Authentification faiblement contrainte par dynamique de frappe au clavier

Romain Giot Mohamad El-Abed Christophe Rosenberger

Laboratoire GREYC
ENSICAEN - Université de Caen Basse Normandie - CNRS

6 Boulevard Maréchal Juin
14000 Caen Cedex
{romain.giot,mohamad.elabed,christophe.rosenberger}@greyc.ensicaen.fr

Résumé

Nous proposons dans cette communication, une méthode d'authentification par dynamique de frappe au clavier basée sur l'utilisation d'une passphrase commune à tous les utilisateurs. Les mécanismes principaux de ce système sont : l'utilisation de peu de données pour créer le modèle de l'utilisateur (5 saisies uniquement), un mécanisme d'apprentissage incrémental et un séparateur à vaste marge pour l'apprentissage. Les résultats expérimentaux sur une base contenant 100 individus montrent l'apport de ce nouveau système.

Mots Clef

Biométrie, dynamique de frappe au clavier, séparateur à vaste marge, apprentissage incrémental.

Abstract

In this communication, we propose an authentification method based on the use of a passphrase associated to keystroke dynamics. The main mechanisms of the system are the use of few data to create users' model (5 captures), the use of an incremental learning and SVM. Experimentals results on a benchmark containing 100 individuals show the benefit of this new system.

Keywords

Biometrics, keystroke dynamics, SVM, incremental learning.

1 Introduction

L'accès par des entités à des ressources contrôlées est généralement géré par des systèmes d'authentification répondant aux deux questions suivantes : qui est l'utilisateur ? et, l'utilisateur est-il bien celui qu'il dit être ? Dans cet article, nous nous intéressons au second cas, où nous voulons vérifier l'identité de la personne à l'aide de sa façon de taper au clavier.

Le but de la dynamique de frappe au clavier (ddf) est de sécuriser l'utilisation du couple identifiant/mot de passe qui souffre de différentes lacunes [20] : les mots de passe peuvent être échangés entre les utilisateurs, volés ou devinés. La ddf ajoute une dimension supplémentaire : ce qui qualifie l'utilisateur ou son comportement (sa façon de taper) dans le but de renforcer la sécurité l'authentification par mot de passe.

Le fonctionnement d'un système biométrique nécessite deux processus principaux : (i) *l'enrôlement* qui consiste à capturer les données biométriques de l'utilisateur afin de créer son modèle et d'ajouter ce dernier au système, (ii) et la *vérification* qui consiste à effectuer une capture et la comparer au modèle afin de prendre la décision d'accepter (en fonction d'un seuil de décision), ou rejeter, l'utilisateur présumé. La capture des données dans le cas de la ddf consiste à enregistrer les instants où les touches sont pressées et relâchées afin de calculer leurs temps de pression et différents temps de latences (Figure 1).

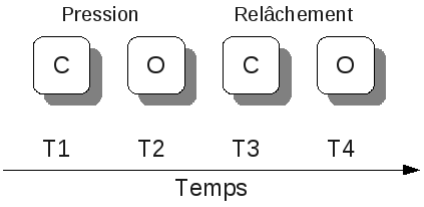

FIG. 1 – Capture de la donnée biométrique. La touche O est pressée avant le relâchement de C (l'utilisateur tape vite)

Différentes métriques permettent de mesurer les performances d'un système biométrique, la liste suivante pré-

sente celles utilisées dans ce papier :

le FAR *False Acceptance Rate*, ou *Taux de Fausse Acceptation*, représente le ratio de captures d'imposteurs acceptées par le système par rapport au nombre total de captures d'imposteur ;

le FRR *False Reject Rate*, ou *Taux de Faux Rejet* représente, le ratio de captures d'utilisateurs légitimes rejetées par le système par rapport au nombre total de captures d'utilisateur légitime ;

le EER *Error Equal Rate*, ou *Taux d'Erreur Égale*, représente le taux d'erreur, lorsque le seul du système est configuré de telle façon à obtenir $FAR = FRR$.

Dans cet article, nous proposons une nouvelle méthode d'authentification par passphrase (l'ensemble des utilisateurs possèdent le même mot de passe, il n'y a donc pas de secret) nécessitant peu de données d'enrôlement (5), et, la comparons à différentes méthodes de l'état de l'art selon différents scénarios. Les sections suivantes présentent un état de l'art de la ddf, notre méthode, le protocole et les résultats expérimentaux. Nous concluons et donnons les différentes perspectives de cette étude.

2 Travaux antérieurs

2.1 Historique

Les premiers travaux sur la ddf sont présentés dans un rapport de la Rand Corporation [6] et datent de 1980. Cette étude a permis de montrer que l'on pouvait distinguer les utilisateurs à leur façon de saisir de longs textes. Par la suite, les chercheurs ont essayé d'améliorer les performances de reconnaissance tout en diminuant la quantité de texte à saisir afin de créer le modèle de l'utilisateur.

En 1986, un brevet américain décrit un schéma dans lequel les utilisateurs saisissent leur nom afin de s'authentifier [7]. Il suppose que le mot de passe sera forcément facile à retenir et que la saisie sera plus constante du fait de saisir quelque chose d'habituel. Une idée proposée dans le brevet est d'utiliser le vecteur moyen des temps de digraphes comme modèle, une distance de Mahalanobis étant ensuite effectuée entre le modèle de l'utilisateur présumé et la capture. Si cette distance est supérieur à 100, l'utilisateur est rejeté, tandis que si elle est inférieure à 50, l'utilisateur est accepté. Dans le cas où elle est entre 50 et 100, il est nécessaire qu'il fasse une nouvelle saisie. Le brevet décrit un autre système où les utilisateurs saisissent 10 fois 1000 des mots les plus courants afin de leur générer un profil, tandis que l'authentification se fait en saisissant une phrase générée aléatoirement.

En 1997, Monrose et Rubin, ont travaillé sur l'analyse de textes libres [14]. Ils recommandent de séparer les utilisateurs en différents groupes (en fonction de leur vitesse de frappe) afin d'accélérer le temps de reconnaissance. Différentes études sur des méthodes statistiques et des réseaux de neurones [15] ont également été effectuées. Les auteurs ont également vérifié les performances d'utilisation des durées de frappe comme mesure, ce qu'a confirmé l'étude. Les meilleures performances ont été obtenues en utilisant à la fois les temps de latence et la durée de pression. Ils ont obtenu un taux d'erreur de 0% à l'aide d'un réseau de neurones utilisant 112 captures pour l'enrôlement et un FAR de 1,9% et FRR de 0,7% en utilisant une mesure de distance.

Améliorer les performances des systèmes semble de plus en plus difficile, cependant il est toujours possible d'améliorer la consistance (diminution de la variabilité intra-classe) de la frappe de l'utilisateur afin de diminuer les erreurs. Hwang *et al.* soutiennent le fait que la qualité des mesures utilisées pour créer le vecteur de référence est plus importante que leur quantité [13]. Ainsi, dans leur étude, les auteurs ont essayé d'augmenter la consistance des motifs en utilisant des pauses (dans le rythme de la frappe) à l'aide de signaux afin d'améliorer les performances d'authentification sans modifier les algorithmes (augmentation de la séparabilité entre les utilisateurs).

Dans [9], les auteurs présentent trois méthodes différentes (basées sur des adaptations et améliorations de méthodes existantes) utilisées en fusion afin d'améliorer les performances. Les méthodes sont de type statistique, basées sur le rythme de frappe, ou la mesure du désordre. Le EER obtenu avoisine les 5%, mais il a été calculé avec une base relativement petite. Les auteurs de [19] ont quant à eux testé la ddf en utilisant un *Séparateur à Vaste Marge*. Ils ont testé des SVM à 1 et 2 classes. Dans le cas d'un SVM à 2 classes les données des imposteurs ont été générées à partir des données de l'utilisateur. Seulement 10 utilisateurs et 5 imposteurs ont pris part à l'étude. Le SVM à une classe a de meilleures performances et un temps de calcul plus faible que l'utilisation de réseaux de neurones (pour la génération du modèle).

D'une manière générale, la création du modèle nécessite une quantité conséquente de captures, ce qui est difficilement applicable en milieu opérationnel.

2.2 Discussion

Nous avons donc vu qu'il existe différentes études sur le sujet. Cependant, contrairement aux autres modalités biométriques (comme l'iris, la signature ou les empreintes digitales), elles ont toutes été faites avec des bases de données différentes : il n'existe pas de base conséquente publique sur la ddf.

L'utilisation de bases de données différentes empêche la réelle comparaison des méthodes de l'état de l'art car le protocole de création de la base diffère en terme de temps

de création de la base, quantité d'utilisateurs, quantité de captures par utilisateurs, etc ... Le tableau 1 présente, pour quelques études, les différences entre les bases de données et leurs configurations.

3 Méthode développée

Le but de la méthode proposée est de limiter le nombre de captures nécessaires pour l'enrôlement (pour des raisons évidentes d'utilisabilité) tout en ayant de bonnes performances.

La méthode que nous proposons est nouvelle, dans le sens où elle nécessite peu de données de la part de l'utilisateur pour l'enrôlement et, que les données sont discrétisées avant d'être utilisées pour l'apprentissage d'un séparateur à vaste marge à deux classes.

Les parties suivantes détaillent le fonctionnement du système, tandis que la Figure 2 en présente un résumé.

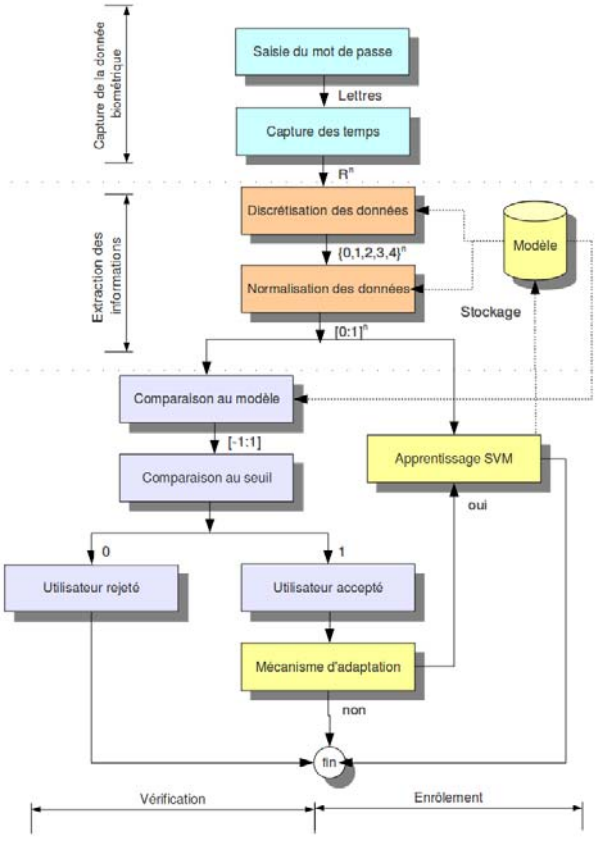

FIG. 2 – Fonctionnement du mécanisme développé.

3.1 Capture

La capture de la donnée biométrique se déroule à la fois lors de l'enrôlement et de la vérification. La capture est faite à partir de la saisie de la passphrase par l'utilisateur. Un vecteur de temps est calculé à partir des données brutes : il est constitué de la concaténation de quatre autres vecteurs de temps (celui codant la durée de pression de chaque touches ($T3 - T1$), ceux des trois types de temps de latence ($T2 - T1$, $T4 - T2$ et $T3 - T2$), voir Figure 1). Ainsi, pour une passphrase de n caractères, le vecteur obtenu a une dimension de $3 * (n - 1) + n$.

Les données sont ensuite discrétisées dans un alphabet de cinq caractères : chaque dimension du vecteur est partitionnée en 5 intervalles de taille égale et la nouvelle valeur est le numéro de la partition associée au temps initial.

3.2 Enrôlement

Les utilisateurs doivent saisir cinq fois une passphrase définie par l'administrateur du système, ainsi, le modèle d'un utilisateur est créé à l'aide de 5 de ses captures et de $5 * m$ captures d'imposteurs pour m imposteurs. Ensuite, un séparateur à vaste marge est utilisé pour l'apprentissage.

Le principe de base des séparateurs à vastes marges [21] est de classifier différentes classes à l'aide d'une marge maximale associée à des vecteurs supports et d'une fonction noyau. La fonction noyau permet d'opérer un changement de repère dans un espace de plus grande dimension afin de se retrouver à un problème de séparation linéaire, lorsque les données ne sont pas linéairement séparables. La frontière de séparation est celle qui maximise la marge (distance entre la frontière de séparation et les échantillons les plus proches). La Figure 3 présente le principe de la marge maximale.

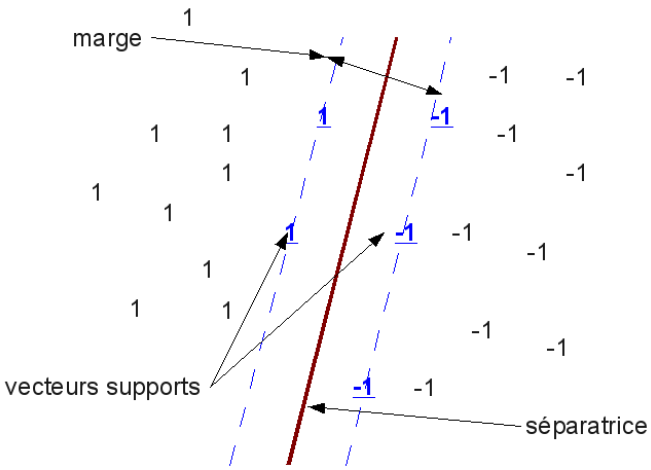

FIG. 3 – Présentation de la séparation linéaire avec la marge maximum et les vecteurs supports associés.

TAB. 1 – Extrait des différences entre les protocoles des méthodes de l'état de l'art. Les informations présentes sont les auteurs de l'article, la durée d'acquisition, le nombre d'individus concernés, si il s'agit d'une acquisition contrôlée, si le seuil utilisé est global au système ainsi que les FAR et FRR du système. " ? ?" indique que l'article ne présente pas l'information.

Papier	**Durée**	**Individus**	**Enrôlement**	**Acquisition**	**Seuil global**	***FAR***	***FRR***
Obaidat et Sadoun [15]	8 semaines	15	112	non	non	0%	0%
Bleha *et al.* [3]	8 semaines	36	30	oui	yes	2.8%	8.1%
Rodriguez *et al.* [18]	4 sessions	20	30	? ?	no	3.6%	3.6%
Hocquet *et al.* [11]	? ?	38	? ?	? ?	non	1.7%	2.1%
Revett *et al.* [17]	14 jours	30	10	? ?	non	0.15%	0.2%
Hosseinzadeh et Krishnan [12]	? ?	41	30	non	non	4.3%	4.8%

Dans notre cas, nous avons utilisé le noyau linéaire ($K(x_i, x_j) = {x_i}^T \cdot x_j$) qui a donné de meilleurs résultats que le noyau gaussien. Nous utilisons un SVM à deux classes : la classe 1 est constituée des n données d'apprentissage de l'utilisateur, tandis que la classe -1 est constituée des n données d'apprentissage des m imposteurs.

Les calculs ont été faits grâce à la bibliothèque libsvm [4].

3.3 Vérification

Le mécanisme de vérification consiste à réaliser une procédure de reconnaissance de la donnée biométrique capturée à l'aide de l'algorithme du SVM. Les données sont préalablement discrétisées à l'aide de l'alphabet présenté précédemment.

Un score est calculé à l'aide des informations retournées par l'algorithme de reconnaissance SVM. Nous avons choisi de calculer le distance de la façon suivante :

$$SVM = -prb * prd \tag{1}$$

avec prb étant la probabilité du résultat (entre 0 et 1), et prd la classe prédite (-1 ou 1). De cette façon, les distances négatives correspondent à l'utilisateur et les positives aux imposteurs. Plus la valeur absolue de la distance est importante, plus la certitude l'est également. Ainsi, nous pouvons prendre un seuil de décision différent de 0 afin de prendre en compte les captures de l'utilisateur mal classées.

Le score est ensuite comparé à un seuil, qui peut être global au système, ou individuel à l'utilisateur, afin de prendre la décision d'accepter ou rejeter l'utilisateur. Si la vérification s'avère être fructueuse, un mécanisme de mise à jour du modèle peut être mis en place après l'acceptation de l'utilisateur.

3.4 Mise à jour du modèle biométrique

La dynamique de frappe au clavier fait partie de la famille des modalités biométriques comportementales, c'est pourquoi, utiliser un mécanisme qui prend en compte l'évolution de la façon de taper, afin d'augmenter les performances peut être intéressant [1, 16].

Quatre différents moyens d'évolution ont été définis :
- aucune adaptation (nous gardons les cinq premiers vecteurs comme vecteurs d'enrôlement). Cette approche est notée "Classique" ;
- une méthode remplaçant, après chaque vérification, la plus ancienne capture par la capture de vérification ; de cette façon il y a toujours cinq vecteurs d'enrôlement. Cette approche est notée "Adaptative" ;
- une méthode ajoutant, après chaque vérification, la capture de test ; le nombre de captures d'enrôlement augmente donc progressivement. Cette approche est nommée "Progressive" ;
- une méthode nommée "Intelligente" basée sur les deux précédentes qui ajoute le vecteur tant qu'il y a moins de quinze vecteurs d'enrôlement et qui remplace le plus ancien sinon. De plus, cet ajout n'est fait que lorsque le vecteur de test (qui appartient toujours à l'utilisateur) n'est pas trop différent (en terme d'écart à la moyenne) des vecteurs du modèle.

Cependant, la mise à jour du modèle biométrique apporte une complexité supplémentaire au problème, car il devient possible, en cas d'erreur, d'ajouter au modèle les données d'un imposteur.

4 Validation du système

4.1 Base de données

Étant donné qu'il n'existe pas de base de données publique conséquente sur la ddf, nous avons décidé de créer la notre et de la rendre publique [8]. Nous avons respecté les principes énoncés dans [5] afin de créer une base de qualité.

133 utilisateurs ont participé à la création de cette base. Ils ont eu la possibilité de participer à une session de capture une à deux fois par semaine pendant plus de deux mois.

Lors d'une session d'acquisition, l'utilisateur devait saisir, par alternance, 6 fois le mot de passe "greyc laboratory" sur deux claviers différents. Une session est ainsi composée de 12 captures. Nous avons choisi ce mot de passe pour

plusieurs raisons : (i) il s'agit du nom de notre laboratoire et contribue à sa promotion, et (ii) il s'agit d'un mot de passe suffisamment long [3] avec un bon positionnement des touches sur le clavier (ce qui aide pour la discriminalité) [16].

Pour cette étude, 100 utilisateurs ont été retenus. Il s'agit des utilisateurs ayant proposé au minimum 60 captures. Les données extraites sont composées des 4 différences de temps possibles : RR, RP, PR et PP (avec R pour relâchement et P pour pression). Il y a donc le temps de pression et 3 temps de latence pour chaque couple de touches.

4.2 Méthodes sélectionnées

Nous avons choisi de sélectionner et implémenter les méthodes de l'état de l'art qui se rapprochent le plus de nos attentes (nécessitant peu de données pour l'enrôlement). Différentes "familles" de méthodes ont été choisies : statistique, distance et rythme.

Les symboles suivants sont utilisés pour représenter :
– v correspond au vecteur à tester (de taille n) ;
– μ est le vecteur moyen des vecteurs d'enrôlement ;
– σ est l'écart type des vecteurs d'enrôlement.

L'ensemble des méthodes retourne une distance : plus le score est petit, plus le vecteur de test est proche du modèle de l'utilisateur.

Algorithmes statistiques. Deux méthodes statistiques sont testées. La première ne prend pas en compte l'écart type des différents temps [3] :

$$STAT1 = \frac{(v-\mu)^t (v-\mu)}{||v||.||\mu||} \quad (2)$$

tandis que la seconde est basée à la fois sur la moyenne et l'écart type des temps [11] :

$$STAT2 = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|v_i-\mu_i|}{\sigma_i}} \quad (3)$$

Calcul de distance. Le calcul de distance est basé sur une distance euclidienne [14] :

$$DIST = \min\left(\forall_{u\subset enrol}, \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}\right) \quad (4)$$

Méthode basée sur le rythme de frappe. Cette méthode consiste à discrétiser les temps selon un alphabet de cinq caractères, puis de calculer une distance de Manhattan entre le modèle discrétisé et le vecteur de test discrétisé [11] :

$$RYTHM = \frac{1}{n}\sum_{i=1}^{n} abs(classe(v_i) - classe(\mu_i)) \quad (5)$$

avec $classe(i)$ une fonction qui retourne la classe de i. Pour calculer cette classe, l'espace est partitionné en cinq partitions de taille identique entre le temps maximum et minimum. La classe correspond au numéro de la partition comprenant le temps.

4.3 Résultats expérimentaux

Le but de cette section est de présenter les résultats expérimentaux aux questions suivantes :
– L'utilisation d'un clavier différent a-t-il une incidence sur les performances des algorithmes ?
– Comment évoluent les performances en fonction du nombre de captures pour l'enrôlement ?
– Est-il nécessaire de prendre en compte l'évolution de la façon de taper au cours du temps ?
– Est-il nécessaire d'utiliser un seuil différent en fonction des utilisateurs ?
– Les résultats dépendent ils du nombre d'utilisateurs dans la base de données ?

Les différentes études ont été faites sur notre algorithme et ceux de l'état de l'art en utilisant 5 captures pour l'enrôlement et des données en provenance des deux claviers.

Différences entre les deux claviers. Le tableau 2 représente les différents EERs en fonction du clavier d'origine des captures utilisées pour l'apprentissage et le test.

Le EER de chaque méthode est calculé en utilisant les 10 premières captures de l'utilisateur pour l'enrôlement et toutes les autres pour la vérification. Le clavier source des captures peut être différent. Aucun mécanisme d'adaptation n'est utilisé. Lorsque la source des captures de test et d'enrôlement est différente, le calcul est fait plusieurs fois en sélectionnant les captures d'enrôlement aléatoirement et en moyennant les résultats.

Les colonnes EER11 et EER22 représentent respectivement le EER lorsque les données appartiennent uniquement au clavier 1 et 2. La colonne EER12 représente le EER calculé en utilisant le clavier 1 pour l'apprentissage et le clavier 2 pour la vérification (et vice-versa pour EER21). La colonne EERaa correspond à l'utilisation des données en provenance des deux claviers à la fois pour l'enrôlement et la vérification.

Les résultats ne sont pas exactement égaux d'une configu-

TAB. 2 – Taux d'erreur des méthodes en fonction des claviers d'origine. "EERnm" signifie que les données d'enrôlement proviennent du clavier "n" et les données de vérification du clavier "m", avec "1", "2", "a"représentant respectivement les données du clavier 1, 2 ou de n'importe lequel d'entre eux. Le meilleur EER de chaque configuration est représenté en gras.

Methode	EER11	EER22	EER12	EER21	EERaa
STAT1	24,91%	23,96%	24,73%	23,51%	25,50%
STAT2	17,68%	16,55%	17,10%	16,65%	17,58%
DIST	27,01%	26,00%	26,46%	25,07%	27,56%
RYTHM	19,40%	20,09%	19,25%	19,50%	19,78%
SVM	**10,68%**	**10,37%**	**10,30%**	**11,76%**	**11,96%**

ration à l'autre, mais nous ne pouvons pas noter de différences majeures ou corrélations particulières. Cependant, nous pouvons remarquer que les résultats les moins bons correspondent, la plupart du temps, à la configuration EERaa. Quatre fois sur six, les meilleurs résultats sont obtenus lorsque les données de test et d'enrôlement proviennent du même clavier. Dans toutes les configurations, les meilleurs résultats sont obtenus grâce à notre méthode.

Analyse du nombre de captures pour l'enrôlement. Une étude intéressante est la visualisation de l'évolution du EER en fonction du nombre de captures nécessaires pour créer le modèle. Ce nombre étant différent en fonction des études, et, souvent au moins égal à vingt tandis que cinq semble être le maximum acceptable par l'utilisateur. Utiliser cette information peut faciliter la comparaison des articles ne proposant pas le même nombre de captures d'enrôlement pour leur méthode.

La Figure 4 représente le EER des differents algorithmes en fonction du nombre de captures utilisées pour créer le modèle. Il est clair que plus le nombre de captures est important, plus le modèle semble représentatif. Pour toutes les méthodes, moins de dix captures donne de mauvais résultats, le minimum semble être quarante[1], bien que le gain entre vingt et quarante ne soit pas très important. Au delà de cinquante captures, les performances peuvent même se dégrader. Ces résultats confirment l'étude réalisée dans [2]. Dans cette analyse aussi, notre méthode donne de meilleurs résultats.

Utilisation d'un mécanisme incrémental. Le tableau 3 présente le EER des méthodes en utilisant les différents modes d'adaptation. Comme expliqué précédemment, le seuil est global, et cinq captures sont utilisées pour créer le modèle initial. Les captures proviennent des deux claviers sans faire de distinction.

Nous pouvons voir dans ce tableau qu'il est important de prendre en compte la modification de la façon de saisir le mot de passe. Pour la plupart des algorithmes, la méthode la plus efficace est "Intelligente" même si elle est susceptible d'utiliser moins de données que la "Progressive". Ainsi ; filtrer les captures avant de les ajouter au modèle peut réduire le EER d'environ 8%. Les meilleurs résultats sont obtenus avec notre méthode qui obtient un EER de 6.69% dans le mode "Progressive".

[1]Cependant, dans ce cas, le nombre de captures utilisées pour la vérification est plus faible

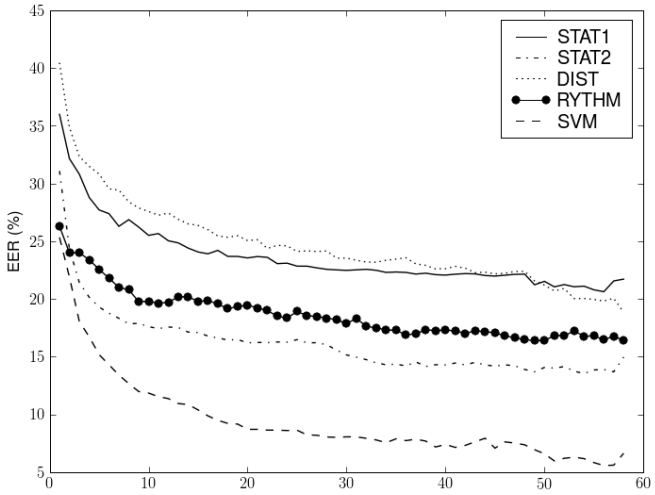

FIG. 4 – Évolution du EER, pour chaque méthode, en fonction du nombre de captures utilisées pour créer le modèle.

Le but de ce test est de montrer qu'il y a une évolution dans la façon de taper de l'utilisateur, pas de présenter la méthode d'adaptation la plus intéressante dans un contexte d'utilisation normal. Une méthode plus opérationnelle consisterait à essayer d'adapter le modèle de l'utilisateur si : la vérification est un succès, la capture passe le filtrage, et avec tous les vecteurs de tests ; il y a donc un risque d'ajouter des imposteurs au modèle (et de dégradation des performances), ce qui n'est pas le cas ici.

TAB. 3 – EER des méthodes en utilisant les différents mécanismes incrémentaux et cinq captures pour créer le modèle initial. Le meilleur EER de chaque mécanisme est représenté en gras.

Méthode	Classique	Progressive	Adaptive	Intelligente
STAT1	27,7%	21,24	23%	20,94%
STAT2	19,29%	15,09%	11,71%	10,75%
DIST	30,81%	23,75%	25,7%	24,65%
RYTHM	22,56%	15,49%	14,36%	13,21%
SVM	**15,38%**	**6,69%**	**9,21%**	**6,96%**
Moyenne	23,15%	16,45%	16,8%	**15,3%**

Utilisation d'un seuil individuel. Les performances statistiques des systèmes biométriques sont différents en fonction de l'utilisation d'un seuil global ou différent pour chaque utilisateur (ou classe d'utilisateurs) [12, 10].

Le seuil spécifique à un utilisateur est celui minimisant le EER correspondant. Celui-ci est calculé à l'aide de ces propres captures pour le FRR et les captures des imposteurs pour le FAR. Utiliser un seuil individuel est supposé améliorer les performances. Le tableau 4 présente, pour chaque méthode, les améliorations en terme de ERR dues à l'utilisation d'un seuil individuel. Les EERs ont été calculés en utilisant cinq captures pour le modèle, ainsi que le mécanisme d'adaptation "Intelligent" avec les données des deux claviers.

TAB. 4 – EER des méthodes en utilisant un seuil global et individuel, en utilisant les données des deux claviers et un mécanisme d'adaptation.

Methode	EER(global)	EER(individuel)	Gain
STAT1	20,94%	19,54%	1,4%
STAT2	10,75%	9,22%	1,53%
DIST	24,65%	21,53%	3,12%
RYTHM	13,21%	10,02%	**3,18%**
SVM	**6,96%**	**6,95%**	0,01%
Moyenne	15,3%	**13,45%**	1,85%

Nos pouvons voir, pour chaque méthode, qu'il y a une légère amélioration des performances de 1,5% en utilisant un seuil individuel. Cependant, pour le configurer, il est nécessaire de posséder des données d'imposteurs (ce qui n'est pas trivial dans un cas autre qu'une authentification par passphrase commune à tous les utilisateurs) ainsi que d'une quantité de données de test suffisante pour le calculer et le configurer avant même que les utilisateurs n'utilisent le système. Une solution à ce problème est présentée dans [11] : une base d'apprentissage (contenant des données d'imposteurs) est utilisée afin de classer les utilisateurs en fonction de différents paramètres (taille du mot de passe, temps, résultats des méthodes, etc) ; lors de l'authentification, les paramètres de l'utilisateur sont ceux du cluster le plus proche.

Dépendance à la base de données biométrique. Comme il est connu que les performances des systèmes biométriques dépendent de la base de données utilisée, nous avons décidé de calculer les performances des méthodes en fonction du nombre d'utilisateurs présents dans la base de données.

La Figure 5 présente cette évolution.

Nous pouvons voir que moins de dix utilisateurs est totalement insuffisant pour tester les performances des algorithmes. Plus il y a d'utilisateurs dans la base, plus les performances diminuent. Les performances sont donc dépendantes de la base de données, et, notamment du nombre d'utilisateurs la composant. Cinquante utilisateurs semble être le nombre minimum acceptable pour obtenir des résultats réalistes (alors que la plupart des études ne respectent pas cette contrainte).

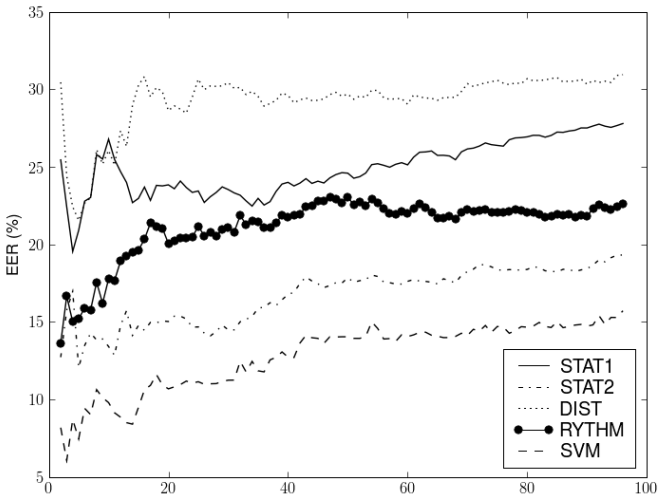

FIG. 5 – Évolution des performances en fonction du nombre d'utilisateurs dans la base de données

5 Conclusion

Les résultats des méthodes de l'état de l'art sont moins bons que dans leur article d'origine. Cela peut s'expliquer par plusieurs raisons : le nombre de captures utilisées pour créer le modèle est inférieur, les données extraites ne sont pas forcément les mêmes, la résolution de l'horloge utilisée pour capturer les temps est sûrement différente. Nous avons également montré que le nombre d'utilisateurs dans la base de données influe également sur les performances du système. Cependant, dans tous les cas, les résultats de notre méthode les dépassent. Ce gain de performance peut être du au fait que les données des imposteurs sont pris en compte dans la création du modèle. Notre méthode reste cependant plus performante et surtout plus rapide (lors de la création du modèle) que l'utilisation d'un réseau de neurones (résultats non présentés dans cette étude).

Nous avons également montré qu'il existe un manque de base de données biométrique dans le cas de la dynamique de frappe au clavier. Notre contribution est également d'apporter une base conséquente afin de pouvoir être utilisée pour analyser les performances des futurs algorithmes.

Il pourrait être intéressant d'analyser les performances du mécanisme lorsque tous les imposteurs ne sont pas utilisés lors de la création des modèles (afin d'évaluer la périodicité de la mise à jour du modèle à mettre en place), et en couplant un mécanisme d'identification à celui de vérification afin de voir si les performances peuvent être améliorées. Améliorer le mécanisme afin d'être utilisé dans un système où tous les utilisateurs ont un mot de passe différent semble également intéressant.

Références

[1] L.C.F. Araujo, Jr. Sucupira, L.H.R., M.G. Lizarraga, L.L. Ling, and J.B.T. Yabu-Uti. User authentication through typing biometrics features. *IEEE Transactions on Signal Processing*, 53(2 Part 2) :851–855, 2005.

[2] D. Bartmann, I. Bakdi, and M. Achatz. On the design of an authentication system based on keystroke dynamics using a predefined input text. *Techniques and Applications for Advanced Information Privacy and Security : Emerging Organizational, Ethical, and Human Issues*, 1(2) :149, 2007.

[3] S. Bleha, C. Slivinsky, and B. Hussien. Computer-access security systems using keystroke dynamics. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 12 (12), 1990.

[4] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] Fouad Cherifi, Baptiste Hemery, Romain Giot, Marc Pasquet, and Christophe Rosenberger. *Behavioral Biometrics for Human Identification : Intelligent Applications*, chapter Performance Evaluation Of Behavioral Biometric Systems, pages 67–74. IGI Global, 2009.

[6] R. Gaines, W. Lisowski, S. Press, and N. Shapiro. Authentication by keystroke timing : some preliminary results. Technical report, Rand Corporation, 1980.

[7] John D. Garcia. Personal identification apparatus, November 1986.

[8] Romain Giot, Mohamad El-Abed, and Rosenberger Christophe. Greyc keystroke : a benchmark for keystroke dynamics biometric systems. In *IEEE International Conference on Biometrics : Theory, Applications and Systems (BTAS 2009)*, 2009. en cours de publication.

[9] Sylvain Hocquet, Jean-Yves Ramel, and Hubert Cardot. Authentification par la dynamique de frappe. In *15e congrès francophone de Reconnaissance des Formes et Intelligence Artificielle RFIA*, 2005.

[10] Sylvain Hocquet, Jean-Yves Ramel, and Hubert Cardot. Estimation of user specific parameters in one-class problems. In *ICPR '06 : Proceedings of the 18th International Conference on Pattern Recognition*, pages 449–452, Washington, DC, USA, 2006. IEEE Computer Society.

[11] Sylvain Hocquet, Jean-Yves Ramel, and Hubert Cardot. User classification for keystroke dynamics authentication. In *The Sixth International Conference on Biometrics (ICB2007)*, pages 531–539, 2007.

[12] D. Hosseinzadeh and S. Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 38(6) :816–826, 2008.

[13] Seong-seob Hwang, Hyoung-joo Lee, and Sungzoon Cho. Improving authentication accuracy of unfamiliar passwords with pauses and cues for keystroke dynamics-based authentication. *Intelligence and Security Informatics*, 3917 :73–78, 2006.

[14] F. Monrose and Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM Press New York, NY, USA, 1997.

[15] MS Obaidat and B. Sadoun. Verification of computer users using keystroke dynamics. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 27(2) :261–269, 1997.

[16] K. Revett, S.T. de Magalhães, and H.M.D. Santos. Enhancing login security through the use of keystroke input dynamics. *Lecture notes in computer science*, 3832 :661, 2006.

[17] K. Revett, S.T. de Magalhaes, and H.M.D. Santos. On the use of rough sets for user authentication via keystroke dynamics. *Lecture notes in computer science*, 4874 :145, 2007.

[18] R.N. Rodrigues, G.F.G. Yared, CR do NCosta, J.B.T. Yabu-Uti, F. Violaro, and L.L. Ling. Biometric access control through numerical keyboards based on keystroke dynamics. *Lecture notes in computer science*, 3832 :640, 2006.

[19] Y. Sang, H. Shen, and P. Fan. *Parallel and Distributed Computing : Applications and Technologies*, chapter Novel impostors detection in keystroke dynamics by support vector machine, pages 666–669. Springer, 2005.

[20] MA Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3) :122–131, 2001.

[21] V. Vapnik. Statistical learning theory. *NY Wiley*, 1998.