On speaker verification from the neural network footprint of personalized acoustic models

Natalia Tomashenko, Salima Mdhaffar, Marc Tommasi, Yannick Estève, Jean-François Bonastre

Sur la vérification du locuteur à partir de traces d'exécution de modèles acoustiques personnalisés

Natalia Tomashenko[1], Salima Mdhaffar[1], Marc Tommasi[2], Yannick Estève[1], Jean-François Bonastre[1]
(1) LIA, Avignon Université, France; (2) Université de Lille, Inria, France
prénom.nom@univ-avignon.fr, prénom.nom@inria.fr

RÉSUMÉ

Les modèles acoustiques personnalisés sont construits par entraînement à partir de données provenant d'un locuteur unique en raffinant un modèle générique. Une question importante est de savoir si l'accès à ces modèles personnalisés permet facilement de construire une attaque permettant d'identifier le locuteur associé. Ce problème est important dans le contexte de l'apprentissage fédéré de modèles pour la reconnaissance de la parole où un modèle global est appris sur un serveur à partir des modifications des paramètres des modèles reçues de plusieurs clients. Nous proposons une méthode qui consiste à construire des empreintes de ces modèles à partir des traces de leur application sur un jeu de données fixe et indépendant que nous appelons *indicateur*. Grâce à ces empreintes, nous développons deux modèles d'attaques très efficaces qui visent à inférer l'identité du locuteur.

ABSTRACT

On speaker verification from the neural network footprint of personalized acoustic models

Speaker personalized acoustic models are obtained from a global model by updating its parameters using speaker's data. An important question is whether access to these personalized models allows to easily build an attack to identify the associated speaker. This problem is especially important in the context of federated learning of speech recognition acoustic models where a global model is learnt on the server using the updates received from multiple clients. We propose an approach to analyze information in neural network acoustic models based on a neural network footprint on the so-called *indicator* dataset. Using this method, we develop two very effective attack models that allow to infer speaker identity from the updated personalized models without access to the users' speech data.

MOTS-CLÉS : Vie privée, apprentissage fédéré, modèles acoustiques, modèles d'attaques, reconnaissance vocale, vérification du locuteur.

KEYWORDS: Privacy, federated learning, acoustic models, attack models, speech recognition, speaker verification.

1 Introduction

L'apprentissage fédéré (AF) pour la reconnaissance automatique de la parole (RAP) connaît une grande popularité dans plusieurs tâches et travaux (Cui *et al.*, 2021; Dimitriadis *et al.*, 2020; Guliani, 2021; Yu *et al.*, 2021; Tomashenko *et al.*, 2022a) [1]. Le respect de la vie privée est l'un des principaux

1. Cet article repose sur le travail présenté dans (Tomashenko *et al.*, 2022a).

défis de l'AF (Li *et al.*, 2020; Mothukuri *et al.*, 2021). Contrairement aux algorithmes d'apprentissage classiques qui utilisent un serveur contenant les données d'apprentissage, l'AF apprend sur des données stockées localement et communique uniquement les modifications (mises à jour). Ceci permet de protéger les données personnelles puisqu'elles ne sont ni stockées dans un serveur, ni partagées avec d'autres utilisateurs. Cependant, ces mises à jour peuvent encore contenir certaines informations sensibles (Geiping *et al.*, 2020; Carlini *et al.*, 2019). Des travaux récents ont montré que les modèles appris par l'AF sont vulnérables à différents types d'attaques (Truex *et al.*, 2019; Wang *et al.*, 2019). Les techniques pour améliorer la confidentialité dans un cadre de l'AF s'appuient principalement sur deux approches (Mothukuri *et al.*, 2021) : le calcul multipartite sécurisé (Bonawitz *et al.*, 2016) et la confidentialité différentielle (Dwork, 2006). Les méthodes de chiffrement (Smaragdis & Shashanka, 2007) comme le chiffrement entièrement homomorphe (Smaragdis & Shashanka, 2007) et le calcul multipartite sécurisé effectuent le calcul dans le domaine crypté. Ces méthodes sont trop coûteuses en termes de calcul. Les méthodes de confidentialité différentielle préservent la confidentialité en ajoutant du bruit aux paramètres des utilisateurs (Dwork, 2006). Cependant, ces solutions peuvent dégrader les performances d'apprentissage à cause de l'incertitude qu'elles introduisent dans les paramètres. Les méthodes alternatives à la protection de la confidentialité pour la parole comprennent les méthodes de suppression qui sont destinées à l'analyse des sons ambiants, et l'anonymisation (Tomashenko *et al.*, 2022b) qui vise à supprimer les informations personnelles identifiables dans le signal vocal en gardant tous les autres attributs. Ces méthodes de protection de la confidentialité peuvent être combinées et intégrées de manière hybride dans un cadre d'AF.

Malgré l'intérêt récent porté à l'AF pour la RAP et à d'autres tâches telles que le repérage de mots-clés (Leroy *et al.*, 2019), la reconnaissance des émotions (Latif *et al.*, 2020), et la vérification du locuteur (Granqvist *et al.*, 2020), il existe très peu d'études sur les attaques de confidentialité, dans un contexte d'AF, des modèles acoustiques (MA) pour la reconnaissance de la parole. Il a tout de même été montré récemment qu'il est possible d'extraire des informations sur le locuteur à partir des modifications portées sur les poids d'un modèle acoustique neuronal lors de sa personnalisation (Mdhaffar *et al.*, 2022).

Nos travaux s'inscrivent dans le cadre de ces attaques : nous étudions les informations propres au locuteur qui peuvent être extraites à partir de modèles acoustiques personnalisés mis à jour localement. Nous explorons différentes modèles d'attaques qui opèrent directement sur les paramètres du modèle mis à jour sans avoir accès aux données réelles de l'utilisateur. L'idée principale des méthodes proposées est d'utiliser un jeu de données externe (*indicateur*) pour analyser l'empreinte des modèles acoustiques sur ces données. Une autre contribution importante de ce travail concerne l'analyse des informations sur le locuteur représentées dans les modèles acoustiques neuronaux adaptés.

2 Apprentissage fédéré pour les modèles acoustiques de RAP

Nous considérons un scénario classique d'apprentissage fédéré où un modèle acoustique neuronal global est entraîné sur un serveur à l'aide des données stockées localement sur plusieurs dispositifs distants (Li *et al.*, 2020). L'apprentissage du modèle global est effectué sous la contrainte que les données vocales d'apprentissage sont stockées et traitées localement sur les dispositifs des utilisateurs (clients). Seules les mises à jour du modèle sont transmises au serveur à partir de chaque client. Le modèle global est appris sur le serveur en fonction des mises à jour reçues de plusieurs clients. La Figure 1 illustre l'AF dans un réseau distribué de clients. Tout d'abord, le modèle acoustique initial de

reconnaissance de la parole W_g est distribué à l'ensemble des systèmes des N utilisateurs (locuteurs). Ensuite, le modèle global initial est exécuté sur chaque dispositif utilisateur s_i ($i \in 1..N$) et mis à jour localement sur les données privées de l'utilisateur. Les modèles mis à jour W_{s_i} sont ensuite transmis au serveur où ils sont agrégés pour obtenir un nouveau modèle global W_g^*. En général, les modèles personnalisés mis à jour sont agrégés en utilisant la moyenne fédérée et ses variations (McMahan *et al.*, 2017). Ensuite, le modèle global mis à jour W_g^* est partagé avec les clients. Ce processus est répété plusieurs fois jusqu'à la convergence du modèle ou en fixant un nombre d'itérations. L'utilité et l'efficacité de l'apprentissage des modèles entraînés dans le cadre d'AF ont été étudiées avec succès dans des travaux récents (Cui *et al.*, 2021; Dimitriadis *et al.*, 2020; Guliani, 2021; Yu *et al.*, 2021). Dans cette étude, nous nous concentrons sur l'aspect de la protection de la vie privée.

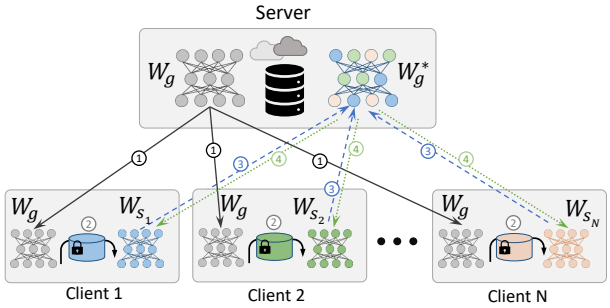

FIGURE 1 – Apprentissage fédéré dans un réseau distribué de clients : 1) Téléchargement du modèle global W_g par les clients. 2) Adaptation au locuteur de W_g sur les appareils locaux en utilisant les données privées de l'utilisateur. 3) Collecte et agrégation de plusieurs modèles personnalisés W_{s_1},...,W_{s_N} sur le serveur. 4) Partage du modèle résultant W_g^* avec les différents clients.

3 Modèles d'attaques

Dans cette section, nous décrivons le scénario de protection de la confidentialité et nous présentons deux modèles d'attaques.

3.1 Scénario de préservation de la confidentialité

La préservation de la confidentialité est formulée comme un jeu entre des *utilisateurs* qui partagent certaines données et des *attaquants* qui accèdent à ces données ou à des données dérivées de celles-ci et visent à déduire des informations sur les utilisateurs (Tomashenko *et al.*, 2020). Les attaquants visent à attaquer les utilisateurs en utilisant les informations détenues par le serveur. Ils peuvent avoir accès à des modèles personnalisés. Dans ce travail, nous supposons qu'un attaquant a accès aux données suivantes :

— Un modèle global initial W_g.

— Un modèle personnalisé W_s du locuteur cible (*target*) s qui est inscrit dans le système d'apprentissage fédéré. Le modèle personnalisé correspondant a été obtenu à partir du modèle

global W_g en ajustant par *fine-tuning* les poids de W_g à l'aide des données du locuteur. Nous considérons ce modèle comme *enrollment* pour un attaquant.

— Des modèles personnalisés des locuteurs non-cibles (*non-target*) et cibles (*target*) : W_{s_1},...,W_{s_N}. Nous appellerons ces modèles *test trial*.

L'objectif de l'attaquant est de réaliser une tâche de vérification automatique du locuteur (VAL) en utilisant le modèle de données d'inscription (*enrollment*) sous la forme de W_s et les données d'essai (*test trial*) de test sous la forme de modèles W_{s_1},...,W_{s_N}. Autrement dit, étant donné un modèle W_s correspondant au locuteur cible, la tâche consiste à identifier quels modèles parmi les W_{s_i} correspondent ou non au locuteur cible (Bonastre *et al.*, 2021).

3.2 Modèles d'attaques

Les deux approches proposées reposent sur l'hypothèse que nous pouvons capturer des informations sur l'identité du locuteur s à partir du modèle correspondant adapté au locuteur W_s et du modèle global W_g en comparant les sorties de ces deux modèles acoustiques neuronaux provenant des couches cachées h sur certaines données vocales. Nous appellerons ces données vocales *indicateur*. Ces données ne sont liées ni aux données de test ni aux données d'apprentissage des modèles.

Modèle d'attaque A1. La Figure 2 illustre la VAL pour le modèle d'attaque **A1** proposé dans ce papier. Ce modèle d'attaque comporte plusieurs étapes. On considère un ensemble d'énoncés (*utterances*) dans l'ensemble de données *indicateur* $\mathbb{I} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_J\}$; une séquence de vecteurs dans l'énoncé $\boldsymbol{u}_j$=$\{u_j^1, \ldots, u_j^{T_j}\}$; un ensemble de modèles personnalisés $\mathbb{W} = \{W_{s_1}, \ldots, W_{s_N}\}$; et un identifiant d'une couche cachée dans le MA global ou personnalisé représenté par h. Nous présentons ci-dessous les étapes pour le modèle d'attaque **A1**.

1. $\forall\, W_{s_i} \in \mathbb{W}, \forall\, \boldsymbol{u}_j \in \mathbb{I}$ nous calculons les valeurs d'activation de la couche h pour les paires de modèles : $W_{s_i}^h(\boldsymbol{u}_j) = \{w_{s_i,j}^{h,t}\}_{t=1}^{T_j}$ et $W_g^h(\boldsymbol{u}_j) = \{w_{g,j}^{h,t}\}_{t=1}^{T_j}$, et les différences par vecteur entre les sorties correspondantes :

$$\boldsymbol{\Delta}_{s_i}^h(\boldsymbol{u}_j) = \{\Delta_{s_i,j}^{h,t}\}_{t=1}^{T_j}, \quad \text{où} \quad \Delta_{s_i,j}^{h,t} = w_{s_i,j}^{h,t} - w_{g,j}^{h,t}, \quad t \in 1..T_j. \tag{1}$$

2. Pour chaque modèle personnalisé, nous calculons les vecteurs de moyenne et d'écart type pour $\Delta_{s_i,j}^{h,t}$ sur tous les vecteurs de parole dans les données *indicateur* $\mathbb{I}$:

$$\boldsymbol{\mu}_{s_i}^h = \frac{\sum_{j=1}^{I}\sum_{t=1}^{T_j}\Delta_{s_i,j}^{h,t}}{\sum_{j=1}^{I}T_j} \quad \text{et} \quad \boldsymbol{\sigma}_{s_i}^h = \left(\frac{\sum_{j=1}^{I}\sum_{t=1}^{T_j}(\Delta_{s_i,j}^{h,t} - \boldsymbol{\mu}_{s_i}^h)^2}{\sum_{j=1}^{I}T_j}\right)^{\frac{1}{2}}. \tag{2}$$

3. Pour une paire de modèles personnalisés W_{s_i} and W_{s_k}, nous calculons un score de similarité ρ pour la couche cachée h sur l'ensemble de données *indicateur* sur la base de la distance euclidienne normalisée $\boldsymbol{L}_2$ entre les paires de vecteurs correspondants pour les moyennes et les écarts types :

$$\rho(W_{s_i}^h, W_{s_k}^h) = \alpha_\mu \frac{\|\boldsymbol{\mu}_{s_i}^h - \boldsymbol{\mu}_{s_k}^h\|_2}{\|\boldsymbol{\mu}_{s_i}^h\|_2\|\boldsymbol{\mu}_{s_k}^h\|_2} + \alpha_\sigma \frac{\|\boldsymbol{\sigma}_{s_i}^h - \boldsymbol{\sigma}_{s_k}^h\|_2}{\|\boldsymbol{\sigma}_{s_i}^h\|_2\|\boldsymbol{\sigma}_{s_k}^h\|_2}, \tag{3}$$

où α_μ, α_σ sont des paramètres fixes dans toutes les expériences.

4. En utilisant les scores de similarité obtenues pour toutes les paires de matrices, nous pouvons effectuer une tâche de vérification du locuteur.

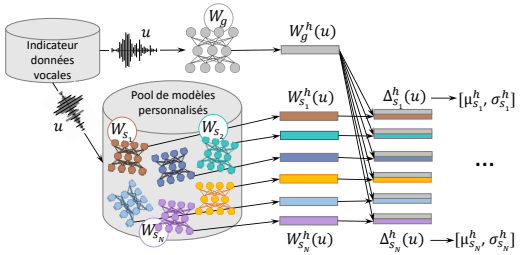

FIGURE 2 – Calcul des statistiques pour le modèle d'attaque **A1**.

Modèle d'attaque A2. Pour le deuxième modèle d'attaque, nous entraînons un modèle réseau neuronal (RN) comme indiqué dans la Figure 3. Ce modèle RN utilise des modèles personnalisés et le modèle globale ainsi que le jeu de données *indicateur* pour l'apprentissage. Il est entraîné à prédire l'identité d'un locuteur à partir du modèle personnalisé correspondant. Lorsque le modèle est entraîné, nous l'utilisons au moment de l'évaluation pour extraire les représentations (*embeddings*) du locuteur de manière similaire aux x-vecteurs (Snyder *et al.*, 2018). Le modèle est composé de deux parties (une partie fixe et une partie à entraîner). Les sorties de la partie fixe sont les séquences $\mathbf{\Delta}_{s_i}^h$ de vecteurs calculés sur les données *indicateur* comme défini dans la Formule (1). Pour chaque modèle personnalisé W_{s_i}, nous calculons $\mathbf{\Delta}_{s_i}^h$ pour tous les énoncés $\boldsymbol{u}$ du corpus *indicateur*. $\mathbf{\Delta}_{s_i}^h(\boldsymbol{u})$ sera utilisé comme entrée de la deuxième partie (entraînée) du RN, qui comprend plusieurs couches de RN de type *time delay neural network* (TDNN) (Peddinti *et al.*, 2015) et une couche de type *statistical pooling*.

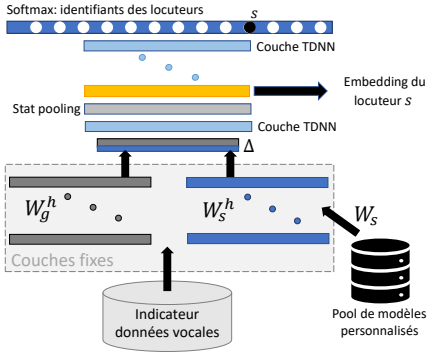

FIGURE 3 – Apprentissage d'un extracteur *d'embeddings* du locuteur pour le modèle d'attaque **A2**.

4 Résultats expérimentaux

4.1 Données

Les expériences ont été menées sur la partition *d'adaptation au locuteur* du corpus TED-LIUM 3 (Hernandez *et al.*, 2018). Ce corpus disponible publiquement contient les conférences TED qui

représentent 452 heures de données vocales en anglais provenant d'environ 2K locuteurs, 16kHz. Nous avons sélectionné trois ensembles de données à partir du corpus d'entraînement TED-LIUM 3 : *Train-G, Part-1, Part-2* avec des sous-ensembles de locuteurs disjoints comme indiqué dans le tableau 1. Le jeu de données *indicateur* a été utilisé pour entraîner les modèles d'attaques. Il est composé de 320 énoncés (*utterances*) sélectionnés parmi les 32 locuteurs des ensembles de données de test et de développement du corpus TED-LIUM 3. Les locuteurs du jeu de données *indicateur* sont disjoints des locuteurs de *Train-G, Part-1*, et *Part-2*. Pour chaque locuteur de l'ensemble de données *indicateur*, nous sélectionnons uniquement 10 énoncés. La taille totale du jeu de données *indicateur* est de 32 minutes. L'ensemble de données *Train-G* a été utilisé pour entraîner un modèle acoustique global initial W_g. *Part-1* et *Part-2* ont été utilisés pour obtenir deux ensembles de modèles personnalisés.

	Train-G	**Part-1**	**Part-2**	**Indicateur**
Durée, heures	200	86	73	0.5
Nombre de locuteurs	880	736	634	32
Nombre de modèles personnalisés	—	1300	1079	—

TABLE 1 – Statistiques de données

4.2 Modèles acoustiques pour la reconnaissance automatique de la parole

Les modèles acoustiques pour la RAP suivent une architecture neuronale de type TDNN (Peddinti *et al.*, 2015) et ont été entraînés en utilisant la boîte à outils de reconnaissance vocale Kaldi (Povey *et al.*, 2011). Les coefficients cepstraux de fréquence Mel (MFCC) à 40 dimensions, concaténés à des i-vecteurs à 100 dimensions, ont été utilisés comme entrée dans les réseaux de neurones. Chaque modèle comporte treize couches cachées de 512 dimensions, suivies d'une couche softmax dans laquelle 3664 états de triphonie ont été utilisés comme *target* [2]. Le modèle global initial W_g a été entraîné en utilisant le critère *lattice-free maximum mutual information* (LF-MMI) (Povey *et al.*, 2016). Les deux types de stratégies d'augmentation des données vocales ont été appliquées pour les données d'entraînement et d'adaptation : perturbation de la vitesse (avec des facteurs de 0,9, 1,0, 1,1) et perturbation du volume, comme dans (Peddinti *et al.*, 2015). Chaque modèle est composé d'environ 13.8 millions de paramètres. Le modèle global initial W_g a été entraîné sur le *Train-G*. Des modèles personnalisés W_{s_i} ont été obtenus en ajustant finement tous les paramètres de W_g sur les données des locuteurs de *Part-1* et *Part-2* comme décrit dans (Tomashenko *et al.*, 2022a). Pour tous les modèles personnalisés de locuteurs, nous utilisons approximativement la même quantité de données vocales pour effectuer le réglage fin (adaptation du locuteur ou *fine-tuning*) – environ 4 minutes par modèle. Pour la plupart des locuteurs (564 dans *Part-1*, 463 dans *Part-2*), nous avons obtenu deux modèles personnalisés différents (par locuteur) sur des sous-ensembles d'adaptation disjoints, pour les autres locuteurs, nous ne disposons de données d'adaptation que pour un seul modèle.

4.3 Modèles d'attaques

Nous étudions deux approches pour les modèles d'attaques : **A1** – une approche simple basée sur l'analyse statistique comparative des sorties du modèle de RN et le score de similarité associé entre

2. En suivant la notation de (Peddinti *et al.*, 2015), la configuration du modèle peut être décrite comme suit : {-1,0,1} $\times$ 6 couches ; {-3,0,3} $\times$ 7 couches.

les modèles personnalisés, et **A2** – une approche basée sur les RN. Pour les essais sur les cibles (*test target trials*), nous utilisons des comparaisons entre différents modèles personnalisés des mêmes locuteurs (564 dans le *Part-2*), et pour les essais sur les non-cibles (*test non-target trials*), nous avons sélectionné aléatoirement 10K paires de modèles de différents locuteurs (choisis au hasard parmi toutes les $1079 \times 1078/2$ comparaisons possibles) dans un ensemble de données correspondant.

Modèle d'attaque A1. Le premier modèle d'attaque a été appliqué comme décrit dans la section (3.2). Les paramètres α_μ, α_σ dans la formule (3) sont respectivement égaux à 1 et 10. Ce modèle a été évalué sur un jeu de données de modèles personnalisés. Le jeu de données *indicateur* est le même dans toutes les expériences.

Modèle d'attaque A2. Pour entraîner le modèle d'attaque **A2**, nous utilisons 1300 modèles de locuteurs personnalisés correspondant à 736 locuteurs uniques de *Part-1*. Lorsque nous avons appliqué la partie fixe de l'architecture présentée dans la Figure 3 au jeu de données *indicateur* de 32 minutes pour chaque modèle de locuteur dans *Part-1*, nous avons obtenu les données d'entraînement avec la quantité correspondant à environ 693h (32×1300). La partie entraînée du modèle neuronal, illustrée dans la Figure 3, a une topologie similaire à celle d'un extracteur de x-vecteur conventionnel (Snyder *et al.*, 2018). Cependant, l'extracteur de x-vecteur permet de prédire l'identité du locuteur pour le segment de discours donné alors que notre modèle proposé apprend à prédire l'identité du locuteur à partir de la partie W_s^h d'un modèle personnalisé du locuteur. Nous avons entraîné deux modèles d'attaques correspondant aux deux valeurs du paramètre $h \in \{1, 5\}$ – une couche cachée dans les MA neuronaux RAP à laquelle nous calculons les activations. Les valeurs h ont été choisies en fonction des résultats obtenus pour le modèle d'attaque **A1**. La dimension de sortie de la partie fixe est 512. La partie fixe est suivie par la partie entraînée qui consiste en sept couches TDNN cachées et une couche de regroupement statistique introduite après la cinquième couche TDNN. La sortie est une couche softmax avec les cibles (sorties) correspondant aux locuteurs dans le pool de modèles personnalisés de locuteurs (nombre de locuteurs uniques dans *Part-1*).

4.4 Résultats

Les modèles d'attaques ont été évalués en termes de *equal error rate* (EER) [3]. Les résultats du modèle d'attaque **A1** sont présentés dans la Figure 4. Les informations de locuteur peuvent être capturées pour toutes les valeurs de h avec un succès variable : EER varie de 0,86% (pour la première couche cachée) à 20,51% (pour la couche cachée supérieure). Pour analyser l'impact de chaque partie de la somme de la formule (3) sur les performances de la VAL, nous calculons séparément le score de similarité ρ en utilisant uniquement les moyennes ($\alpha_\sigma = 0$) ou uniquement les écarts types ($\alpha_\mu = 0$). L'impact de chaque terme de la somme change pour les différentes couches cachées. Lorsque nous utilisons uniquement les écarts-types, nous observons le plus faible EER sur la première couche. Dans le cas de l'utilisation des moyennes uniquement, la première couche est, au contraire, l'une des moins informatives pour la vérification du locuteur. Pour toutes les autres couches, la combinaison des moyennes et des écarts-types fournit des résultats supérieurs à ceux obtenus dans les cas où une seule de ces composantes est utilisée.

Nous choisissons deux valeurs $h \in \{1, 5\}$ qui montrent des résultats prometteurs pour le modèle **A1**, et nous utilisons les sorties correspondantes pour entraîner deux modèles d'attaques avec la

3. En désignant par $P_{\text{fa}}(\theta)$ et $P_{\text{miss}}(\theta)$ les taux de faux positifs (*false alarm*) et de faux négatifs (*miss rates*) au seuil θ, l'EER correspond au seuil θ_{EER} pour lequel les deux taux d'erreur de détection sont égaux : $\text{EER} = P_{\text{fa}}(\theta_{\text{EER}}) = P_{\text{miss}}(\theta_{\text{EER}})$.

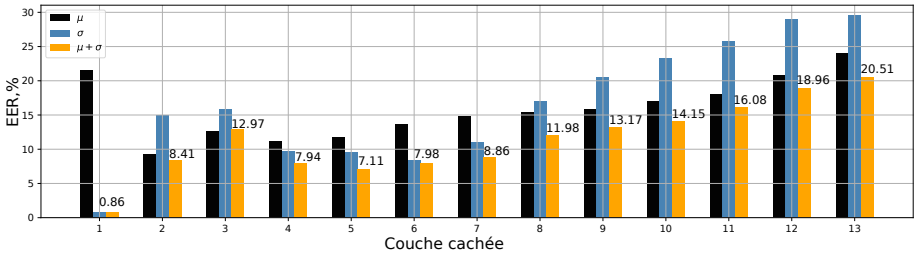

FIGURE 4 – EER, % pour le modèle d'attaque **A1** en fonction de la couche cachée h, évalué sur *Part-2*. $\mu + \sigma$ – les moyennes et les écarts types ont été utilisés pour calculer le score de similarité. ρ; μ – uniquement les moyennes ; et σ – uniquement les écarts types ont été utilisés.

configuration **A2**. Les résultats comparatifs des deux modèles d'attaques sont présentés dans le tableau 2. Pour $h = 5$, le deuxième modèle d'attaque fournit une amélioration significative des performances par rapport au premier et réduit l'EER de 7% à 2%. Pour $h = 1$, nous n'avons pu obtenir aucune amélioration en entraînant un modèle d'attaque basé sur un réseau de neurones : les résultats pour **A2** dans ce cas sont moins bons par rapport à l'approche simple **A1**.

Modèle d'attaque	**h=1**	**h=5**
A1	**0.86**	7.11
A2	12.31	**1.94**

TABLE 2 – EER, % évalué sur *Part-2*, h - indicateur d'une couche cachée

5 Conclusions

Dans cette étude, nous nous sommes concentrés sur le problème de la protection de la vie privée pour les modèles acoustique de RAP construits dans un cadre d'apprentissage fédéré. Nous avons exploré dans quelle mesure ces modèles de RAP sont vulnérables aux attaques contre la confidentialité. Nous avons développé deux modèles d'attaques qui visent à déduire l'identité du locuteur à partir des modèles personnalisés mis à jour localement sans avoir accès aux données vocales des locuteurs cibles. Un modèle d'attaque est basé sur le score de similarité proposé entre les modèles acoustiques personnalisés, calculé sur un ensemble de données *indicateur* externe, et un autre est un modèle neuronal. Nous avons démontré sur le corpus TED-LIUM 3 que les deux modèles d'attaque sont très efficaces et peuvent fournir un EER d'environ 1% pour le modèle d'attaque simple **A1** et 2% pour le modèle d'attaque neuronal **A2**. Une autre contribution importante de ce travail est la découverte que la première couche des modèles acoustiques personnalisés contient une grande quantité d'informations sur le locuteur qui sont principalement contenues dans les valeurs de déviation standard calculées sur les données *indicateur*. Cette propriété intéressante des modèles acoustiques neuronaux personnalisés ouvre de nouvelles perspectives également pour la VAL. Dans des travaux futurs, nous prévoyons de l'utiliser pour développer un système VAL efficace.

Remerciements

Ces travaux ont été financé par les projets VoicePersonae (ANR-18-JSTS-0001), DEEP-PRIVACY (ANR18-CE23-0018) et programme de Recherche et d'Innovation Horizon 2020 (Marie Skłodowska-Curie grant, No 101007666). Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2021-AD011013331 attribuée par GENCI.

Références

BONASTRE J.-F. *et al.* (2021). Benchmarking and challenges in security and privacy for voice biometrics. In *Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication*, p. 52–56.

BONAWITZ K., IVANOV V., KREUTER B. *et al.* (2016). Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv :1611.04482*.

CARLINI N., LIU C., ERLINGSSON Ú., KOS J. & SONG D. (2019). The secret sharer : Evaluating and testing unintended memorization in neural networks. p. 267–284.

CUI X. *et al.* (2021). Federated acoustic modeling for automatic speech recognition. In *ICASSP*.

DIMITRIADIS D. *et al.* (2020). A federated approach in training acoustic models. p. 981–985.

DWORK C. (2006). Differential privacy.

GEIPING J. *et al.* (2020). Inverting gradients–how easy is it to break privacy in federated learning ?

GRANQVIST F. *et al.* (2020). Improving on-device speaker verification using federated learning with privacy.

GULIANI D. A. (2021). Training speech recognition models with federated learning : A quality/cost framework.

HERNANDEZ F., NGUYEN V., GHANNAY S., TOMASHENKO N. *et al.* (2018). TED-LIUM 3 : twice as much data and corpus repartition for experiments on speaker adaptation. p. 198–208.

LATIF S. *et al.* (2020). Federated learning for speech emotion recognition applications. p. 341–342.

LEROY D. *et al.* (2019). Federated learning for keyword spotting. p. 6341–6345.

LI T., SAHU A. K., TALWALKAR A. & SMITH V. (2020). Federated learning : Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, **37**(3), 50–60.

MCMAHAN B., MOORE E., RAMAGE D. *et al.* (2017). Communication-efficient learning of deep networks from decentralized data. p. 1273–1282.

MDHAFFAR S. *et al.* (2022). Retrieving speaker information from personalized acoustic models for speech recognition. In *ICASSP*.

MOTHUKURI V., PARIZI R. M., POURIYEH S., HUANG Y. *et al.* (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, **115**, 619–640.

PEDDINTI V., POVEY D. & KHUDANPUR S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts.

POVEY D., GHOSHAL A., BOULIANNE G. *et al.* (2011). The Kaldi speech recognition toolkit.

POVEY D., PEDDINTI V., GALVEZ D., GHAHREMANI P., MANOHAR V., NA X. *et al.* (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. p. 2751–2755.

SMARAGDIS P. & SHASHANKA M. (2007). A framework for secure speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, **15**(4), 1404–1413.

SNYDER D., GARCIA-ROMERO D., SELL G., POVEY D. & KHUDANPUR S. (2018). X-vectors : Robust DNN embeddings for speaker recognition. p. 5329–5333.

TOMASHENKO N., MDHAFFAR S., TOMMASI M., ESTÈVE Y. & BONASTRE J.-F. (2022a). Privacy attacks for automatic speech recognition acoustic models in a federated learning framework. *ICASSP*.

TOMASHENKO N., SRIVASTAVA B. M. L., WANG X., VINCENT E., NAUTSCH A., YAMAGISHI J., EVANS N. *et al.* (2020). Introducing the VoicePrivacy initiative. p. 1693–1697.

TOMASHENKO N., WANG X., VINCENT E. *et al.* (2022b). The VoicePrivacy 2020 Challenge : Results and findings. *Computer Speech & Language*, **74**, 101362.

TRUEX S., LIU L., GURSOY M. E., YU L. & WEI W. (2019). Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*.

WANG Z., SONG M., ZHANG Z., SONG Y., WANG Q. & QI H. (2019). Beyond inferring class representatives : User-level privacy leakage from federated learning. p. 2512–2520.

YU W., FREIWALD J., TEWES S., HUENNEMEYER F. & KOLOSSA D. (2021). Federated learning in ASR : Not as easy as you think. *arXiv preprint arXiv :2109.15108*.