Impact des conditions d'attaques sur les contre-mesures pour la reconnaissance du locuteur

Nathan Souviraà-Labastie, Maxime Baelde, Thomas Malet, Raphaël Greff

Impact des conditions d'attaques sur les contre-mesures pour la reconnaissance du locuteur

Nathan SOUVIRAÀ-LABASTIE[1], Maxime BAELDE[1], Thomas MALET[2], Raphaël GREFF[1]

[1]A-Volute SAS, 19 rue de la Ladrie, Polygône Square, 59491 Villeneuve d'Ascq, France

[2]École Centrale de Lille, 59650 Villeneuve d'Ascq, France

prenom.nom@nahimic.com ; prenom.nom@centralelille.fr

Résumé – Cet article est une pré-étude en vue d'une soumission au *challenge* ASVSpoof 2019. Ce *challenge* permet d'évaluer les contre-mesures aux attaques de systèmes de reconnaissance du locuteur (par voie logicielle ou physique, les deux tâches du *challenge*). Une première étape est l'étude des faiblesses des systèmes de référence. En effet, ils ont plus de difficultés sur certaines catégories d'attaques que d'autres. En prenant en compte cet élément, nous avons créé, pour les deux tâches, des systèmes spécialisés améliorant les performances sur ces catégories problématiques. Ces systèmes seront utilisés dans la procédure de fusion de notre soumission au *challenge* (article soumis à INTERSPEECH 2019).

Abstract – This article is a preliminary towards our submission to the ASVSpoof 2019 Challenge. This challenge allows to evaluate countermeasures to speaker recognition system attacks (by logical or physical access, the two tasks of the challenge). A first step is to study the weaknesses of the baseline systems. Indeed, they have more difficulties on some categories of attacks than others. Taking this into account, we have created, for both tasks, specialized systems that improve performance in these problematic categories. These systems will be used in a fusion procedure of our submission to the challenge (article submitted to INTERSPEECH 2019).

1 Le challenge ASVSpoof 2019

Le challenge ASVSpoof [1] a pour but d'évaluer les contre-mesures (CM) utilisées en tandem avec les systèmes de reconnaissance du locuteur (*Automatic Speaker Recognition*, ASV). L'édition 2019 est composée de deux tâches précédemment définies dans l'édition 2015 [1] (tâche *Logical Access* (LA), focalisée sur la voix de synthèse), et 2017 [2] (tâche *Physical Access* (PA), focalisée sur les *replay attacks*).

1.1 Description des tâches

La tâche LA, détection de voix de synthèse. Elle comprend six systèmes de synthèse vocale pour générer les exemples de voix synthétisées. Ces systèmes sont notés SS_1, SS_2, SS_4, US_1, VC_1 et VC_4, et utilisent principalement des réseaux de neurones et des vocodeurs (pour générer de la voix de synthèse). Plus de détails sur ces systèmes sont disponibles dans le plan d'évaluation de l'édition 2019 du challenge [3].

La tâche PA, *replay attacks*. Elle comprend cinq catégories de conditions [2]. Tout d'abord, trois catégories liées à l'*environnement* : la taille de la pièce ($\mathcal{S}$), le temps de réverbération T60 (R), et la distance entre le locuteur et le système de reconnaissance vocale ($\mathcal{D}_s$). Deuxièmement, l'*attaque* qui consiste en deux catégories : la distance attaquant - locuteur ($\mathcal{Z}$) et la qualité du système de restitution (Q). Chaque paramètre peut prendre des valeurs dans trois plages distinctes décrites dans la Table 1. Plus de détails sur la spécification de ces plages de valeurs sont disponibles dans le plan d'évaluation de l'édition 2019 du challenge [3].

TABLE 1 – Plages de valeurs des catégories de la tâche PA [3].

	Plages →	a/A	b/B	c/C
Catégories	$\mathcal{S}$ (m^2)	2-5	5-10	10-20
	R (ms)	50-200	200-600	600-1000
	$\mathcal{D}_s$ (cm)	10-50	50-100	100-150
	$\mathcal{Z}$ (cm)	10-50	50-100	> 100
	Q	Parfait	Élevé	Faible

1.2 Les métriques d'évaluation : EER et t-DCF

L'*Equal Error Rate* (EER) [4] est obtenu pour le point de fonctionnement du système pour lequel le taux de vrais positifs est égal au taux de faux positifs (FP). L'EER présente certains inconvénients. Tout d'abord, l'EER est mal adapté aux problèmes déséquilibrés. Deuxièmement, l'EER n'est pas un indicateur fiable de la performance lorsque plusieurs systèmes sont utilisés en tandem, comme ce sera le cas avec les systèmes ASV et CM.

La *tandem Detection Cost Function* (t-DCF) [4] est calculé en prenant en compte les différents types d'erreurs de détection du système tandem ASV-CM. Un coût est attribué à chaque type d'erreurs (faux positifs ou faux négatifs) de chaque système.

1. http ://www.asvspoof.org/
2. Les notations des plages et catégories sont celles du *challenge* [3].

C'est depuis 2019 la métrique officielle utilisée pour le classement du *challenge* ASVSpoof. C'est pourquoi les tables de cet article affichent rarement l'EER bien que les résultats en terme d'EER soient également commentés.

2 Les faiblesses de la *baseline*

Les organisateurs du *challenge* ont défini deux systèmes de référence, dits *baselines*, (LFCC-GMM et CQCC-GMM) [3] qui ont démontré les meilleures performances (en tant que système non-fusionné) sur les corpus des précédents *challenge*s 2015 et 2017.

2.1 Description de la *baseline*

Chaque *baseline* utilise un des deux types de *features* suivant : les LFCC (Linear Frequency Cepstral Coefficients) [5] calculés en prenant la transformée en cosinus discrète du spectre de puissance, les coefficients derivés et dérivés secondes étant également concaténés au LFCC original ; les CQCC (Constant Q Cepstral Coefficients) [6] calculés en prenant le cepstre de la transformation en Q constant du signal audio.

Pour chaque classe (*genuine* et *spoof*), un modèle de mélange gaussien (*Gaussian Mixture Model*, GMM) est entrainé sur les *features* correspondants (LFCC ou CQCC) en utilisant un nombre prédéfini de composantes (512 dans le cas de la *baseline*). Pour chaque exemple, une log-vraisemblance est calculée pour les classes *genuine* et *spoof* et le *score* final est la différence de ces deux log-vraisemblances.

Nous avons reproduit les deux *baselines* à l'aide du code fourni par l'organisation et nous avons obtenu des résultats similaires. Le lecteur peut trouver le détail de cette reproduction dans les deux dernières colonnes de la Table 2 pour la tâche LA et dans les sous-légendes des Tables 3 pour la tâche PA.

2.2 LA : Influence du type de synthèse vocale.

Les résultats par catégorie (de systèmes de synthèse vocale) sur le corpus de développement pour les deux *baselines* sont disponibles dans la Table 2. L'EER et le min t-DCF obtenus avec les *baselines* sont déjà faibles voire égaux à 0 pour certains systèmes de synthèse vocale. Cependant, certains systèmes introduisent plus d'erreurs que d'autres, par exemple US_1 et VC_4. Par conséquent, l'une de nos contributions décrites dans la section suivante consiste à prendre en compte les différents systèmes de synthèse au cours de l'apprentissage du modèle afin de mieux résoudre la tâche LA.

2.3 PA : Influence des conditions d'attaques.

Les résultats par catégorie sur le corpus de développement des deux *baselines* sont affichés dans la Table 3. Il est à noter que les moyennes affichées sont des moyennes pondérées, le nombre d'exemples par plage étant différent. Nous pouvons observer que les deux *baselines* ont des difficultés à classer correctement les exemples audio lorsque le temps de réverbération T60 est très faible (condition « a » de R) ou lorsque la qualité du système de restitution est parfaite (condition « A » de Q). Lorsque ces deux conditions sont présentes, les résultats sont d'autant plus dégradés (par exemple 36,7% EER et 0,811 min t-DCF pour le CQCC-GMM). Ainsi, nous pouvons en déduire que la réverbération et les effets induits par un mauvais système de restitution sont des informations significatives pour la détection de *replay attack* (en considérant que la *baseline* est assez performante). Nous pouvons déjà observer que les LFCC sont mieux conçus que les CQCC pour la classification des exemples avec un petit T60. Inversement, les CQCC fonctionnent mieux dans le cas des systèmes de restitution parfaits.

Nos contributions décrites dans la section suivante consistent à prendre en compte toutes ces observations pour mieux résoudre la tâche PA.

3. Notés plus tard † et ♣.

3 Contributions

3.1 GMMs spécialisés par catégorie d'attaques

Notre principale contribution est une méthode de classification qui exploite les informations de la « catégorie » (par exemple $\mathcal{S}$, R, $\mathcal{D}_s$, $\mathcal{Z}$ ou Q dans la tâche PA) durant l'apprentissage. Suite aux observations de la section 2, N GMMS spécifiques sont entrainés sur N différents sous-ensembles du corpus d'apprentissage *spoof* (les sous-ensembles examinés sont décrits ci-après). La log-vraisemblance globale est ensuite calculée à l'étape de classification comme suit :

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{genuine}} - \log\left(\sum_{n\in\mathcal{E}} \exp(\mathcal{L}_n)\right), \qquad (1)$$

où $\mathcal{E}$ est l'ensemble des N sous-ensembles et $\mathcal{L}_n$ est la log-vraisemblance produite par le n-ème GMM de la classe *spoof*. Ce principe est utilisé par les trois *classifiers* suivants.

Multi-catégories pour la tâche LA. Un *spoof-GMM* est entraîné pour chacun des six types de systèmes de synthèse pris dans $\mathcal{E} = \{SS_1, SS_2, SS_4, SS_4, US_1, VC_1, VC_4\}$. Les six *spoof-GMMS* sont composés de 85 composantes et le *genuine GMM* de 512 composantes. Ainsi, le nombre total de composantes GMM est équivalent à celui de la *baseline* ($85 * 6 = 510 \approx 512$). Ce système est appelé *multicat.* dans la Table 2.

Le *3-spoof-GMMs* pour la tâche PA. Un *spoof-GMM* est entraîné pour chaque type de « Qualité du système de restitution (Q) » pris dans $\mathcal{E} = \{\text{Parfait}, \text{Élevé}, \text{Faible}\}$, soit un total de quatre $(3 + 1)$ GMMS composés de 512 composantes.

Le *9-spoof-GMMs* pour la tâche PA. Un *spoof-GMM* est entraîné pour chaque combinaison (9) de « Qualité du système de restitution (Q) » pris dans $\mathrm{Q} = \{\text{Parfait}, \text{Élevé}, \text{Faible}\}$ et de plages de T60 (R) pris dans $\mathrm{R} = \{50 - 200, 200 - 600, 600 - 1000\}$, pour un total de dix $(3 \times 3 + 1)$ GMMs composés de 512 composantes.

TABLE 2 – Résultats en terme de min t-DCF par catégorie pour la tâche LA.

	Catégories →	SS_1	SS_2	SS_4	US_1	VC_1	VC_4	Moyenne	Moyenne dans [3]
Systèmes	LFCC *baseline*	0,0016	0,0000	0,0000	0,1228	0,0035	0,1804	0,07900	0,0663
	LFCC multicat.	0,0044	0,0000	0,0000	0,1471	0,0117	0,0811	0,06370	-
	CQCC *baseline*	0,0000	0,0000	0,0012	0,0002	0,0322	0,0010	0,01300	0,0123
	CQCC multicat.	0,0000	0,0000	0,0009	0,0000	0,0121	0,0000	**0,00319**	-

TABLE 3 – Résultats (min t-DCF) des *baselines* par plage de paramètres pour chaque catégorie PA ($\mathcal{S}$, R, $\mathcal{D}_s$, $\mathcal{Z}$ et Q).

(a) Min t-DCF de la *baseline* † (moyenne *0,2257* ; *0,2554* dans [3]).

↓ Plages ↓	$\mathcal{S}$	**R**	$\mathcal{D}_s$	$\mathcal{Z}$	**Q**
aaa/AA	0,2174	**0,2902**	0,2345	0,2383	**0,5121**
bbb/BB	0,2358	0,1757	0,2272	0,2135	0,0615
ccc/CC	0,2220	0,1371	0,2106	0,2172	0,0285

(b) Min t-DCF de la *baseline* ♣ (moyenne *0,1928* ; *0,1953* dans [3]).

↓ Plages ↓	$\mathcal{S}$	**R**	$\mathcal{D}_s$	$\mathcal{Z}$	**Q**
aaa/AA	0,1720	**0,3324**	0,2061	0,1970	**0,4583**
bbb/BB	0,1872	0,1483	0,2038	0,1772	0,0530
ccc/CC	0,2039	0,0604	0,1668	0,1986	0,0237

3.2 *Features* Dual-LFCC

Notre deuxième contribution est un nouveau type de *feature* appelé "Dual-LFCC". Il s'agit d'une concaténation des LFCC calculés sur le signal déréverbéré et des LFCC calculés sur le résidu de la déréverbération. La déréverbération est obtenue en utilisant l'algorithme de DOIRE [7], le résidu est obtenu en soustrayant le signal déréverberbé au signal original et les LFCC sont calculés avec les mêmes paramètres que pour la *baseline*.

Cette contribution cible la tâche PA et en particulier le sous-ensemble d'exemples où R est dans la plage $\ll$ a $\gg$, c'est-à-dire où la réverbération fournira peu d'informations au *classifier*. Le but des Dual-LFCC est de mettre en évidence cette information de réverbération afin de réduire la complexité de la tâche de classification. Des *features* semblables, comme le LFCC résiduel, le LFCC réverbe et les homologues CCQC, ont été essayés, mais avec moins de succès que le Dual-LFCC (les résultats ne sont pas présentés dans ce document).

3.3 Le *9-layers-DNN*

Cette contribution se concentre sur l'utilisation d'un réseau de neurones profond (*Deep Neural Network*, DNN) classique. L'objectif est d'évaluer dans quelle mesure le GMM manque de complexité dans la modélisation. Le DNN considéré est un réseau de neurones à 9 couches construit à la manière d'un auto-encodeur où le nombre de neurones par couche est $(250, 200, 150, 100, 150, 200, 250, 90, 2)$. La couche finale est une couche softmax avec 2 neurones, qui peut être interprétée comme les deux probabilités *a posteriori* des classes *genuine* et *spoof*. L'apprentissage est réalisé avec 10 itérations en utilisant l'optimiseur Adam et la perte d'entropie croisée binaire. Des couches *Batch Normalization* sont ajoutées entre chaque couche.

4 Expériences pour la tâche LA

Les résultats de la *baseline* CQCC-GMM sur le corpus de développement était déjà satisfaisants. Cependant, le GMM multi-catégories avec les CQCC que nous avons soumis au *challenge* démontre des améliorations importantes tant en termes d'EER que de min t-DCF (voir les deux dernières colonnes de la Table 2). En particulier, le *classifier* multi-catégories affiche des performances supérieures dans les catégories où les pires résultats ont été observés (voir Section 2.2). Cependant, alors que dans la plupart des catégories, notre système affiche 0% EER sur le corpus de développement, nous pouvons soupçonner que ce système a été surajusté aux données d'entrainement et qu'il ne montrera pas de capacités de généralisation sur le corpus d'évaluation. Par conséquent, nous n'avons pas mis en œuvre de stratégies de fusion pour cette tâche en raison de l'efficacité de notre système unique.

5 Expériences pour la tâche PA

Dans cette section, les différentes expériences ciblant la tâche PA sont présentées. Au total, douze systèmes sont entrainés pour traiter cette tâche, douze correspondant à toutes les combinaisons des trois différents types de *features* (LFCC, CQCC et Dual-LFCC) et des quatre *classifiers* (*baseline*, *3-spoof-GMMs*, *9-spoof-GMMs* et *9-layers-DNN*) (voir Table 4 en Section 5.1). Les résultats de ces douze systèmes en fonction de la plage de paramètres sont ensuite analysés (voir Table 5 dans la Section 5.2) avec pour objectif d'extraire des informations utiles sur la spécificité de chaque système.

5.1 Analyse des combinaisons *classifier/features*

Les résultats globaux sur le corpus de développement de chaque combinaison *classifiers/features* sont disponibles dans la Table 4. Pour l'ensemble des systèmes, on observe peu d'amélioration en terme de min t-DCF comparé aux deux *baselines* † et ♣. Les résultats en terme d'EER (non-affichés) étaient légèrement plus prometteurs, mais pas non plus significatifs. Nous pouvons également noter que le DNN n'améliore pas les résultats, mais sa contribution dans la procédure de fusion [8]

TABLE 4 – Résultats (min t-DCF) pour la tâche PA.

↓ *Classifier* ↓	LFCC	CQCC	Dual-LFCC
baseline	0,226†	0,193♣	0,255‡
3-spoof-GMMs	0,234∗	**0,191**♢	0,257★
9-spoof-GMMs	0,2296	**0,191**♡	0,2328
9-layers-DNN	0,295♯	0,251♠	0,302♮

sera significative. De même, les *classifiers 3-spoof-GMMs* et *9-spoof-GMMs* devaient permettre de prendre en compte la variabilité des différentes catégories (en particulier le temps de réverbération et la qualité du système de restitution). Nous allons voir dans la section suivante que leur apport n'est pas nul. Les symboles (†, ♣, ‡, ∗, ♢, ★, 6, ♡, 8, ♯, ♠, ♮) présents dans la Table 4 représentent les douze systèmes. Ils sont réutilisés dans la Table 5.

5.2 Analyse par plage de paramètres des douze systèmes proposés

Les résultats par plage de paramètres des douze systèmes évalués sont disponibles dans la Table 5. Les résultats sont donnés pour les plages ≪ a ≫ de R et ≪ A ≫ de Q qui ont été identifiées comme les plages problématiques pour la tâche PA. La plupart des systèmes proposés améliorent les résultats sur ces catégories spécifiques (en EER ou en min t-DCF). Les meilleurs résultats sont indiqués en caractères gras.

Il est intéressant de se rappeler que nous avons conçu les Dual-LFCC pour résoudre le problème lié au faible temps de réverbération, mais dans la pratique, les résultats ne se sont pas améliorés sur ce cas mais plutôt sur celui où la qualité du système de restitution est de bonne qualité. Le système 8 (resp. ★) est notamment le (resp. troisième) plus performant en terme d'EER sur la plage ≪ A ≫ de Q. Nous n'avons pour le moment pas réussi à expliquer cette tendance.

De plus, les systèmes identifiés dans la Table 5 améliorent significativement les résultats globaux (dans le cadre des systèmes fusionnés [8]) alors que leurs résultats globaux individuels présentés dans la Table 4 étaient comparables à ceux des *baselines*. Tous ces résultats spécifiques seront pris en compte dans la conception de systèmes fusionnés en vue de notre soumission au *challenge*.

6 Conclusions

Cet article présente plusieurs de nos contributions en vue du *challenge* ASVSpoof 2019. La principale contribution présentée ici est la prise en compte au moment de l'apprentissage des conditions d'attaques fournies pour chaque exemple du corpus d'apprentissage. En effet, une étude expérimentale approfondie nous a permis d'observer de faibles résultats pour les deux *baselines* sur certaines catégories d'exemples, ce qui a motivé ce choix. Nos contributions, à savoir un apprentissage par catégorie et des *features* mettant en avant les informations liées à la réverbération, ont permis d'améliorer les performances sur les catégories d'exemples les plus problématiques. Ces résultats et contributions serviront de base à la préparation de systèmes fusionnés en vue de notre soumission au *challenge* [8].

TABLE 5 – Résultats par plage de paramètres. Tâche PA.

Plages →	a de R		A de Q	
↓ Système ↓	EER	t-DCF	EER	t-DCF
†	16,22	0,290	23,13	0,512
♣	17,88	0,332	21,33	0,458
‡	15,54	0,314	20,29	0,470
∗	**11,98**	0,252	20,79	0,506
♢	17,17	0,355	18,35	0,420
★	12,94	0,280	18,31	0,450
6	12,54	0,271	19,83	0,487
♡	17,72	0,362	17,94	**0,413**
8	12,99	0,298	**17,37**	0,450
♯	14,72	0,287	24,91	0,656
♠	13,72	**0,237**	26,16	0,610
♮	16,55	0,311	27,68	0,689

Références

[1] Z. Wu, *et al*. *Spoofing and countermeasures for speaker verification : A survey*. Speech Communication, vol. 66, pp. 130-153, 2015.

[2] Z. Wu, *et al*. *ASVspoof : The automatic speaker verification spoofing and countermeasures challenge*. IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 4, pp. 588-604, 2017.

[3] ASVspoof consortium. *Asvspoof 2019 : Automatic speaker verification spoofing and countermeasures challenge evaluation plan*. p. 19, 2019.

[4] T. Kinnunen, *et al*. *t-DCF : a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification*. Odyssey 2018 The Speaker and Language Recognition Workshop. ISCA, pp. 312-319, 2018.

[5] M. Sahidullah, *et al*. *A comparison of features for synthetic speech detection*. INTERSPEECH 2015, pp. 2087-2091.

[6] M. Todisco, *et al*. *Constant q cepstral coefficients : a spoofing countermeasure for automatic speaker verification*. Computer, Speech and Language, vol. 45, pp. 516-535, 2017.

[7] C. S. J. Doire, *et al*. *Single-Channel Online Enhancement of Speech Corrupted by Reverberation and Noise*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 3, pp. 572-587, Mar. 2017.

[8] M. Baelde, *et al*. *Influence of the attack conditions on countermeasures for Automatic Speaker Verification*. Soumis dans INTERSPEECH 2019.