Présentation Finale

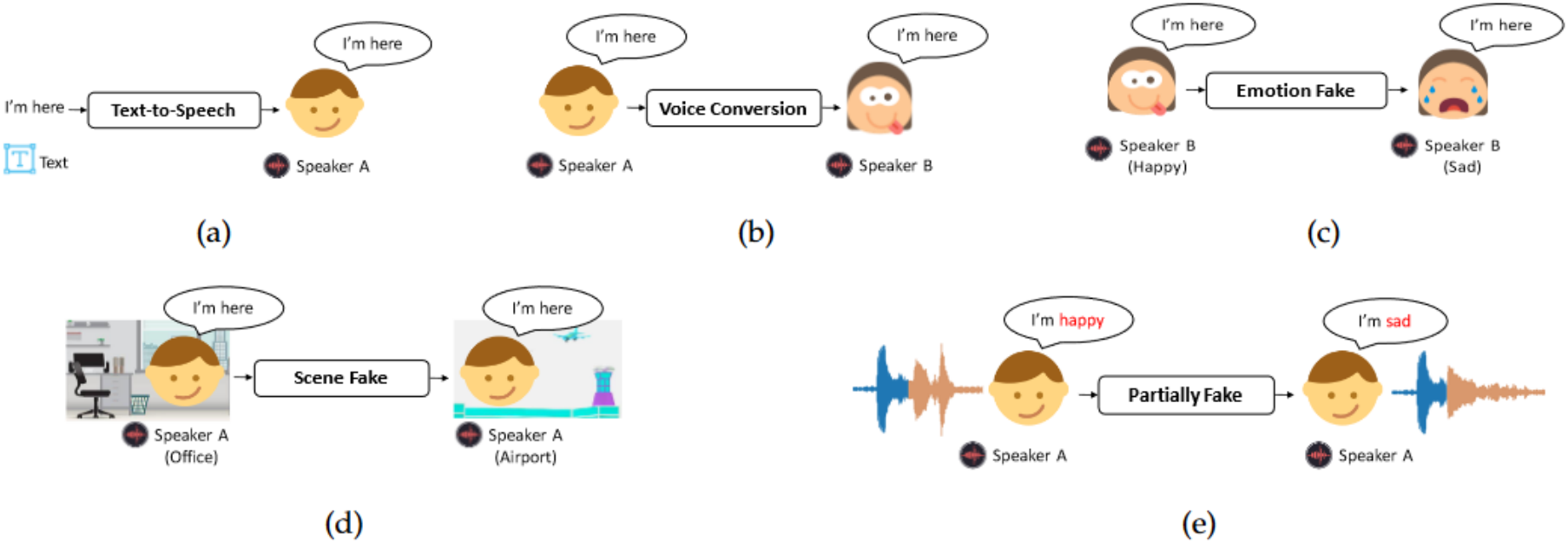
Amélioration des systèmes de vérification du locuteur contre les attaques Deepfakes

Encadré par **Réda Dehak** et **Théo Lepage**

Contexte

Usurpation d'identité avec de l'audio deepfake

Différents types de deepfakes [10]:



Objectif du système:

- Détecter les *artefacts* laissé par l'algorithme dans l'audio

Problèmes:

- Type d'attaque inconnu
- Compression de l'audio sur les plateformes en lignes
- Les attaquants ont accès aux systèmes de détections pour améliorer la qualité de leur algorithme

Contexte

Base de données ASVSPOOF5 [9]

Phase I : Génération de deepfakes

- les participants soumettent des systèmes et peuvent les *optimiser* contre des systèmes de détections pré-existants
- seuls les systèmes *optimisés* sont utilisé dans le jeu d'évaluation
- les systèmes *non-optimisé* sont réservé pour le jeu d'entraînement
- des algorithmes de compressions sont appliqués sur les extraits audio du jeu d'évaluation

Phase II : Détection de deepfakes

- optimisation de systèmes de détection à partir des jeux de données d'entraînement de de développement

Etat de l'art

Modèle auto-supervisé : le WavLM [1]

Structure du WavLM [1]

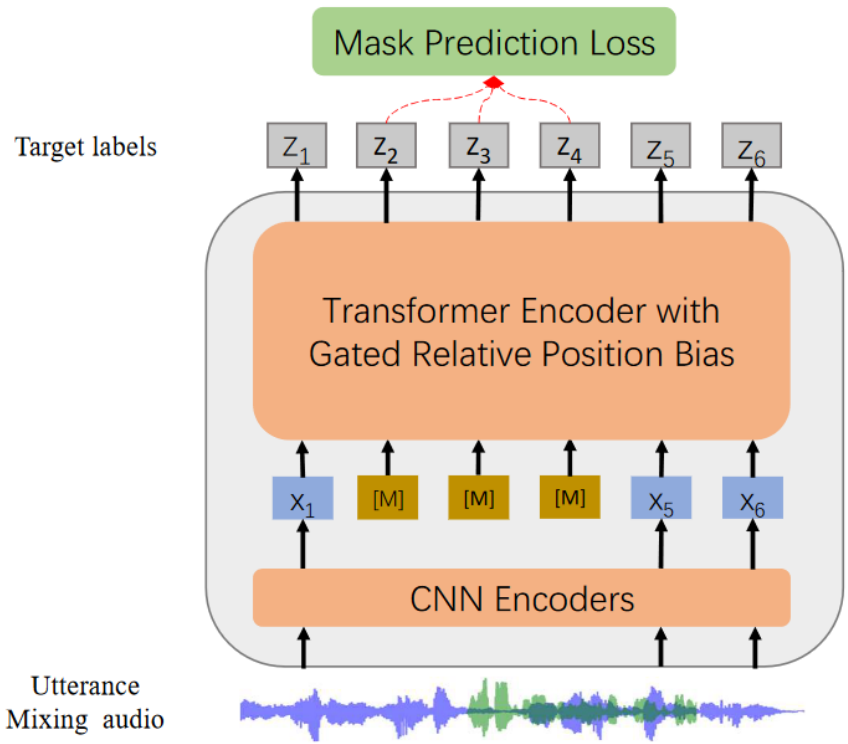


Fig. 1. Model Architecture.

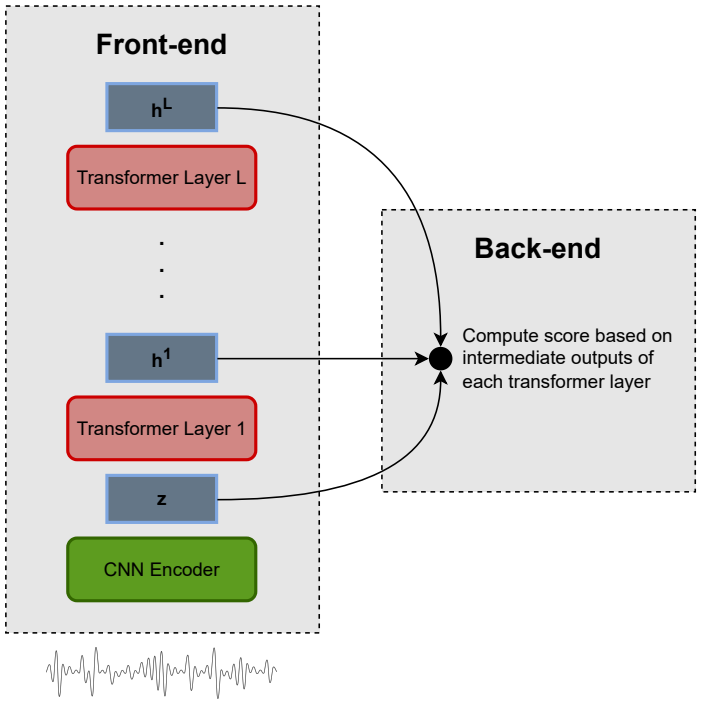
- s'entraine à prédire les tokens sur lesquelles du bruit ou d'autres extraits audio ont été ajouté
- entrainé sur de grands corpus de voix (960h pour la version *Base*)
- apprend les caractéristiques de la voix humaine, cela en fait un excellent extracteur de caractéristiques pour la détection de deepfake

Etat de l'art

Modèle auto-supervisé pour l'extraction de caractéristiques

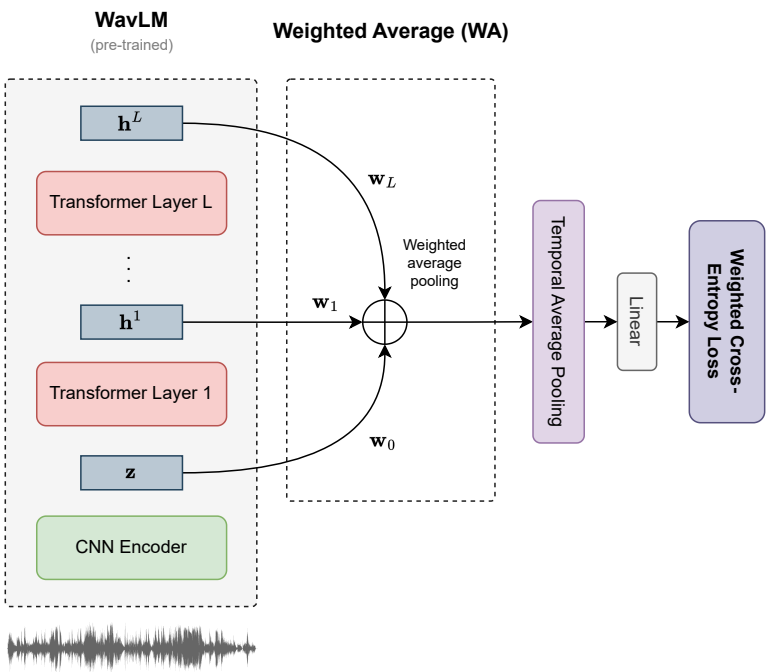
Système composé en deux parties:

- *Front-end*: modèle pré-entrainé pour extraire les caractéristiques de l'audio
- *Back-end*: modèle entrainé "from scratch", se basant sur les sorties du front-end



Méthode simple pour utiliser efficacement les sorties du front-end [7]

Structure du modèle[7]:



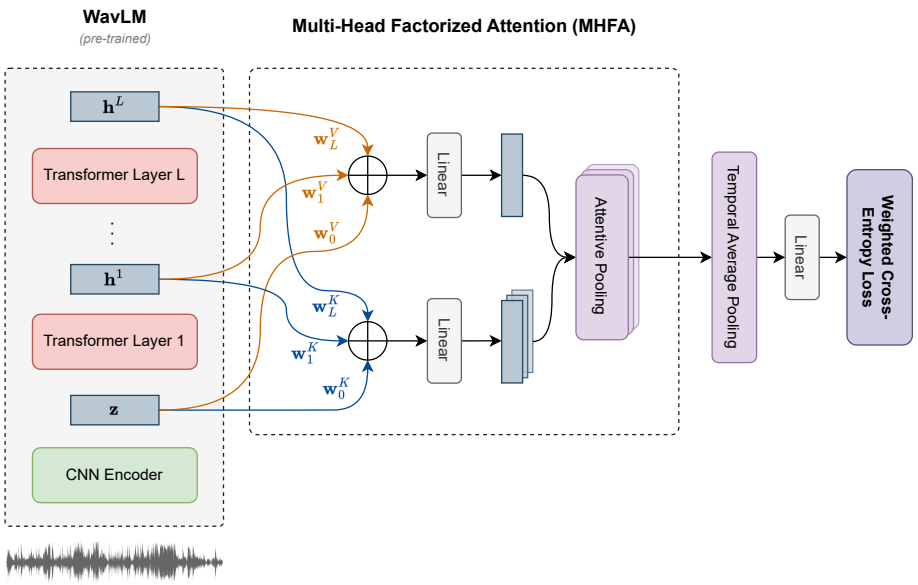
- moyenne pondéré des couches Transformer
- moyenne sur la dimension du temps

Etat de l'art

Modèle MHFA pour la détection de deepfake

Modèle utilisé originellement pour la vérification du locuteur [6] [4], adapté pour la détection de deepfake [7]

Structure du modèle[7]:



- calcul un score d'attention (*clés*, *valeurs*) pour chacune des couches Transformers
- projette l'espace des *clés* vers plusieurs *têtes* d'attention
- applique les poids des *clés* sur les *valeurs* et effectue une moyenne sur la dimension du temps

Proposition

Objectif :

Trouver la meilleure manière d'utiliser les sorties du front-end.

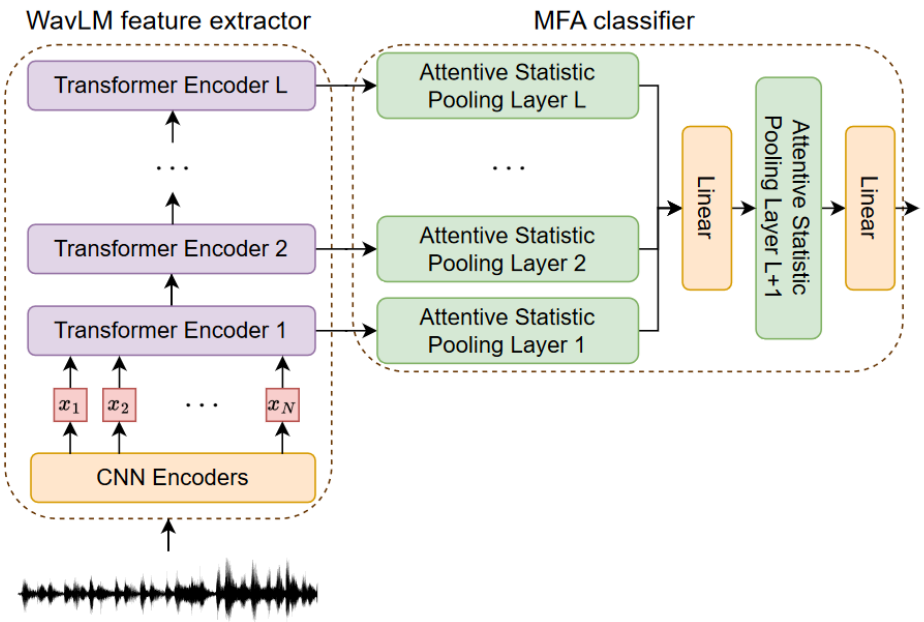
Méthodes comparées :

- MFA : Multi-Fusion Attentive classifier [2]
- Attentive Merging method [5]
- WA AASIST : fusion de la méthode Weighted Average [7] avec le modèle AASIST [3] [8]

Proposition

MFA

Structure du modèle[2]:



Couche *Attentive Statistics Pooling* [2]

$$\mu = \sum_{t=1}^{T} \alpha_t \mathbf{z}_t,$$

$$\sigma = \sqrt{\sum_{t}^{T} \alpha_t \mathbf{z}_t \odot \mathbf{z}_t - \mu \odot \mu},$$

$$e_t = \boldsymbol{v}^T f\left(\boldsymbol{W}\mathbf{z}_t + \boldsymbol{b}\right) + k,$$

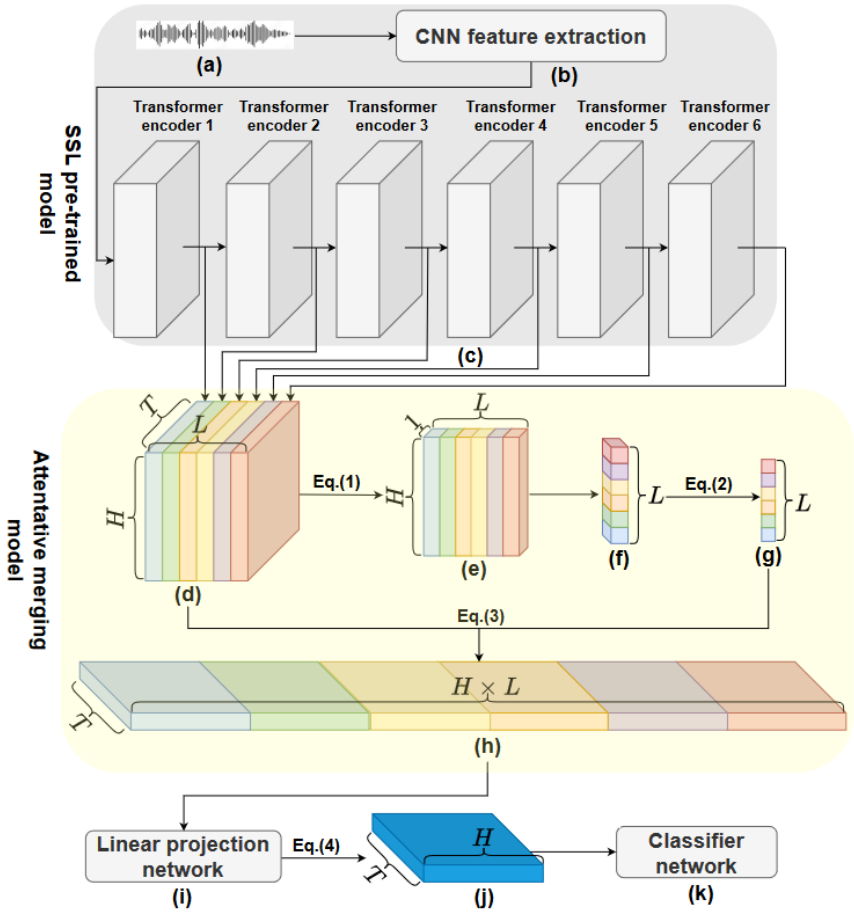
$$\alpha_t = \frac{\exp\left(e_t\right)}{\sum_{\tau}^{T} \exp\left(e_\tau\right)},$$

- applique la couche *ASP* sur chacune des couches Transformer sur la dimension du temps
- applique la couche *ASP* sur chacune des sorties, sur la nouvelle dimension créée

Proposition

Attentive Merging method

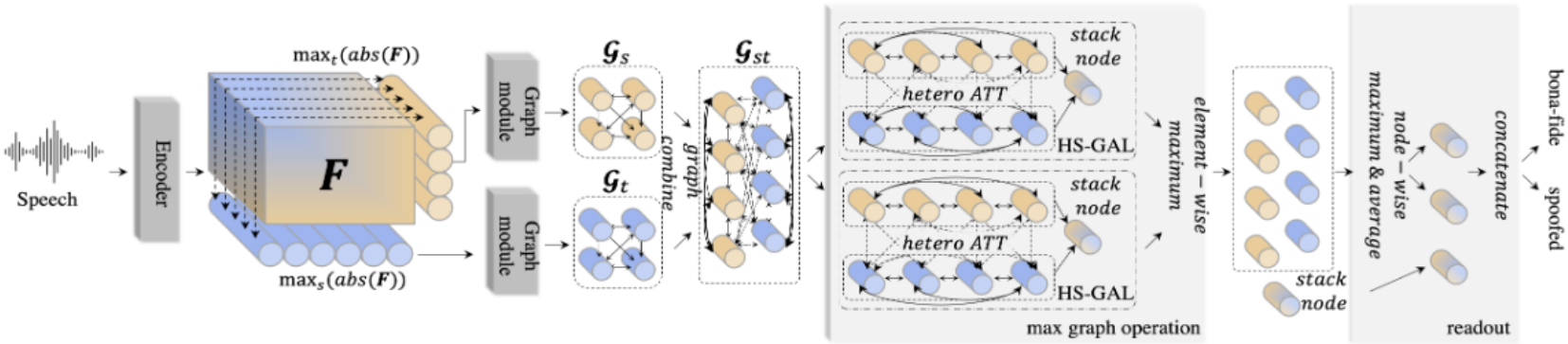
Structure du modèle[5]:



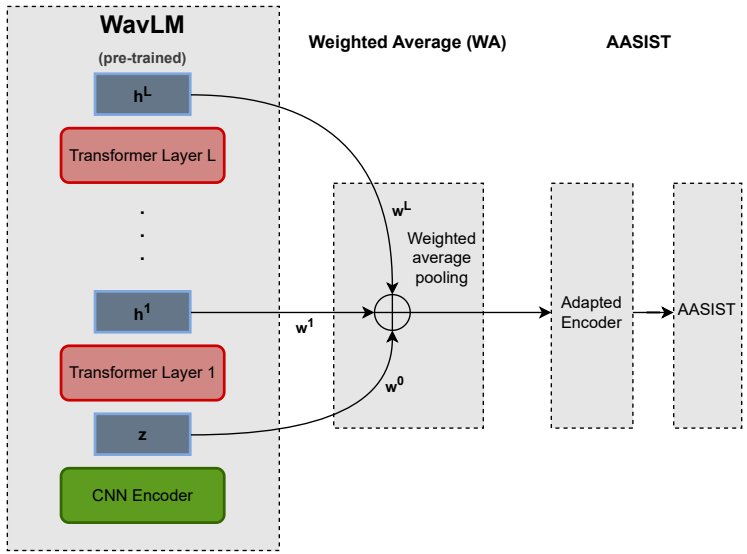
- calcule un score d'attention pour chacune des couches Transformer *(d)* - *(g)*
- applique les poids sur chacune des couches et aplati la dimension *(h)*

Proposition
WA AASIST

Modèle original AASIST [3] :



Structure adaptée, intégrant la méthode de Weighted Average, inspiré du travail [8] :



Protocole d'évaluation

Jeu de donnée *ASVSPOOF5*

L'ensemble de développement a été divisé en deux parties pour créer un jeu de validation et un jeu d'évaluation

Subset	**Usage**	**# utterances**	**% spoofed**
Training	Model training	182,357	89.69
Development (1)	Validation	37,091	77.77
Development (2)	Evaluation	103,859	77.77

Equal Error Rate (EER) sur la tâche de détection: Probabilité d'erreur lorsque l'on choisis un seuil pour lequel on a autant de faux positifs que de faux négatifs sur le jeu de données d'évaluation.

min-DCF: Score qui prend en compte la probabilité de la catégorie cible (P_{tar}), et qui attribue un coût aux deux types d'erreurs: faux positif (C_{FA}) et faux négatif (C_{miss}).

$$C_{det}(P_{miss}, P_{FA}) = C_{miss} P_{miss} P_{tar} + C_{FA} P_{FA} (1 - P_{tar})$$

Augmentation des données

Augmentations utilisées en vérification du locuteur (dénoté *basique*):

- Réverbération
- Ajout d'audio par dessus l'extrait: bruit, musique, voix

Augmentations spécifiques à la détection de deepfake:

- *Codec*: Applique une compression puis décompression (avec perte) selon un certain codec
- *Trans-Codecs*: Applique deux augmentations *Codec* différentes à la suite

Résultats

Résultats sur le jeu de validation :

Modèle	**min-DCF**	**EER**
Weighted Average	***0.0486***	***1.98%***
MHFA	0.0533	2.12%
MFA	0.0564	2.45%
Attentive Merging	0.1690	6.61%
WA AASIST	0.0692	2.94%

Résultats

Résultats sur le jeu d'évaluation :

Modèle	**min-DCF**	**EER**
Weighted Average	***0.0529***	***2.04%***
MHFA	0.0540	2.09%
MFA	0.0586	2.39%
Attentive Merging	0.1764	6.90%
WA AASIST	0.0697	2.93%

Conclusions et perspectives

- L'usage de modèles auto-supervisés pré-entraînés devient primordial pour détecter efficacement les deepfakes, notamment sur des attaques inconnus au système

- La manière d'exploiter et d'optimiser les paramètres du front-end impacte beaucoup les performances

Bibliographie I

[1] Sanyuan Chen et al. "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6 (Oct. 2022), pp. 1505–1518.

[2] Yinlin Guo et al. *Audio Deepfake Detection with Self-Supervised WavLM and Multi-Fusion Attentive Classifier*. 2024. eprint: 2312.08089.

[3] Jee-weon Jung et al. *AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks*. 2021. eprint: 2110.01200.

[4] Victor Miara, Theo Lepage, and Reda Dehak. *Towards Supervised Performance on Speaker Verification with Self-Supervised Learning by Leveraging Large-Scale ASR Models*. 2024. eprint: 2406.02285.

Bibliographie II

[5] Zihan Pan et al. *Attentive Merging of Hidden Embeddings from Pre-trained Speech Model for Anti-spoofing Detection*. 2024. eprint: 2406.10283.

[6] Junyi Peng et al. *An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification*. 2022. eprint: 2210.01273.

[7] Theophile Stourbe et al. *Exploring WavLM Back-ends for Speech Spoofing and Deepfake Detection*. 2024. eprint: 2409.05032.

[8] Hemlata Tak et al. *Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation*. 2022. eprint: 2202.12233.

[9] Xin Wang et al. *ASVspoof 5: Crowdsourced Speech Data, Deepfakes, and Adversarial Attacks at Scale*. 2024. eprint: 2408.08739.

Bibliographie III

[10] Jiangyan Yi et al. *Audio Deepfake Detection: A Survey*. 2023. eprint: 2308.14970.

Questions

Merci de votre attention !

Des questions ?