet, leverages residual connections and attention mechanisms to improve feature extraction for deepfake detection. AASIST (Audio Anti-Spoofing using Integrated Spectro-Temporal features) [9] integrates spectro-temporal representations to enhance robustness against unseen attacks. RawGAT [10], a model combining raw waveform processing with Graph Attention Networks (GAT), has shown promising results in handling diverse spoofing techniques. More recently, SLIM (Style-Linguistics Mismatch Model) [11] aims to leverage the mismatch betwen a speaker's style and linguistic content through self-supervised pretraining on real speech to detect spoofing.

But deepfake most current detectors learn to spot acoustic artefacts produced by the generator , which is a major source of brittleness. Trained deepfake detectors have poor performance when deployed in a real-world setting, regardless of performance on standardized benchmarks [4].

citation
needed

Aiming to increase robustness in deepfake detection, we propose a new pipeline that incorporates contextual information. Deepfakes can be leveraged to misattribute quotes to public figures. While a deepfake detector focused on audio might fail to find the spoof, a human might spot the mismatch between the content of the speech and the public figure's otherwise well-known opinion. Our approach, named Defense Against the Deepfake Arts (DADA), takes this intuition into account. We augment an audio-based deepfake classifier with an authorship attribution model. The latter maps the speech transcript to the most likely author. A final classification layer detects a mismatch between the authors predicted by the audio-based classifier and the text-based author predictor. We want to answer the question of what is the probability of the anticipated speaker having made the given statement.

Deepfake generation methods broadly fall into 6 categories [12]. 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 second 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. Emotion fake alters the emotions of a speaker's voice while preserving the original words and speaker identity [13]. Scene fake involves modifying the background sounds of a recording, leaving the speaker and content untouched [14]. Finally, partial fakes involve 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 [15]. Any of these techniques may be used to create deepfakes of public figures.

Recent advancements in authorship attribution using deep

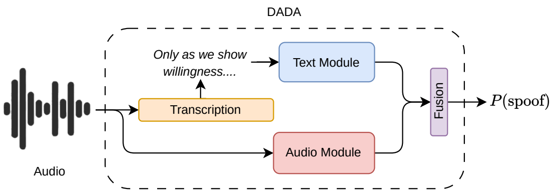

Figure 1: *Overview of the DADA pipeline. Audio files are first transcribed to obtain textual content. Features are then extracted from both the audio and text using separate models. Finally, these features are combined (using either mid fusion or late fusion strategies) to generate the final prediction.*

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 [16, 17]. For instance, BertAA [18] is a fine-tuned pretrained BERT model for authorship attribution, demonstrating competitive performance on multiple datasets. We can also find several papers that use techniques from contrastive learning in creating discriminative embedding spaces. Works like DeepStyle [19], 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 [20]. A transformer-based model [21] are used also for authorship attribution on arXiv manuscripts, achieving high attribution accuracy in a large-scale dataset with up to 2,000 authors.

Most current state of the art systems are benchmarked on the ASVspoof dataset [5], which comprises scripted recordings and spoofs generated using predefined synthesis techniques. A new dataset, In The Wild [4], was released recently to better reflect the actual landscape of deepfake generation. This dataset contains real-world recordings and spoofs generated using a variety of techniques, including voice conversion and TTS. We aim to evaluate our proposed method on both ASVspoof and In The Wild datasets to assess its robustness in detecting real-world deepfakes.

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. An overview of the pipeline is shown in Figure 1.

Our method operates in three stages: transcription, feature extraction and fusion. First, the audio files are transcribed using Whisper [22] 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 features 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 [23]. 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 [24]. 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 λ. We define the function as follows:

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

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

The audio model uses a Wav2Vec2 [25] backbone for feature extraction, which allows learning representations directly from raw audio waveforms. To capture a comprehensive set of features, multiple layers from the model are pooled into a single feature vector. This pooling is achieved using a compression module that employs attentive mean pooling, which allows the model to focus on the most informative parts of the audio signal. Additionally, following [11], a non-linear bottleneck is applied to further refine the feature vector, ensuring that it captures the essential characteristics needed for accurate deepfake detection.

Similar to the text module, audio features (denoted as f_ϕ) are trained using Contrastive Learning, with a triplet loss as in Equation 1 using the cosine distance function. To optimize the margin used in the triplet loss function, we employ an iterative process that involves three stages: initial training with a fixed margin, adaptive margin adjustment, and final training with a fixed margin.

In the first stage, we train the model for a few epochs using an a priori fixed margin, λ_0. This initial training helps achieve some separation between classes and provides a good starting point for further optimization.

Next, we train the model for additional epochs using an adaptive triplet loss introduced in [26]. This constitutes adding an additional soft constraint on the virtual angle between the anchor and the negative sample using another margin λ' as follows.

$$\begin{aligned}\mathcal{L}_{ada} = [d(f_\phi^A, f_\phi^P) - d(f_\phi^A, f_\phi^N) + \lambda]_+ \\ + [d(f_\phi^A, f_\phi^N) + \lambda']_+\end{aligned} \tag{2}$$

In the final stage, we fix the margin as the final margin obtained from the adaptive stage, λ_f, and train the model for a couple more epochs to achieve optimal performance.

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 (Figure 2), 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.

We formulate this fusion strategy solely for the scenario where there exists a purported speaker, consistent with deepfakes for a specific individual/celebrity. We add a binary classification head to both the audio and text models. 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. The final prediction is obtained by taking the weighted mean of the scores from the two branches as shown in Figure 3. 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, the final prediction is obtained as follows:

$$\hat{y} = \alpha \cdot p(\hat{y}|f_\phi) + (1 - \alpha) \cdot p(\hat{y}|f_\theta) \tag{3}$$

Where $p(\hat{y}|f_\phi)$ and $p(\hat{y}|f_\theta)$ are the predictions from the audio and text models respectively, and α is the weight assigned to the audio model's prediction.

See Section 7 after the bibliography for a math description of late fusion

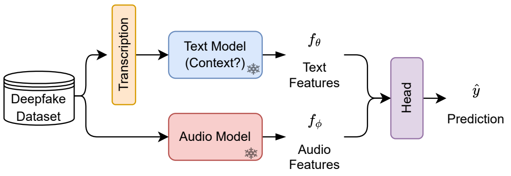

Figure 2: *Pipeline for Mid-Fusion. Features from the text and audio modules are concatenated and passed through an MLP to produce the final classification.*

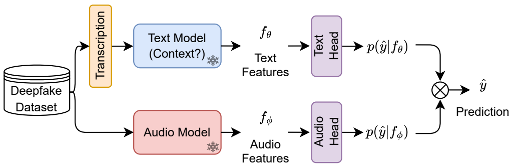

Figure 3: *Pipeline for Late-Fusion. Predictions from the audio and text models are combined using a weighted mean to produce the final classification.*

Method	ASVSpoof21		In The Wild		MLAAD[en]	
	F1 Score	EER	F1 Score	EER	F1 Score	EER
LCNN [7]	0.197	25.5	0.373	65.6	0.654	37.2
Rawnet2 [8]	0.213	22.3	0.602	37.8	0.676	33.9
RawGAT [10]						
SLIM [11]	0.651	4.4	0.898	12.5	0.892	10.7
AASIST [9]	0.707	3.6	0.847	17.5	0.856	14.5
DADA Variants (Ours)						
Audio-Only	…	…	…	…	…	…
Late Fusion	—	—			—	—
Mid Fusion	…	…	…	…	…	…

Table 1: *Performance of DADA vs benchmarks [WIP]*

4. Experiments and Results

4.1. Text Module

4.1.1. Datasets

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 [27, 28]. 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. 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. A random decision would yield 50% accuracy.

4.1.3. *Benchmarking*

We consider several state-of-the-art encoder-only architectures to benchmark the effectiveness of the text model, including Flan-T5 [29], RoBERTa [30], DeBERTa [31], ModernBERT [32], and the encoder of T5[33]. 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.

Model	Wikiquotes		In The Wild		VoxCeleb	
	F1	ABX	F1	ABX	F1	ABX
T5 Encoder [33]		0.76				
Flan-T5 [29]		0.73				
ModernBERT [32]						
DeBERTa [31]		0.77				

Table 2: *Performance of Text Model finetuned on Wikiquotes*

All experiments were conducted on a single NVIDIA A100 GPU with 40GB vRAM.

4.2. Audio Module

The audio module is pre-trained on the speaker representation learning task (Section 3.2) on an aggregation of the CommonVoice [34], RAVDESS [35] and VoxCeleb2 [36] datasets. Both CommonVoice and RAVDESS contain recordings of speakers reading predefined sentences, while VoxCeleb2 contains speech from celebrities extracted from YouTube videos.

We train the model using the adaptive-margin contrastive learning strategy described in Section 3.2, and evaluate it performance with ABX accuracy.

Abhay: might want to shift voxceleb2 desc to text module

4.3. Mid Fusion

We train both our fusions pipeline on the ASVSpoof Dataset. This dataset contains a variety of real and spoofed audio samples designed to benchmark the performance of automatic speaker verification systems against spoofing attacks. We then evaluate the models' performance on out-of-domain datasets, In The Wild [4] and the English subset of MLAAD [37]. This helps us assess its generalization and robustness to real-world deepfakes.

4.3.1. *Metrics*

We consider two metrics for our evaluations: F1-score and Equal Error Rate (EER). EER is a standard metric for audio deepfake detection systems, denoting the the point at which the false acceptance rate (FAR) and the false rejection rate (FRR) are equal. A lower EER indicates better performance.

4.4. Results

Our results can be found in Table 1. We observe that both the mid and late fusion strategies outperform the state-of-the-art benchmarks on …

5. Conclusions / Discussion

In this paper, we introduced DADA, a novel approach for audio deepfake detection that leverages both audio and textual features to improve detection accuracy. By incorporating context awareness through authorship attribution, our method enhances the robustness of deepfake detection systems. We proposed two fusion strategies, mid fusion and late fusion, to effectively combine audio and text features. Our experiments demonstrate that DADA outperforms state-of-the-art benchmarks on multiple datasets, highlighting the potential of integrating contextual information in deepfake detection pipelines. Future work will focus on further refining the fusion strategies and exploring additional datasets to validate the generalizability of our approach.

6. References

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

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

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

[4] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" *Interspeech*, 2022.

[5] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, and K. A. Lee, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 31, pp. 2507–2522, 2023.

[6] X.-M. Zeng, J.-T. Zhang, K. Li, Z.-L. Liu, W.-L. Xie, and Y. Song, "Deepfake algorithm recognition system with augmented data for add 2023 challenge," in *Proceedings of IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis*, 2023.

[7] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection." [Online]. Available: https://arxiv.org/abs/2103.11326

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

[9] J. weon Jung, H.-S. Heo, H. Tak, H. jin Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," 2021. [Online]. Available: https://arxiv.org/abs/2110.01200

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

[11] Y. Zhu, S. Koppisetti, T. Tran, and G. Bharaj, "Slim: Style-linguistics mismatch model for generalized audio deepfake detection," 2024. [Online]. Available: https://arxiv.org/abs/2407.18517

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[26] K. Nguyen, H. H. Nguyen, and A. Tiulpin, "Adatriplet: Adaptive gradient triplet loss with automatic margin learning for forensic medical image matching," 2022. [Online]. Available: https://arxiv.org/abs/2205.02849

[27] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair abx task: Analysis of the classical mfc/plp pipeline," in *INTERSPEECH 2013: 14th Annual Conference of the International Speech Communication Association*, 2013, pp. 1–5.

[28] M. Hallap, E. Dupoux, and E. Dunbar, "Evaluating context-invariance in unsupervised speech representations," 2023. [Online]. Available: https://arxiv.org/abs/2210.15775

[29] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[30] Y. Liu, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, vol. 364, 2019.

[31] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.

[32] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen *et al.*, "Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference," *arXiv preprint arXiv:2412.13663*, 2024.

[33] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[34] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 2020, pp. 4211–4215.

[35] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess)," Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1188976

[36] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[37] N. M. Müller, P. Kawa, W. H. Choong, E. Casanova, E. Gölge, T. Müller, P. Syga, P. Sperl, and K. Böttinger, "Mlaad: The multi-language audio anti-spoofing dataset," 2024. [Online]. Available: https://arxiv.org/abs/2401.09512

7. Late fusion

X the random variable equal to 0 if text is spoofed, 1 if real

A the random variable equal to author of text

T the random variable equal to text

The author id model gives

$$p(A = a|T = t, X = 1)$$

so I'm not sure how to use it.

We want to quantify $p(X = 0|T = t)$

Rewrite as

$$p(X = 0|T = t) = \sum_a p(X = 0|T = t, A = a)p(A = a|T = t)$$

(marginalization + chain rule)

Make the following assumption:

$$p(X = 0|T = t, A = a) \approx p(X = 0|A = a)$$

It means that the content of the text does not add much w.r.t. to detecting spoofing, as long as one has the author information. This would be particularly justified if certain authors are spoofed much more often than others: just knowing that a certain piece of text is attributed to an author would be a good indication of whether or not it has been spoofed, independently of the textual content.

With this

$$p(X = 0|T = t) = \sum_a p(X = 0|A = a)p(A = a|T = t)$$

$p(X = 0|A = a)$ is given by how often author a has been spoofed (can be computed on the training set), and $p(A = a|T = t)$ is given by author id model.

In the end, if $p(X = 0|A = a)$ is quite constant across authors, e.g. $\forall a\, p(X = 0|A = a) \approx \tau$ then $p(X = 0|T = t) = \sum_a \tau p(A = a|T = t) = \tau$