Création automatique de classes de signatures manuscrites pour l'authentification en ligne

Nicolas Ragot, J. Fortune, P. M'Bongo, N. Vincent, H. Cardot

Création automatique de classes de signatures manuscrites pour l'authentification en ligne

Nicolas Ragot[1] – Julie Fortune[1,2] – Paul M'Bongo[1] – Nicole Vincent[3] – Hubert Cardot[1]

[1] Laboratoire d'Informatique, Université François Rabelais de Tours
64 av. Jean Portalis, 37200, Tours, France

[2] Atos Worldline
19 rue de la Valle Maillard, 41000 Blois, France

[3] Laboratoire SIP / CRIP5, Université René Descartes
45 rue des Saints-Pères, 75270 Paris Cedex 06, France

[1,2] {nicolas.ragot, julie.fortune, hubert.cardot}@univ-tours.fr,
[3] nicole.vincent@math-info.univ-paris5.fr

Résumé *: Nous nous intéressons dans ce papier à l'optimisation d'un système d'authentification par signature manuscrite. Celui-ci est basé sur une approche Coarse To Fine et utilise l'algorithme Dynamic Time Warping ainsi qu'un seuil de décision global pour accepter ou rejeter un signataire. L'optimisation proposée réside dans l'utilisation d'un algorithme de classification non supervisée afin de déterminer automatiquement des classes de signatures. Pour chacune des classes, un seuil de décision spécifique est établi. Dans ces travaux, nous nous sommes plus particulièrement attaché à étudier l'impact de la classification sur les performance. Les résultats expérimentaux sur la base SVC montrent que l'on peut améliorer les performances en diminuant le taux d'erreur égale de 14,4%. Cependant la sensibilité de la classification est très grande et la notion de classe unique pour un signataire semble trop restrictive.*

Mots-clés *:* biométrie, signature manuscrite, authentification, en ligne, classification non supervisée, DTW

1 Introduction

A la différence des problèmes d'identification (par exemple de scripteurs), l'authentification par signature manuscrite en ligne est un problème de reconnaissance de formes « à une classe ». Les paramètres du système qui déterminent l'acceptation de la signature ou son rejet sont donc particulièrement sensibles et, dans cadre applicatif réel, ils peuvent souffrir d'un manque de généralisation. Dans ce contexte, il peut être intéressant de se rapprocher des recherches effectuées dans la communauté de l'écrit et qui ont montré qu'il existe différents types de scripteurs [CRET 95, NOS 99, SER 02, BEN 05, SID 07]. Nous pouvons alors faire l'hypothèse raisonnable qu'il existe également différents types, ou classes, de signataires (ou de signatures) et que l'on va pouvoir adapter les paramètres du système d'authentification, pour un signataire, en fonction de la classe de signature à laquelle il appartient et ainsi améliorer les performances.

Bien qu'il existe des travaux, notamment de graphologie [SED 02], qui doivent permettre de déterminer différentes particularités dans les signatures et donc des classes associées, cette méthode reposant sur les connaissances d'un expert n'est pas sans poser de problèmes : elle contraint à l'utilisation de descripteurs les plus proches possibles de ceux utilisés par l'expert, alors mêmes que ceux-ci ne sont pas forcément les plus intéressants du point de vue du système d'authentification ; cette connaissance peut être coûteuse à modéliser et à extraire du signal, ce qui n'est pas forcément compatible avec des terminaux d'authentification légers ; enfin, ces experts en graphologie sont souvent spécialisés sur des signatures d'une origine géographique bien précise (française, anglo-saxone, asiatique, etc.) rendant l'expertise inutilisable en dehors de son domaine de compétence. Il parait donc bien plus avantageux d'essayer de déterminer automatiquement les classes de signatures à partir d'échantillons disponibles. Rien n'empêchera par la suite d'enrichir ou de modifier cette classification si de nouvelles données sont disponibles.

Dans cette étude, nous nous sommes donc tournés vers l'utilisation d'une classification non supervisée pour créer automatiquement des classes de signatures. Nous avons plus particulièrement étudié l'impact de la classification sur un prototype d'authentification léger - basé sur l'algorithme *Dynamic Time Warping* (DTW) [JAI 02, OHI 00], notamment en fonction du nombre de classes, de l'espace de représentation ainsi que de la méthode de classification utilisée.

Dans la suite, la section 2 décrit le système d'authentification existant sur lequel nous nous sommes appuyés pour conduire notre étude. La section 3 présente ensuite le processus utilisé pour déterminer automatiquement des classes de signatures et utiliser celles-ci pour améliorer les performances du système

d'authentification. Enfin, les résultats expérimentaux sont analysés dans la section 4.

2 Système d'authentification

Notre étude se base sur l'utilisation d'un système d'authentification conçu lors de travaux précédents [WIR 05a, WIR 05b]. Celui-ci nécessite deux étapes. Lors de la première, la phase d'enrôlement (cf. section 2.2), la personne à authentifier s'enregistre en fournissant quelques signatures pour que le système crée son profil (ensemble de signatures de référence). La deuxième étape (cf. section 2.3) est l'authentification à proprement parler : une personne - la personne préalablement enregistrée ou une autre qui attaque le système – soumet une signature qui est authentifiée ou non. Dans tous les cas, l'acquisition des signatures et les prétraitements sont identiques (cf. section 2.1).

2.1 Acquisition des signatures

L'acquisition des signatures s'effectue sur un TabletPC, une tablette graphique ou un PDA. Afin de conserver le maximum de compatibilité, seuls les coordonnées de chacun des points de la signature ainsi que les posés et levés de stylet sont conservés. Ensuite, une normalisation est effectuée sur les signatures. Cette normalisation consiste en : une rotation selon l'axe d'inertie [LEJ 01, WIR 05b] ; une homothétie pour que toutes les signatures aient la même taille tout en conservant leurs proportions ; et enfin en une translation pour centrer la signature par rapport au repère. Les figures 1 et 2 illustrent le résultat de cette transformation.

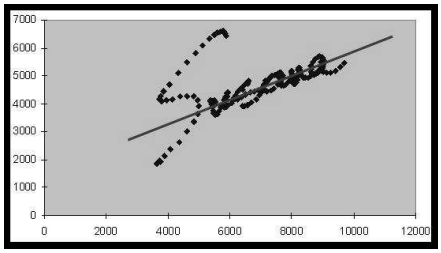

FIG. 1 – Signature originale avec son axe d'inertie.

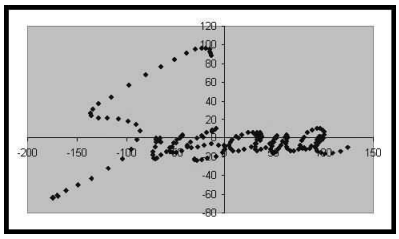

FIG. 2 – Signature normalisée.

Le dernier traitement effectué consiste à ne conserver que certains points significatifs de la signature. L'intérêt est ici multiple. Cela permet de réduire la taille de la signature et donc d'accélérer les traitements et cela permet également de diminuer l'espace de stockage nécessaire pour le profil. Ces deux propriétés sont particulièrement importantes pour une application devant fonctionner dans un cadre réel d'utilisation. En effet, la CNIL interdit les bases de données biométriques. Pour chaque utilisateur, ses données biométriques doivent donc être stockées sur une carte à puce personnelle. De plus, pour assurer la sécurité du système, aucune information ne doit pouvoir être récupérée de cette carte. C'est la raison pour laquelle nous stockons le profil d'un utilisateur directement sur une carte à puce. L'authentification s'effectue également sur cette carte. Ainsi le système est plus difficilement attaquable, la contrepartie étant l'utilisation de ressources système limitées. Un deuxième avantage, et non des moindres, à la sélection de points est que si ceux-ci sont bien choisis, les performances du système peuvent être accrues. Ici, après des études comparatives sur plusieurs méthodes de sélection de points, nous avons choisis de ne conserver que les points de vitesse minimale [WIR 05b].

2.2 Enrôlement

Cette étape a pour objectif de créer le profil d'un utilisateur. Après une phase d'entrainement, nécessaire à la prise en main du périphérique d'acquisition, l'utilisateur fournit 5 signatures au système. En cas de problème lors de l'acquisition, l'utilisateur peut annuler une saisie et la recommencer. Le système effectue également une vérification pour éviter tout problème lors de la création du profil de l'utilisateur. Cette vérification consiste à demander 5 nouvelles signatures au signataire. Si parmi ces 5 signatures de vérification l'une d'entre elle est trop dissemblable des 5 signatures de références, l'utilisateur doit recommencer la procédure d'enrôlement. La dissemblance entre 2 signatures est évaluée en considérant le temps total et la longueur totale de la signature, exactement de la même façon que lors de la première étape de notre processus d'authentification (cf. section 2.3.1).

Une fois les 5 signatures valides acquises, elles subissent les traitements ci-dessus (cf. section 2.1) puis sont stockées comme signatures de référence en attente d'une future authentification.

2.3 Authentification

Lors de cette phase, un utilisateur souhaitant s'authentifier signe sur le périphérique d'acquisition. La signature test ainsi obtenue, après avoir subi les prétraitements décrits section 2.1, va être comparée aux signatures de référence afin de déterminer si le signataire est bien celui qu'il prétend être. Pour cela, la comparaison s'effectue en deux étapes, selon une approche *Coarse To Fine* [WIR 05a, WIR 05b]. L'approche *Coarse* (cf. section 2.3.1) permet d'éliminer directement les signatures très différentes des signatures de référence. Ces attaques supposées seront donc rejetées sans passer par l'étape *Fine* (cf. section 2.3.2) qui effectue une comparaison plus précise mais aussi plus coûteuse en temps de calculs. Ce principe permet donc d'accélérer les traitements lors de l'authentification, l'étape *Fine* n'étant utilisée que si nécessaire.

2.3.1 Etape *Coarse*

Cette première étape élimine une signature trop dissemblable des signatures de référence. Elle se base pour cela sur une comparaison des signatures représentées par des caractéristiques globales et stables [WIR 05b] : la longueur totale (C^1) et le temps total (C^2) de la signature. Une signature test est acceptée

– et l'étape *Fine* est utilisée - si la condition suivante est vérifiée pour les deux caractéristiques C^j ($j=1,2$) :

$$\delta \times min_i(C^j_i) \leq C^j \leq \lambda \times max_i(C^j_i), \quad (1)$$

avec C^j la valeur de la caractéristique pour la signature test, C^j_i la valeur de cette même caractéristique pour la signature de référence i ($i=1,\ldots,5$), δ et λ des coefficients fixés expérimentalement à *0,6* et *1,4* lors d'études précédentes et à partir d'une base privée [WIR 05b].

2.3.2 Etape *Fine*

Cette seconde étape effectue une nouvelle comparaison entre la signature testée et les 5 signatures de référence, mais cette fois en utilisant une mesure de distance plus précise. Dans notre cas, cette distance est une variante du DTW [JAI 02, OHI 00] adaptée pour la comparaison de signatures manuscrites en ligne [WIR 05a, WIR 05b]. Plusieurs métriques peuvent être associées au DTW : la distance spatiale, la distance temporelle, la distance curviligne [WIR 05a]. Ici, nous utilisons la plus performante des trois, à savoir la distance spatiale qui correspond à la distance euclidienne entre deux points mis en correspondance et projetés dans le même repère.

Pour qu'une signature test soit acceptée, il faut que la distance entre elle et au moins une des signatures de référence soit inférieure au seuil μ. Ce seuil global est déterminé lors d'une phase préalable d'apprentissage du système. Pendant celle-ci, une base de validation est utilisée pour simuler le processus d'authentification et obtenir les performances du système : TFA (Taux de Faux Acceptés, ou *False Acceptance Rate* – FAR) et TFR (Taux de Faux Rejet ou *False Rejection Rate* - FRR, i.e. le taux de signatures authentiques rejetées). Différentes valeurs de seuils sont ensuite testées de façon quasi exhaustive (par incrément fixe) jusqu'à obtenir le TEE (Taux d'Erreur Egale, ou *Equal Error Rate* – EER) sur cette base de validation (cf. Fig. 3).

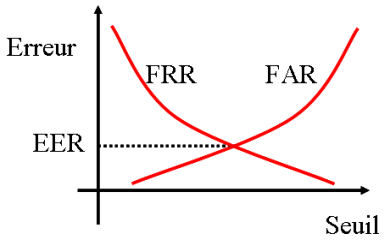

FIG. 3 – Définition du TEE (ou EER).

3 Création automatique de classes de signatures

Un des inconvénients du système précédent réside essentiellement dans l'utilisation du seuil de décision global μ qui est unique quels que soient les utilisateurs du système. C'est pourquoi nous avons souhaité étudier l'impact de la création de classes de signatures sur les performances du système. Dans la section 3.1 nous décrivons le mécanisme utilisé pour créer les classes de signatures. Puis, dans la section 3.2, nous décrivons les changements que cela induit sur le paramétrage du système. Enfin, la section 3.3 décrit le nouveau fonctionnement lors de l'authentification.

3.1 Création de classes de signatures

L'important pour la création de classes de signatures est avant tout d'arriver à déterminer une classification qui soit stable pour chacun des utilisateurs du système. Idéalement, on souhaite qu'un signataire n'appartienne qu'à une seule classe, c'est-à-dire que toutes ses signatures (de références et à authentifier) appartiennent à la même classe. C'est l'hypothèse forte sur laquelle repose cette première étude et nous verrons comment le nombre de classes (K), l'espace de représentation ainsi que l'algorithme utilisé pour déterminer les classes influent sur la classification ainsi que sur les performances du système.

Pour obtenir les K classes, nous utilisons soit les *K-means* [MCQ 67], soit les *Fuzzy C-means* [BEZ 81], sur l'ensemble des signatures de référence dont on dispose lors de la phase d'apprentissage. Ensuite, nous définissons la classe d'un signataire comme celle dans laquelle se retrouvent la majorité de ses signatures de références (appartenance stricte).

En ce qui concerne l'espace de représentation, nous avons choisi dans un premier temps de travailler sur des caractéristiques globales[1]. Parmi celles-ci, le temps total de la signature ainsi que sa longueur totale sont reconnues comme étant des caractéristiques stables pour un signataire (cf. étape *Coarse*, section 2.3.1). Cependant, pour aller plus loin, nous avons également travaillé sur un ensemble plus vaste de 12 caractéristiques, décrivant la forme et la dynamique des signatures. Ces caractéristiques sont :

- Longueur totale (longueur Totale) ;
- Rapport ente les déplacements vers la gauche et vers la droite (rapDepHor) ;
- Rapport ente les déplacements vers le haut et vers le bas (rapDepVer) ;
- Rapport entre les déplacements horizontaux et verticaux (rapDepXY) ;
- Distance entre le premier et le dernier point (distPremDer) ;
- Déplacement horitzontal moyen (depHorMoy);
- Déplacement vertical moyen (depVerMoy) ;
- Angle entre l'horizontale et la droite joignant le premier et le dernier point (angleSign) ;
- Temps total (tpsTotal) ;
- Accélération moyenne (accMoy) ;
- Vitesse moyenne verticale (vtsMoyVert) ;
- Vitesse moyenne horitzontale (vtsMoyHor).

Afin de déterminer les corrélations entre ces caractéristiques nous avons effectué une analyse en composantes principales (ACP). Le cercle des corrélations correspondant est donné dans la figure 4. Sur celle-ci on constate que la longueur totale et le temps total restent assez fortement corrélés. On se doute alors que l'algorithme de classification ne pourra s'appuyer que sur un pouvoir de représentation de l'espace de description assez limité.

[1] Des caractéristiques locales pourraient également être utilisées mais cela fera partie d'une prochaine étude.

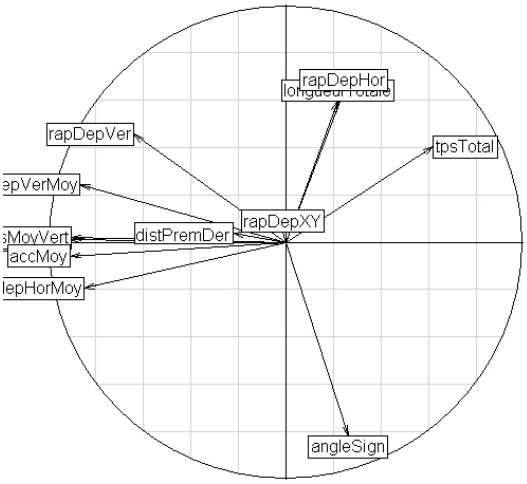

FIG. 4 – Corrélation entre les caractéristiques testées pour la création de classes de signatures.

3.2 Optimisation du système grâce aux classes de signatures

Une fois les classes de signatures obtenues à partir des signatures de référence de la base d'apprentissage, nous utilisons les signatures de la base de validation pour définir un seuil de décision μ_k pour chacune de ces classes k ($k=1,\ldots K$). Le principe est le même que pour le système initial (cf. section 2.3.2) sauf que l'on cherche les seuils qui permettent d'obtenir le TEE sur chaque classe. Ainsi, chaque signature de la base de validation, est testée par rapport à chaque signataire de la base d'apprentissage, c'est-à-dire qu'elle est comparée à chacune des signatures de référence correspondantes. Lors de l'étape *Fine*, le seuil utilisé pour accepter ou rejeter la signature est le seuil μ_k correspondant à la classe des signatures de référence en cours. Le TFA ou le TFR de la classe sont alors mis à jour en conséquence. Lorsque toute la base de validation a été utilisée, le seuil est modifié puis les TFA et TFR de chaque classe sont modifiés jusqu'à obtenir le TEE sur chaque classe.

3.3 Authentification à partir de classes de signatures

Dans le mode de fonctionnement réel, lors de la phase d'enrôlement, le profil du signataire est toujours constitué de ses 5 signatures de références. Pour ce qui concerne le seuil de décision à appliquer lors de l'étape *Fine* des futures authentifications, on détermine dans quelle classe se trouvent la majorité des signatures de référence de ce signataire et on utilisera alors le seuil μ_k correspondant défini lors de la phase d'apprentissage.

4 Résultats expérimentaux

4.1 Protocole de test

Pour évaluer l'impact de la création de classes sur notre système d'authentification par signature manuscrite en ligne, nous avons utilisé la base SVC (Signature Verification Competition) [SVC 04, YEU 04]. Cette base est constituée de 40 signataires. Pour chaque signataire, 20 signatures sont authentiques et 20 autres sont des faux expérimentés, réalisés par des personnes ayant accès à une vidéo de la personne en train de signer. Dans notre travail d'expérimentation, nous ne nous intéressons dans un premier temps qu'aux performances sur des faux aléatoires[2] et nous n'utilisons donc que 40x20 signatures. Nous sommes également conscients que cette base a une taille limitée. Cependant, il s'agit de l'une des rares bases disponibles et utilisée comme benchmark de référence. Il faut également noter que nous n'avons pas de connaissances a priori sur la représentativité de cette base quant aux différents styles de signatures et donc quant au nombre de classes. En effet, l'origine des contributeurs n'est pas connue. De plus, ceux-ci n'ont pas signé avec leur signature habituelle mais en ont inventé une pour l'occasion.

Nous effectuons nos tests en *leave-one-out*. 39 signataires sont utilisés pour l'apprentissage et la validation : les cinq premières signatures de chacun des 39 signataires sont utilisées comme signatures de référence afin de simuler la phase d'enrôlement et effectuer la recherche de classes ; les 15*39 signatures restantes sont utilisées comme base de validation pour rechercher les seuils. Enfin, le dernier signataire est utilisé lors de la phase de test pour évaluer les performances du système en phase de généralisation : les 5 premières signatures du signataire simulent l'enrôlement ; les 15 suivantes servent pour évaluer le TFR ; enfin, les 15*39 signatures de la base de validation[3] sont utilisées comme attaques. Cette phase d'apprentissage/test est répétée de façon à ce que chaque signataire serve exactement une fois pour le test ce qui permet d'obtenir les TFA et TFR moyens du système.

4.2 Résultats du système initial

Le tableau 1 donne les TFA et TFR moyens obtenus en *leave-one-out* sur la base SVC avec notre système initial, c'est-à-dire sans classes de signatures. Il s'agit donc de notre référence pour évaluer l'impact de l'utilisation de classes de signatures. Le TEE correspondant est obtenu en prenant la moyenne du TFA et du TFR.

	TFA (%)	**TFR (%)**	**TEE (%)**
Sans classe	1,88	2,00	1,94

TAB. 1 – Performances du système initial.

Notre objectif étant avant tout d'étudier l'apport de la classification, nous ne comparons pas notre approche avec d'autres. En outre, la comparaison est souvent difficile. En effet, beaucoup utilisent leur propre base de signatures. De plus, les protocoles expérimentaux diffèrent souvent de façon assez significative (nombre de signatures lors de l'enrôlement, pas de *leave-one-out*, etc.). Ainsi, même sur la base SVC, il est difficile de fournir des données comparatives. Pour information, la

[2] Les faux expérimentés semblent moins intéresser les industriels qui estiment que, dans la réalité, il existe peu de chances que quelqu'un fasse un faux expérimenté en ligne (les faussaires auraient besoin d'une vidéo précise du signataire).

[3] Bien que ces signatures aient servi pour déterminer les seuils en phase de validation, elles n'ont pas servies par rapport au signataire testé en phase de généralisation. On peut donc les considérer comme des faux aléatoires quelconques et le biais éventuel devrait être très réduit.

meilleure approche sur cette base avec les faux expérimentés obtient un TEE de 2,8%. Ensuite, les taux passent à 4,4% puis 5% pour les deuxièmes et troisièmes meilleures approches [KHO 05].

4.3 Résultats avec la classification basée sur le temps total et la longueur totale

Cette première série de résultats donne les performances obtenues selon le protocole défini plus haut en utilisant les caractéristiques de temps total et de longueur totale pour déterminer des classes de signatures avec les *K-means*. Le tableau 2 montre les résultats pour différents nombre de classes (*K*). Les taux donnés représentent une moyenne sur 10 exécutions successives du protocole complet. Nous fournissons également l'écart type correspondant (EcA et EcR).

K	TFA	EcA	TFR	EcR	TEE
2	1,47	0,02	2,20	0,6	1,84
3	1,48	0,04	2,33	0,00	1,92
4	1,58	0,03	2,17	0,00	1,87
5	1,61	0,09	2,17	0,00	1,89
6	1,56	0,02	2,22	0,18	1,89
7	1,64	0,12	2,06	0,07	1,85
8	1,65	0,11	2,89	0,77	2,27
9	1,8	0,06	3,06	0,6	2,43
10	1,78	0,17	3,22	0,53	2,5

TAB. 2 – Performances avec les *K-means* en utilisant le temps total et la longueur totale.

Dans ce tableau, on constate que jusqu'à 7 classes environ, les résultats sont globalement meilleurs que ceux du système initial – jusqu'à 5,2% de mieux sur le TEE. Au-delà, les performances se dégradent (y compris au-delà de 10 classes). On notera particulièrement que le TFA est plus faible, notamment avec un faible nombre de classes, puis qu'il ré-augmente à partir de *K*=7. On peut donc en déduire que notre spécialisation des seuils est bien efficace (jusqu'à un certain point) puisque les attaques sont mieux décelées. Cependant, la contrepartie est que le TFR tend à augmenter avec *K*. Une explication est que pour un même signataire, toutes ses signatures ne correspondent pas en réalité à la même classe. Ces dernières sont donc plus souvent rejetées lors de l'authentification. Ce point est confirmé en observant la répartition par classes des signatures de chaque signataire. Nous pouvons donc en déduire que : soit notre hypothèse de départ – toutes les signatures d'un signataire appartiennent à la même classe - est trop forte ; soit que les deux caractéristiques choisies et les *K-means* ne sont pas à même d'effectuer une bonne classification des signatures. On notera également que l'écart-type est assez faible pour 3, 4, 5 et 7 classes alors qu'il est plutôt élevé dans les autres cas, ce qui signifie pour ceux-ci que la classification n'est pas réellement stable d'une fois sur l'autre.

4.4 Résultats avec la classification et différentes caractéristiques

Le tableau 3 présente les résultats obtenus avec d'autres jeux de caractéristiques. Nous avons dans un premier temps sélectionné la longueur totale et l'accélération moyenne puisque celles-ci apparaissent comme peu corrélées (cf. figure 4). Pour les mêmes raisons, nous avons également sélectionné le temps total et le rapport entre les déplacements haut et bas. Enfin, nous avons évalué les résultats en utilisant les caractéristiques synthétiques correspondant aux 2 et 3 premiers axes principaux qui apportent respectivement 70 et 80% de l'inertie.[4] Pour toutes ces expérimentations, les résultats sont toujours la moyenne et l'écart type sur 10 exécutions. Nous ne reportons que les résultats pour la valeur de *K* qui donne les meilleures performances.

Caracs	*K*	TFA	EcA	TFR	EcR	TEE
longTotale accMoy	2	1,42	0,04	3,04	0,17	2,23
tempsTotal rapDepVer	2	1,84	0,07	1,87	0,21	1,86
ACP x2	3	1,56	0,06	2,13	0,06	1,85
ACP x3	2	1,85	0,00	2,00	0,00	1,93

TAB. 3 – Performances avec *K* classes et en utilisant différentes caractéristiques.

La première constatation que nous pouvons faire est la même que précédemment : les meilleures performances sont obtenues pour un faible nombre de classes, ici 2 ou 3. Au-delà, le TEE est toujours moins bon que celui du système initial. Le second point, très important, est que l'utilisation de caractéristiques peu corrélées n'apporte pas nécessairement une amélioration et qu'elle s'accompagne généralement d'un EcR assez élevé. On peut en déduire que la classification produite n'est pas stable d'un coup sur l'autre. L'utilisation des axes de l'ACP permet d'améliorer nettement cette stabilité (EcA et EcR faibles), avec un TEE équivalent au meilleur trouvé jusqu'alors si l'on utilise 2 axes (avec 3 axes les performances se dégradent).

Dans tous les cas, les résultats sont toujours en-dessous de ce à quoi on pourrait s'attendre. Cela s'explique là encore par le fait que pour un certain nombre de signataires, des signatures ne sont pas dans la même classe que les signatures de références, voire que toutes les signatures de références ne sont pas dans la même classe. Cela signifie qu'il semble difficile de trouver une classification qui soit correcte pour tous les utilisateurs en même temps.

4.5 Résultats avec les *Fuzzy C-means*

Les résultats précédents tendent à montrer que la classification est en général extrêmement sensible (écart-type élevé) et pas toujours favorable à tous les signataires. Par conséquent, nos dernières expérimentations ont portées sur l'utilisation d'un algorithme de classification plus stable que les *K-means* : les *Fuzzy C-means*. Le tableau 4 montre les performances obtenues avec cet algorithme dans les mêmes conditions que précédemment. Les résultats sont

[4] Nous avons également évalué plusieurs combinaisons avec plus de 3 caractéristiques mais cela n'améliorait pas les performances du système. Enfin, nous avons utilisé l'algorithme de sélection de caractéristiques SFFS en essayant d'optimiser une mesure d'entropie et une mesure d'ambiguïté, mais sans succès notable.

fournis avec différents types de caractéristiques et à chaque fois avec la meilleure valeur de *K*.

Caracs	*K*	TFA	EcA	TFR	EcR	TEE
longTotale tempsTotal	3	1,75	0,00	2,00	0,00	1,87
ACP x2	3	1,66	0,00	1,67	0,07	1,66
ACP x3	2	1,79	0,00	2,00	0,00	1,89

TAB. 4 – Performances avec *K* classes et en utilisant différentes caractéristiques.

Comme attendu, nous pouvons constater que la classification est très stable en général : Ec quasi nul (y compris pour d'autres valeurs de *K* faibles). De plus, le TEE est systématiquement meilleur par rapport aux tests équivalents avec les *K-means*. On arrive même à une classification particulièrement avantageuse avec 3 classes et les 2 axes de l'ACP. Le gain obtenu est alors de 14,4% sur le TEE.

5 Conclusions et perspectives

Dans ce papier, nous avons étudié l'impact de la création de classes de signatures pour un système d'authentification en ligne. Ces classes sont élaborées automatiquement par les *K-means/Fuzzy C-means* sans utilisation de connaissances a priori, ce qui permet d'utiliser la méthode à partir de n'importe quelle base d'expérimentation. Une amélioration des performances peut ainsi être obtenue en spécialisant le seuil de décision pour chacune des classes. Quand un nouvel utilisateur s'enrôle, le seuil utilisé est celui de la classe la plus en adéquation avec son style de signature.

Les expérimentations, conduites sur la base SVC, ont permis de mettre en évidence l'importance du choix de l'espace de représentation, du nombre de classes ainsi que de l'algorithme de classification. Ainsi, on a pu constater que le choix de caractéristiques peu corrélées n'apportait pas toujours de bonnes performances et qu'elles s'accompagnaient souvent d'un écart-type important sur le TFR. L'utilisation des axes de l'ACP permet de réduire cette instabilité mais il ne garantie pas à lui seul de bonnes performances. Il en va de même de l'utilisation des *Fuzzy C-means*, bien qu'elles améliorent globalement la qualité de la classification et des résultats. Enfin, on constate qu'un trop grand nombre de classes dégrade rapidement le TEE. Ainsi, les meilleurs résultats – TEE de 1,66%, i.e. un gain de 14,4% par rapport au système initial - sont obtenus pour une unique combinaison de ces critères : utilisation des *Fuzzy C-means* avec 3 classes et les 2 axes issus de l'ACP. Les autres combinaisons améliorent très faiblement le TEE ou plus généralement le dégrade.

La difficulté à trouver la bonne combinaison vient du fait que la classification obtenue n'est pas pertinente pour tous les signataires. Notamment, certains ont des signatures qui appartiennent à plusieurs classes. Dans ce cadre, il semblerait plus avantageux : soit de considérer qu'un même signataire puisse appartenir à différente classes, soit de faire plusieurs classifications à partir de différentes caractéristiques et de choisir pour chaque signataire la classification la plus adaptée.

Références

[BEN 05] A. BENSEFIA, T. PAQUET, L HEUTTE, Identification et vérification du scripteur dans des documents manuscrits, *Traitement du Signal*, vol. 22, no. 3, pp.249-259, 2005.

[BEZ 81] J. C. BEZDEK, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, 1981.

[CRET 95] J.-P. CRETTEZ, A set of handwriting families: style recognition. Actes de *ICDAR'95*, vol. 1, pp. 489–494, 1995.

[JAI 02] A. JAIN, F. GRIESS, S. CONNELL, On-line signature Verification, *Pattern Recognition*, vol. 35, n°12, pp. 2963–2972, 2002.

[KHO 06] A. KHOLMATOV, B. YANIKOGLU, Identity authentication using improved online signature verification method, *Pattern Recognition Letters*, vol. 26, pp. 2400–2408, 2005.

[LEJ 01] D. Z. LEJTMAN, S. E. GEORGE, On-line handwritten signature verification using wavelets and back-propagation neural networks, Actes de *ICDAR'01*, pp. 992-996, 2001.

[MCQ 67] J. B. MCQUEEN, Some methods for classification and analysis of multivariate observations, Actes du *5ème Symposium on Mathematical Statistics and Probability de Berkeley*, vol. 1, pp. 281–296, 1967.

[NOS 99] A. NOSARY, L. HEUTTE, T. PAQUET, Y. LECOURTIER, Defining writer's invariants to adapt the recognition task. Actes de *ICDAR'99*, vol. 1, pp. 765–768, 1999.

[OHI 00] T. OHISHI, Y. KOMIYA, T. MATSUMOTO, On-line signature verification using pen-position, pen-pressure and pen-inclination trajectories, Actes de *ICPR'00*, vol. 4, pp. 547–550, 2000.

[SED 02] M.-J. SEDEYN, *Délits d'écrits : lettres anonymes, faux témoignages, chèques falsifiés...*, Éd. Alternatives, 2002.

[SER 02] A. SEROPIAN, N. VINCENT, Writers authentication and fractal compression. Actes de *IWFHR'02*, pp. 434–439, 2002.

[SID 07] I. SIDDIQI, N. VINCENT, Writer Identification in Handwritten Documents, Actes de *ICDAR'07*, vol. 1, pp. 108-112, 2007.

[SVC 04] Signature Verification Competition: http://www.cs.ust.hk/svc2004/download.html, 2004.

[WIR 05a] M. WIROTIUS, J.-Y. RAMEL, N. VINCENT, Contribution of global temporal information for authentication by on-line handwritten signatures. Actes de *IGS'05*, pp. 266-270, 2005.

[WIR 05b] M. WIROTIUS , Authentification par signature manuscrite sur support nomade, *Thèse de doctorat en Informatique*, Université de Tours, 2005.

[YEU 04] D.-Y. YEUNG, H. CHANG, Y. XIONG, S. GEORGE, R. KASHI, T. MATSUMOTO AND G. RIGOLL, SVC2004: First International Verification Competition, Actes de *ICBA'04*, pp. 16-22, 2004.